

Manuel PHP 4.3.0 – Version Française

<http://www.nexen.net>

Sommaire

Manuel PHP.....	1
1.1 Les auteurs.....	1
1.2 L'éditeur.....	1
1.3 Note légale.....	1
2 Préface.....	2
3 Comment 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.....	7
3.2.4 Utiliser un formulaire.....	9
3.2.5 Utiliser des codes anciens avec les nouvelles versions de PHP.....	10
3.2.6 Et après?.....	10
3.3 Installation.....	10
3.3.1 Considérations générales sur l'installation.....	10
3.3.2 Installation sous UNIX.....	11
3.3.3 Installation sous Linux/Unix.....	12
3.3.4 Installation sous HP-UX.....	13
3.3.5 Installation sous Solaris.....	14
3.3.6 Installations Unix/OpenBSD.....	14
3.3.7 Installation sous Unix et Mac OS X.....	15
3.3.8 Liste des options de configuration diverses.....	17
3.3.9 Installation sous Windows.....	20
3.3.10 Installation avec le serveur Apache.....	28
3.3.11 Installation avec les serveurs Apache 2.0.....	31
3.3.12 CGI/ Installation pour exécution en ligne de commande.....	33
3.3.13 Installation avec les serveurs fhttpd.....	34
3.3.14 Installation sur le serveur Caudium.....	34
3.3.15 Installation avec les serveurs IIS/PWS.....	35
3.3.16 Installation sous Netscape et iPlanet Enterprise Serveur.....	37
3.3.17 Installation pour les serveurs OmniHTTPd.....	41
3.3.18 Installation Oreilly Website Pro Server.....	42
3.3.19 Serveurs Sambar.....	42
3.3.20 Installation Xitami.....	42
3.3.21 Autres serveurs web.....	43
3.3.22 Des problèmes?.....	43
3.4 Configuration.....	44
3.4.1 Le fichier de configuration.....	44
3.4.2 Comment modifier la configuration.....	44
3.4.3 Directives de configuration.....	46
4 Référence.....	52
4.1 La syntaxe de base.....	52
4.1.1 Passer du HTML au PHP.....	52
4.1.2 Le séparateur d'instructions.....	53
4.1.3 Commentaires.....	53
4.2 Les types.....	54

Sommaire

4 Référence

<u>4.2.1 Introduction</u>	54
<u>4.2.2 Booléens</u>	55
<u>4.2.3 Entiers</u>	56
<u>4.2.4 Les nombres à virgule flottante</u>	58
<u>4.2.5 Les chaînes de caractères</u>	59
<u>4.2.6 Les tableaux</u>	65
<u>4.2.7 Les objets</u>	71
<u>4.2.8 Ressources</u>	72
<u>4.2.9 La valeur NULL</u>	72
<u>4.2.10 Pseudo-types utilisés dans cette documentation</u>	73
<u>4.2.11 Définition du type</u>	73
<u>4.3 Les variables</u>	76
<u>4.3.1 Essentiel</u>	76
<u>4.3.2 Variables prédéfinies</u>	77
<u>4.3.3 Portée des variables</u>	78
<u>4.3.4 Les variables dynamiques</u>	82
<u>4.3.5 Variables externes à PHP</u>	83
<u>4.4 Les constantes</u>	85
<u>4.4.1 Syntaxe</u>	86
<u>4.4.2 Constantes prédéfinies</u>	86
<u>4.5 Les expressions</u>	87
<u>4.6 Les opérateurs</u>	89
<u>4.6.1 La précedence des opérateurs</u>	89
<u>4.6.2 Les opérateurs arithmétiques</u>	90
<u>4.6.3 Les opérateurs d'assignation</u>	90
<u>4.6.4 Opérateurs sur les bits</u>	91
<u>4.6.5 Opérateurs de comparaison</u>	91
<u>4.6.6 Opérateur de contrôle d'erreur</u>	92
<u>4.6.7 Opérateur d'exécutions</u>	92
<u>4.6.8 Opérateurs d'incrément/Décrément</u>	93
<u>4.6.9 Les opérateurs logiques</u>	93
<u>4.6.10 Opérateurs de chaînes</u>	94
<u>4.6.11 Opérateurs de tableaux</u>	94
<u>4.7 Les structures de contrôle</u>	94
<u>4.7.1 if</u>	95
<u>4.7.2 else</u>	95
<u>4.7.3 elseif</u>	96
<u>4.7.4 Syntaxe alternative</u>	96
<u>4.7.5 while</u>	97
<u>4.7.6 do..while</u>	97
<u>4.7.7 for</u>	98
<u>4.7.8 foreach</u>	99
<u>4.7.9 break</u>	101
<u>4.7.10 continue</u>	101
<u>4.7.11 switch</u>	102
<u>4.7.12 declare</u>	104
<u>4.7.13 return</u>	105
<u>4.7.14 require</u>	105
<u>4.7.15 include</u>	106
<u>4.7.16 require_once</u>	108
<u>4.7.17 include_once</u>	109

Sommaire

4 Référence

<u>4.8 Les fonctions</u>	110
<u>4.8.1 Les fonctions</u>	110
<u>4.8.2 Les arguments de fonction</u>	111
<u>4.8.3 Les valeurs de retour</u>	113
<u>4.8.4 Variables fonctions</u>	113
<u>4.8.5 Fonctions internes</u>	114
<u>4.9 Les classes et les objets</u>	114
<u>4.9.1 Les classes : class</u>	114
<u>4.9.2 extends : héritage</u>	116
<u>4.9.3 Constructor : constructeur</u>	117
<u>4.9.4 Opérateur ::</u>	118
<u>4.9.5 parent</u>	119
<u>4.9.6 Sauvegarde d'objets – cas des sessions</u>	120
<u>4.9.7 Les fonctions magiques __sleep et __wakeup</u>	121
<u>4.9.8 Références dans un constructeur</u>	121
<u>4.9.9 Comparer des objets en PHP 4</u>	123
<u>4.9.10 Comparer des objets en PHP 5</u>	124
<u>4.10 Les références</u>	126
<u>4.10.1 Qu'est ce qu'une référence?</u>	126
<u>4.10.2 Que font les références ?</u>	126
<u>4.10.3 Ce que les références ne sont pas</u>	127
<u>4.10.4 Passage par référence</u>	127
<u>4.10.5 Retourner des références</u>	128
<u>4.10.6 Détruire une référence</u>	128
<u>4.10.7 Repérer une référence</u>	128

5 Sécurité.....130

<u>5.1 Sécurité</u>	130
<u>5.1.1 Considérations générales</u>	130
<u>5.1.2 Binaires CGI</u>	131
<u>5.1.3 Module Apache</u>	133
<u>5.1.4 Sécurité des fichiers</u>	134
<u>5.1.5 Sécurité des bases de données</u>	135
<u>5.1.6 Rapport d'erreurs</u>	140
<u>5.1.7 Utilisation des variables super-globales</u>	141
<u>5.1.8 Données transmises par les internautes</u>	142
<u>5.1.9 Masquer PHP</u>	142
<u>5.1.10 Etre à jour</u>	143

6 Caractéristiques.....144

<u>6.1 Identification HTTP avec PHP</u>	144
<u>6.2 Cookies</u>	145
<u>6.3 Gestion des chargements de fichier</u>	146
<u>6.3.1 Chargements de fichiers par méthode POST</u>	146
<u>6.3.2 Explication sur les messages d'erreurs de chargement de fichiers</u>	148
<u>6.3.3 Erreurs classiques</u>	148
<u>6.3.4 Télécharger plusieurs fichiers simultanément</u>	148
<u>6.3.5 Chargement par méthode PUT</u>	149
<u>6.4 Utilisation des fichiers à distance</u>	150
<u>6.5 Gestion des connexions</u>	151
<u>6.6 Connexions persistantes aux bases de données</u>	152

Sommaire

6 Caractéristiques

<u>6.7 Safe mode</u>	153
<u>6.7.1 Sécurité et Safe Mode</u>	153
<u>6.7.2 Fonctions désactivées par le Safe Mode</u>	156
<u>6.8 Utiliser PHP en ligne de commande</u>	158

7 Index des fonctions.....165

<u>7.1 Apache</u>	165
<u>7.1.1 Introduction</u>	165
<u>7.1.2 Installation</u>	165
<u>7.1.3 Configuration à l'exécution</u>	165
<u>7.1.4 Types de ressources</u>	165
<u>7.1.5 Constantes prédefinies</u>	165
<u>7.1.7 apache_lookup_uri()</u> : Effectue une requête partielle pour l'URI spécifiée et renvoie toutes les informations.....	166
<u>7.1.8 apache_note()</u> : Affiche ou affecte le paramètre "apache request notes".....	166
<u>7.1.9 apache_request_headers()</u> : Lit tous les en-têtes HTTP de la requête.....	166
<u>7.1.10 apache_response_headers()</u> : Lit tous les en-têtes de réponse HTTP.....	167
<u>7.1.11 apache_setenv()</u> : Modifie une variable de subprocess_env Apache.....	167
<u>7.1.12 ascii2ebcdic()</u> : Transforme une chaîne ASCII en EBCDIC.....	167
<u>7.1.13 ebcdic2ascii()</u> : Transforme une chaîne EBCDIC en ASCII.....	168
<u>7.1.14 getallheaders()</u> : Lit tous les en-têtes de la requête HTTP.....	168
<u>7.1.15 virtual()</u> : Effectue une sous-requête Apache.....	168
<u>7.2 Tableaux</u>	169
<u>7.2.1 Introduction</u>	169
<u>7.2.2 Pré-requis</u>	169
<u>7.2.3 Installation</u>	169
<u>7.2.4 Configuration à l'exécution</u>	169
<u>7.2.5 Types de ressources</u>	169
<u>7.2.6 Constantes prédefinies</u>	169
<u>7.2.7 Voir aussi</u>	170
<u>7.2.9 array_chunk()</u> : Sépare un tableau en tableaux de taille inférieure.....	171
<u>7.2.10 array_combine()</u> : Crée un tableau à partir de deux autres tableaux.....	172
<u>7.2.11 array_count_values()</u> : Compte le nombre de valeurs dans un tableau.....	172
<u>7.2.12 array_diff_assoc()</u> : Calcule la différence de deux tableaux, en prenant en compte les clés.....	173
<u>7.2.13 array_diff()</u> : Calcule la différence entre deux tableaux.....	173
<u>7.2.14 array_fill()</u> : Remplis un tableau avec une même valeur.....	174
<u>7.2.15 array_filter()</u> : Filtre les éléments d'un tableau.....	174
<u>7.2.16 array_flip()</u> : Remplace les clés par les valeurs, et les valeurs par les clés.....	175
<u>7.2.17 array_intersect_assoc()</u> : Calcule l'intersection de deux tableaux avec des tests sur les index.....	176
<u>7.2.18 array_intersect()</u> : Calcule l'intersection de tableaux.....	176
<u>7.2.19 array_key_exists()</u> : Vérifie si une clé existe dans un tableau.....	177
<u>7.2.20 array_keys()</u> : Retourne toutes les clés d'un tableau.....	177
<u>7.2.21 array_map()</u> : Applique sur fonction sur des tableaux.....	178
<u>7.2.22 array_merge_recursive()</u> : Combine plusieurs tableaux ensembles, récursivement.....	180
<u>7.2.23 array_merge()</u> : Fusionne plusieurs tableaux.....	181
<u>7.2.24 array_multisort()</u> : Tri multi-dimensionnel.....	182
<u>7.2.25 array_pad()</u> : Complète un tableau jusqu'à la longueur spécifiée, avec une valeur.....	183

Sommaire

7 Index des fonctions

<u>7.2.26 array_pop()</u> : Dépile un élément de la fin d'un tableau.....	183
<u>7.2.27 array_push()</u> : Empile un ou plusieurs éléments à la fin d'un tableau.....	184
<u>7.2.28 array_rand()</u> : Prend une ou plusieurs valeurs, au hasard dans un tableau.....	184
<u>7.2.29 array_reduce()</u> : Réduit itérativement un tableau.....	185
<u>7.2.30 array_reverse()</u> : Inverse l'ordre des éléments d'un tableau.....	185
<u>7.2.31 array_search()</u> : Recherche dans un tableau la clé associée à une valeur.....	186
<u>7.2.32 array_shift()</u> : Dépile un élément au début d'un tableau.....	186
<u>7.2.33 array_slice()</u> : Extrait une portion de tableau.....	187
<u>7.2.34 array_splice()</u> : Efface et remplace une portion de tableau.....	187
<u>7.2.35 array_sum()</u> : Calcule la somme des valeurs du tableau.....	188
<u>7.2.36 array_unique()</u> : Dédoublonne un tableau.....	188
<u>7.2.37 array_unshift()</u> : Empile un ou plusieurs éléments au début d'un tableau.....	189
<u>7.2.38 array_values()</u> : Retourne les valeurs d'un tableau.....	190
<u>7.2.39 array_walk()</u> : Exécute une fonction sur chacun des membres d'un tableau.....	190
<u>7.2.40 array()</u> : Crée un tableau.....	191
<u>7.2.41 arsort()</u> : Trie un tableau en ordre inverse.....	192
<u>7.2.42 asort()</u> : Trie un tableau en ordre.....	193
<u>7.2.43 compact()</u> : Crée un tableau contenant les variables et leur valeur.....	193
<u>7.2.44 count()</u> : Compte le nombre d'éléments d'une variable.....	194
<u>7.2.45 current()</u> : Transforme une variable en tableau.....	195
<u>7.2.46 each()</u> : Retourne chaque paire clé/valeur d'un tableau.....	195
<u>7.2.47 end()</u> : Positionne le pointeur de tableau en fin de tableau.....	196
<u>7.2.48 extract()</u> : Importe les variables dans la table des symboles.....	196
<u>7.2.49 in_array()</u> : Indique si une valeur appartient à un tableau.....	198
<u>7.2.50 key()</u> : Retourne une clé d'un tableau associatif.....	199
<u>7.2.51 krsort()</u> : Trie un tableau en sens inverse et suivant les clés.....	199
<u>7.2.52 ksort()</u> : Trie un tableau suivant les clés.....	199
<u>7.2.53 list()</u> : Transforme une liste de variables en tableau.....	200
<u>7.2.54 natcasesort()</u> : Tri d'un tableau avec l'algorithme à "ordre naturel" insensible à la casse.....	201
<u>7.2.55 natsort()</u> : Tri d'un tableau avec l'algorithme à "ordre naturel".....	201
<u>7.2.56 next()</u> : Avance le pointeur interne d'un tableau.....	202
<u>7.2.57 pos()</u> : Retourne l'élément courant d'un tableau.....	203
<u>7.2.58 prev()</u> : Recule le pointeur courant de tableau.....	203
<u>7.2.59 range()</u> : Crée un tableau contenant un intervalle d'éléments.....	203
<u>7.2.60 reset()</u> : Remet le pointeur interne de tableau au début.....	204
<u>7.2.61 rsort()</u> : Trie en ordre inverse.....	204
<u>7.2.62 shuffle()</u> : Mélange les éléments d'un tableau.....	205
<u>7.2.63 sizeof()</u> : Alias de count.....	205
<u>7.2.64 sort()</u> : Trie un tableau.....	205
<u>7.2.65 uasort()</u> : Trie un tableau en utilisant une fonction de comparaison personnalisée.....	206
<u>7.2.66 uksort()</u> : Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur.....	206
<u>7.2.67 usort()</u> : Trie un tableau en utilisant une fonction de comparaison personnalisée.....	206
<u>7.3 Aspell [Obs.]</u>	209
<u>7.3.1 Introduction</u>	209
<u>7.3.2 Pré-requis</u>	209
<u>7.3.3 Installation</u>	209
<u>7.3.4 Voir aussi</u>	209

Sommaire

7 Index des fonctions

7.3.6 <u>aspell_check()</u> : Vérifie un mot [Obsolète].....	209
7.3.7 <u>aspell_new()</u> : Charge un nouveau dictionnaire [Obsolète].....	210
7.3.8 <u>aspell_suggest()</u> : Suggère l'orthographe d'un mot [Obsolète].....	210
7.4 <u>Nombres de grande taille BCMath</u>	211
7.4.1 <u>Introduction</u>	211
7.4.2 <u>Pré-requis</u>	211
7.4.3 <u>Installation</u>	211
7.4.4 <u>Configuration à l'exécution</u>	211
7.4.5 <u>Types de ressources</u>	211
7.4.6 <u>Constantes prédefinies</u>	211
7.4.8 <u>bccomp()</u> : Compare deux nombres de grande taille.....	212
7.4.9 <u>bcdiv()</u> : Divise deux nombres de grande taille.....	212
7.4.10 <u>bcmod()</u> : Retourne le reste d'une division entre nombres de grande taille.....	212
7.4.11 <u>bcmul()</u> : Multiplie deux nombres de grande taille.....	212
7.4.12 <u>bcpow()</u> : Elève un nombre à la puissance n-ième.....	213
7.4.13 <u>bcpowmod()</u> : Calcule le reste modulo d'un nombre élevé à une puissance.....	213
7.4.14 <u>bcscale()</u> : Détermine le nombre de décimales par défaut.....	213
7.4.15 <u>bcsqrt()</u> : Renvoie la racine carrée d'un nombre de grande taille.....	213
7.4.16 <u>bcsub()</u> : Soustrait un nombre de grande taille à un autre.....	214
7.5 <u>Compression Bzip2</u>	215
7.5.1 <u>Introduction</u>	215
7.5.2 <u>Pré-requis</u>	215
7.5.3 <u>Installation</u>	215
7.5.4 <u>Configuration à l'exécution</u>	215
7.5.5 <u>Types de ressources</u>	215
7.5.6 <u>Constantes prédefinies</u>	215
7.5.7 <u>Exemples</u>	215
7.5.9 <u>bzcompress()</u> : Comprime une chaîne avec bzip2.....	216
7.5.10 <u>bzdecompress()</u> : Décomprime une chaîne bzip2.....	216
7.5.11 <u>bzerrno()</u> : Retourne le numéro d'erreur bzip2.....	217
7.5.12 <u>bzerror()</u> : Retourne le numéro et le message d'erreur bzip2 dans un tableau.....	217
7.5.13 <u>bzerrstr()</u> : Retourne le message d'erreur bzip2.....	217
7.5.14 <u>bzflush()</u> : Force l'écriture de toutes les données compressées.....	217
7.5.15 <u>bzopen()</u> : Ouvre un fichier compressé avec bzip2.....	218
7.5.16 <u>bzread()</u> : Lecture binaire d'un fichier bzip2.....	218
7.5.17 <u>bzwrite()</u> : Ecriture binaire dans un fichier bzip2.....	218
7.6 <u>Calendrier</u>	220
7.6.1 <u>Introduction</u>	220
7.6.2 <u>Installation</u>	220
7.6.3 <u>Configuration à l'exécution</u>	220
7.6.4 <u>Types de ressources</u>	220
7.6.5 <u>Constantes prédefinies</u>	220
7.6.7 <u>cal_from_jd()</u> : Convertit le nombre de jours Julien en un calendrier et retourne des détails.....	221
7.6.8 <u>cal_info()</u> : Retourne des détails sur un calendrier.....	221
7.6.9 <u>cal_to_jd()</u> : Convertit un calendrier en nombre de jours Julien.....	222
7.6.10 <u>easter_date()</u> : Retourne un timestamp UNIX pour Pâques, à minuit.....	222
7.6.11 <u>easter_days()</u> : Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée.....	222
7.6.12 <u>frenchtojd()</u> : Convertit une date du calendrier français républicain en	

Sommaire

7 Index des fonctions

<u>nombre de jours du calendrier Julien.....</u>	223
<u>7.6.13 gregoriantojd() : Convertit une date grégorienne en nombre de jours du</u>	
<u>calendrier Julien.....</u>	223
<u>7.6.14 jddayofweek() : Retourne le numéro du jour de la semaine.....</u>	224
<u>7.6.15 jdmonthname() : Retourne le nom du mois.....</u>	224
<u>7.6.16 jdto french() : Convertit le nombre de jours du calendrier Julien en date du</u>	
<u>calendrier français républicain.....</u>	224
<u>7.6.17 jdtogregorian() : Convertit le nombre de jours du calendrier Julien en date</u>	
<u>grégorienne.....</u>	224
<u>7.6.18 jdto jewish() : Convertit le nombre de jours du calendrier Julien en date du</u>	
<u>calendrier juif.....</u>	225
<u>7.6.19 jdtojulian() : Convertit le nombre de jours du calendrier Julien en date du</u>	
<u>calendrier Julien.....</u>	225
<u>7.6.20 jdtounix() : Convertit un nombre de jours Julien en timestamp UNIX.....</u>	225
<u>7.6.21 jewishtojd() : Convertit une date du calendrier juif en nombre de jours du</u>	
<u>calendrier Julien.....</u>	225
<u>7.6.22 juliantojd() : Convertit le nombre de jours du calendrier Julien en date du</u>	
<u>calendrier Julien.....</u>	226
<u>7.6.23 unixtojd() : Convertit un timestamp UNIX en nombre de jours Julien.....</u>	226
<u>7.7 Paiement C CVS.....</u>	227
<u>7.7.1 Introduction.....</u>	227
<u>7.7.2 Installation.....</u>	227
<u>7.7.3 Voir aussi.....</u>	227
<u>7.7.5 ccvs_auth() : Perform credit authorization test on a transaction.....</u>	228
<u>7.7.6 ccvs_command() : Performs a command which is peculiar to a single</u>	
<u>protocol, and thus is not available in the general C CVS API.....</u>	228
<u>7.7.7 ccvs_count() : Find out how many transactions of a given type are stored in</u>	
<u>the system.....</u>	228
<u>7.7.8 ccvs_delete() : Delete a transaction.....</u>	228
<u>7.7.9 ccvs_done() : Terminate C CVS engine and do cleanup work.....</u>	228
<u>7.7.10 ccvs_init() : Initialize C CVS for use.....</u>	228
<u>7.7.11 ccvs_lookup() : Look up an item of a particular type in the database #.....</u>	229
<u>7.7.12 ccvs_new() : Create a new, blank transaction.....</u>	229
<u>7.7.13 ccvs_report() : Return the status of the background communication process.....</u>	229
<u>7.7.14 ccvs_return() : Transfer funds from the merchant to the credit card holder.....</u>	229
<u>7.7.15 ccvs_reverse() : Perform a full reversal on an already-processed</u>	
<u>authorization.....</u>	229
<u>7.7.16 ccvs_sale() : Transfer funds from the credit card holder to the merchant.....</u>	229
<u>7.7.17 ccvs_status() : Check the status of an invoice.....</u>	230
<u>7.7.18 ccvs_textvalue() : Get text return value for previous function call.....</u>	230
<u>7.7.19 ccvs_void() : Perform a full reversal on a completed transaction.....</u>	230
<u>7.8 Support COM pour Windows.....</u>	231
<u>7.8.1 Introduction.....</u>	231
<u>7.8.2 Pré-requis.....</u>	231
<u>7.8.3 Installation.....</u>	231
<u>7.8.4 Configuration à l'exécution.....</u>	231
<u>7.8.5 Constantes prédefinies.....</u>	231
<u>7.8.6 Voir aussi.....</u>	232
<u>7.8.8 com_get() : Lit la valeur d'un propriété d'un composant COM.....</u>	233
<u>7.8.9 com_invoke() : Appelle une méthode d'un composant COM.....</u>	233
<u>7.8.10 com_isenum() : Récupère un IEnumVariant.....</u>	233

Sommaire

7 Index des fonctions

<u>7.8.11 com_load_typelib()</u> : Charge une librairie de type (Typelib).....	233
<u>7.8.12 com_load()</u> : Crée une référence sur un composant COM.....	233
<u>7.8.13 com_propget()</u> : Lit la valeur d'un propriété d'un composant COM.....	233
<u>7.8.14 com_propput()</u> : Modifie une propriété d'un composant COM.....	233
<u>7.8.15 com_propset()</u> : Modifie une propriété d'un composant COM.....	234
<u>7.8.16 com_release()</u> : Décrémente le compteur de références.....	234
<u>7.8.17 com_set()</u> : Modifie une propriété d'un composant COM.....	234
<u>7.9 Classes et Objets</u>	235
<u>7.9.1 Introduction</u>	235
<u>7.9.2 Pré-requis</u>	235
<u>7.9.3 Installation</u>	235
<u>7.9.4 Configuration à l'exécution</u>	235
<u>7.9.5 Types de ressources</u>	235
<u>7.9.6 Constantes prédefinies</u>	235
<u>7.9.7 Exemples</u>	235
<u>7.9.9 call_user_method()</u> : Appelle une méthode utilisateur d'un objet.....	237
<u>7.9.10 class_exists()</u> : Vérifie qu'une classe a été définie.....	238
<u>7.9.11 get_class_methods()</u> : Retourne les noms des méthodes d'une classe.....	238
<u>7.9.12 get_class_vars()</u> : Retourne les valeurs par défaut des attributs d'une classe.....	238
<u>7.9.13 get_class()</u> : Retourne la classe d'un objet.....	239
<u>7.9.14 get_declared_classes()</u> : Liste toutes les classes définies.....	240
<u>7.9.15 get_object_vars()</u> : Retourne un tableau associatif des propriétés d'un objet.....	240
<u>7.9.16 get_parent_class()</u> : Retourne le nom de la classe d'un objet.....	241
<u>7.9.17 is_a()</u> : Retourne TRUE si un objet a pour parent une classe donnée.....	242
<u>7.9.18 is_subclass_of()</u> : Détermine si un objet est une sous-classe.....	242
<u>7.9.19 method_exists()</u> : Vérifie que la méthode existe pour une classe.....	242
<u>7.10 ClibPDF</u>	243
<u>7.10.1 Introduction</u>	243
<u>7.10.2 Pré-requis</u>	243
<u>7.10.3 Installation</u>	243
<u>7.10.4 Configuration à l'exécution</u>	244
<u>7.10.5 Constantes prédefinies</u>	244
<u>7.10.6 Exemples</u>	244
<u>7.10.7 Voir aussi</u>	245
<u>7.10.9 cpdf_add_outline()</u> : Ajoute un signet à la page courante.....	247
<u>7.10.10 cpdf_arc()</u> : Dessine un arc de cercle.....	247
<u>7.10.11 cpdf_begin_text()</u> : Démarre une section de texte.....	247
<u>7.10.12 cpdf_circle()</u> : Dessine un cercle.....	248
<u>7.10.13 cpdf_clip()</u> : Aligne les dessins sur le chemin courant.....	248
<u>7.10.14 cpdf_close()</u> : Ferme un fichier PDF.....	248
<u>7.10.15 cpdf_closepath_fill_stroke()</u> : Remplit le chemin, dessine le bord et ferme le chemin.....	248
<u>7.10.16 cpdf_closepath_stroke()</u> : Ferme le fichier et dessine une ligne le long du chemin.....	248
<u>7.10.17 cpdf_closepath()</u> : Ferme le chemin.....	249
<u>7.10.18 cpdf_continue_text()</u> : Imprime le texte à la ligne suivante.....	249
<u>7.10.19 cpdf_curveto()</u> : Dessine une courbe.....	249
<u>7.10.20 cpdf_end_text()</u> : Termine une section de texte.....	249
<u>7.10.21 cpdf_fill_stroke()</u> : Remplit le chemin, et dessine le bord.....	249

Sommaire

7 Index des fonctions

<u>7.10.22 cpdf_fill()</u> : Remplit le chemin courant.....	250
<u>7.10.23 cpdf_finalize_page()</u> : Termine une page.....	250
<u>7.10.24 cpdf_finalize()</u> : Termine un document.....	250
<u>7.10.25 cpdf_global_set_document_limits()</u> : Fixe les limites d'un document PDF.....	250
<u>7.10.26 cpdf_import_jpeg()</u> : Ouvre une image JPEG.....	250
<u>7.10.27 cpdf_lineto()</u> : Dessine une ligne.....	251
<u>7.10.28 cpdf_moveto()</u> : Fixe le point courant.....	251
<u>7.10.29 cpdf_newpath()</u> : Commence un nouveau chemin.....	251
<u>7.10.30 cpdf_open()</u> : Ouvre un nouveau document PDF.....	251
<u>7.10.31 cpdf_output_buffer()</u> : Affiche le document PDF qui est en mémoire.....	252
<u>7.10.32 cpdf_page_init()</u> : Commence une nouvelle page.....	252
<u>7.10.33 cpdf_place_inline_image()</u> : Place une image dans une page.....	252
<u>7.10.34 cpdf_rect()</u> : Dessine un rectangle.....	252
<u>7.10.35 cpdf_restore()</u> : Restaure un environnement.....	253
<u>7.10.36 cpdf_rlineto()</u> : Dessine une ligne, relativement.....	253
<u>7.10.37 cpdf_rmoveto()</u> : Fixe le point courant relativement.....	253
<u>7.10.38 cpdf_rotate_text()</u> : Configure l'inclinaison d'un texte.....	253
<u>7.10.39 cpdf_rotate()</u> : Effectue une rotation.....	254
<u>7.10.40 cpdf_save_to_file()</u> : Ecris un document PDF dans un fichier.....	254
<u>7.10.41 cpdf_save()</u> : Sauve l'environnement courant.....	254
<u>7.10.42 cpdf_scale()</u> : Modifie l'échelle.....	254
<u>7.10.43 cpdf_set_action_url()</u> : Paramètre un hyper lien.....	254
<u>7.10.44 cpdf_set_char_spacing()</u> : Fixe l'espacement des caractères.....	254
<u>7.10.45 cpdf_set_creator()</u> : Fixe le créateur d'un document PDF.....	255
<u>7.10.46 cpdf_set_current_page()</u> : Fixe la page courante.....	255
<u>7.10.47 cpdf_set_font_directories()</u> : Précise les dossiers à fouiller pour rechercher des polices externes.....	255
<u>7.10.48 cpdf_set_font_map_file()</u> : Configure le fichier de remplacement de polices externes.....	255
<u>7.10.49 cpdf_set_font()</u> : Sélectionne la police courante et sa taille.....	255
<u>7.10.50 cpdf_set_horiz_scaling()</u> : Fixe l'échelle horizontale du texte.....	256
<u>7.10.51 cpdf_set_keywords()</u> : Fixe les mot clés d'un document PDF.....	256
<u>7.10.52 cpdf_set_leading()</u> : Fixe la distance entre deux lignes.....	256
<u>7.10.53 cpdf_set_page_animation()</u> : Fixe l'animation de la transition entre les pages.....	256
<u>7.10.54 cpdf_set_subject()</u> : Fixe le sujet d'un document PDF.....	256
<u>7.10.55 cpdf_set_text_matrix()</u> : Fixe la matrice du texte.....	257
<u>7.10.56 cpdf_set_text_pos()</u> : Fixe la position du texte.....	257
<u>7.10.57 cpdf_set_text_rendering()</u> : Détermine le rendu du texte.....	257
<u>7.10.58 cpdf_set_text_rise()</u> : Fixe l'élévation du texte.....	257
<u>7.10.59 cpdf_set_title()</u> : Fixe le titre d'un document PDF.....	257
<u>7.10.60 cpdf_set_viewer_preferences()</u> : Indique comment afficher le document à l'utilisateur.....	257
<u>7.10.61 cpdf_set_word_spacing()</u> : Fixe l'espacement des mots.....	258
<u>7.10.62 cpdf_setdash()</u> : Fixe le motif de pointillé.....	258
<u>7.10.63 cpdf_setflat()</u> : Fixe la platitude (flatness).....	258
<u>7.10.64 cpdf_setgray_fill()</u> : Modifie le niveau de gris comme couleur de remplissage.....	258
<u>7.10.65 cpdf_setgray_stroke()</u> : Choisit un niveau de gris comme couleur de dessin.....	258
<u>7.10.66 cpdf_setgray()</u> : Modifie un niveau de gris comme couleur de dessin et de	

Sommaire

7 Index des fonctions

<u>remplissage</u>	258
<u>7.10.67 cpdf_setlinecap()</u> : Fixe le paramètre linecap.....	259
<u>7.10.68 cpdf_setlinejoin()</u> : Fixe le paramètre linejoin.....	259
<u>7.10.69 cpdf_setlinewidth()</u> : Fixe la largeur de ligne.....	259
<u>7.10.70 cpdf_setmiterlimit()</u> : Fixe le paramètre miter limit.....	259
<u>7.10.71 cpdf_setrgbcolor_fill()</u> : Choisit une couleur rgb comme couleur de remplissage.....	259
<u>7.10.72 cpdf_setrgbcolor_stroke()</u> : Choisit une couleur rgb comme couleur de dessin.....	259
<u>7.10.73 cpdf_setrgbcolor()</u> : Choisit une couleur rgb comme couleur de dessin et de remplissage.....	260
<u>7.10.74 cpdf_show_xy()</u> : Affiche un texte à une position.....	260
<u>7.10.75 cpdf_show()</u> : Imprime un texte à la position courante.....	260
<u>7.10.76 cpdf_stringwidth()</u> : Retourne la taille de la chaîne.....	260
<u>7.10.77 cpdf_stroke()</u> : Dessine une ligne le long du chemin.....	260
<u>7.10.78 cpdf_text()</u> : Imprime un texte avec des options.....	261
<u>7.10.79 cpdf_translate()</u> : Modifie l'origine du système de coordonnées.....	261
<u>7.11 Crack</u>	262
<u>7.11.1 Introduction</u>	262
<u>7.11.2 Pré-requis</u>	262
<u>7.11.3 Installation</u>	262
<u>7.11.4 Configuration à l'exécution</u>	262
<u>7.11.5 Types de ressources</u>	262
<u>7.11.6 Constantes prédefinies</u>	262
<u>7.11.7 Exemples</u>	262
<u>7.11.9 crack_closedict()</u> : Referme le dictionnaire CrackLib.....	263
<u>7.11.10 crack_getlastmessage()</u> : Retourne le message de diagnostic.....	263
<u>7.11.11 crack_opendict()</u> : Ouvre un nouveau dictionnaire CrackLib dictionary.....	264
<u>7.12 CURL</u>	265
<u>7.12.1 Introduction</u>	265
<u>7.12.2 Pré-requis</u>	265
<u>7.12.3 Installation</u>	265
<u>7.12.4 Constantes prédefinies</u>	265
<u>7.12.5 Exemples</u>	268
<u>7.12.7 curl_errno()</u> : Retourne un entier représentant le dernier message d'erreur CURL.....	269
<u>7.12.8 curl_error()</u> : Retourne une chaîne contenant le dernier message d'erreur CURL.....	269
<u>7.12.9 curl_exec()</u> : Exécute une session CURL.....	269
<u>7.12.10 curl_getinfo()</u> : Lit les informations détaillant un transfert CURL.....	269
<u>7.12.11 curl_init()</u> : Initialise une session CURL.....	270
<u>7.12.12 curl_multi_add_handle()</u> : Ajoute une ressource cURL à un cURL multiple.....	271
<u>7.12.13 curl_multi_close()</u> : Termine un jeu de transferts cURL.....	271
<u>7.12.14 curl_multi_exec()</u> : Exécute les sous-requêtes d'un cURL.....	271
<u>7.12.15 curl_multi_getcontent()</u> : Retourne le contenu obtenu avec l'option CURLOPT_RETURNTRANSFER.....	271
<u>7.12.16 curl_multi_info_read()</u> : Lit les informations sur les transferts actuels.....	271
<u>7.12.17 curl_multi_init()</u> : Retourne un nouveau cURL multiple.....	272
<u>7.12.18 curl_multi_remove_handle()</u> : Retire un cURL multiple d'un jeu de cURL.....	272
<u>7.12.19 curl_multi_select()</u> : Retourne les sockets associées à CURL, pour utilisation.....	272

Sommaire

7 Index des fonctions

7.12.20 <u>curl_setopt()</u> : Modifie une option de transfert CURL.....	272
7.12.21 <u>curl_version()</u> : Retourne la version courante de CURL.....	274
7.13 <u>Paiement Cybercash</u>	275
7.13.1 <u>Installation</u>	275
7.13.3 <u>cybercash_base64_encode()</u> :.....	275
7.13.4 <u>cybercash_decr()</u> :.....	275
7.13.5 <u>cybercash_encr()</u> :.....	275
7.14 <u>Cyrus IMAP</u>	276
7.14.1 <u>Introduction</u>	276
7.14.2 <u>Installation</u>	276
7.14.3 <u>Constantes prédefinies</u>	276
7.14.5 <u>cyrus_bind()</u> : Ajoute une fonction de callback sur une connexion Cyrus IMAP.....	276
7.14.6 <u>cyrus_close()</u> : Ferme la connexion à un serveur Cyrus IMAP.....	276
7.14.7 <u>cyrus_connect()</u> : Se connecte à un serveur Cyrus IMAP.....	277
7.14.8 <u>cyrus_query()</u> : Envoie une requête à un serveur Cyrus IMAP.....	277
7.14.9 <u>cyrus_unbind()</u> : Supprime une fonction de callback sur une connexion Cyrus IMAP.....	277
7.15 <u>Caractères</u>	278
7.15.1 <u>Introduction</u>	278
7.15.2 <u>Pré-requis</u>	278
7.15.3 <u>Installation</u>	278
7.15.4 <u>Configuration à l'exécution</u>	278
7.15.5 <u>Types de ressources</u>	278
7.15.6 <u>Constantes prédefinies</u>	278
7.15.8 <u>ctype_alpha()</u> : Vérifie qu'une chaîne est alphabétique.....	279
7.15.9 <u>ctype_cntrl()</u> : Vérifie qu'un caractère est un caractère de contrôle.....	279
7.15.10 <u>ctype_digit()</u> : Vérifie qu'une chaîne est un entier.....	279
7.15.11 <u>ctype_graph()</u> : Vérifie qu'une chaîne est imprimable.....	279
7.15.12 <u>ctype_lower()</u> : Vérifie qu'une chaîne est en minuscules.....	280
7.15.13 <u>ctype_print()</u> : Vérifie qu'une chaîne est imprimable.....	280
7.15.14 <u>ctype_punct()</u> : Vérifie qu'une chaîne contient de la ponctuation.....	280
7.15.15 <u>ctype_space()</u> : Vérifie qu'une chaîne est faite caractères blancs.....	280
7.15.16 <u>ctype_upper()</u> : Vérifie qu'une chaîne est en majuscules.....	280
7.15.17 <u>ctype_xdigit()</u> : Vérifie qu'un caractère représente un nombre hexadécimal.....	280
7.16 <u>DBA</u>	282
7.16.1 <u>Introduction</u>	282
7.16.2 <u>Pré-requis</u>	282
7.16.3 <u>Installation</u>	282
7.16.4 <u>Configuration à l'exécution</u>	283
7.16.5 <u>Types de ressources</u>	283
7.16.6 <u>Constantes prédefinies</u>	283
7.16.7 <u>Exemples</u>	283
7.16.9 <u>dba_delete()</u> : Efface une entrée.....	284
7.16.10 <u>dba_exists()</u> : Vérifie qu'une clé existe.....	285
7.16.11 <u>dba_fetch()</u> : Lit les données liées à une clé.....	285
7.16.12 <u>dba_firstkey()</u> : Lit la première clé.....	285
7.16.13 <u>dba_handlers()</u> : Liste les gestionnaires dba disponibles.....	286
7.16.14 <u>dba_insert()</u> : Insère une entrée.....	286
7.16.15 <u>dba_list()</u> : Liste tous les fichiers de bases de données ouverts.....	286

Sommaire

7 Index des fonctions

<u>7.16.16 dba_nextkey()</u> : Lit la clé suivante.....	286
<u>7.16.17 dba_open()</u> : Ouvre une base de données.....	286
<u>7.16.18 dba_optimize()</u> : Optimise une base.....	287
<u>7.16.19 dba_popen()</u> : Ouvre une connexion persistante à une base de données dba.....	287
<u>7.16.20 dba_replace()</u> : Remplace ou insère une entrée.....	288
<u>7.16.21 dba_sync()</u> : Synchronise une base de données.....	288
<u>7.17 Dates et heures</u>	289
<u>7.17.1 Introduction</u>	289
<u>7.17.2 Pré-requis</u>	289
<u>7.17.3 Installation</u>	289
<u>7.17.4 Configuration à l'exécution</u>	289
<u>7.17.5 Types de ressources</u>	289
<u>7.17.6 Constantes prédefinies</u>	289
<u>7.17.8 date()</u> : Formate une date/heure locale.....	290
<u>7.17.9 getdate()</u> : Retourne la date/heure.....	292
<u>7.17.10 gettimeofday()</u> : Retourne l'heure actuelle.....	292
<u>7.17.11 gmdate()</u> : Formate une date/heure GMT/CUT.....	292
<u>7.17.12 gmmktime()</u> : Retourne le timestamp UNIX d'une date GMT.....	293
<u>7.17.13 gmstrftime()</u> : Formate une date/heure GMT/CUT en fonction des paramétrages locaux.....	293
<u>7.17.14 localtime()</u> : Lit l'heure locale.....	293
<u>7.17.15 microtime()</u> : Retourne le timestamp UNIX actuel avec microsecondes.....	294
<u>7.17.16 mktime()</u> : Retourne le timestamp UNIX d'une date.....	294
<u>7.17.17 strftime()</u> : Formate une date/heure locale avec les options locales.....	295
<u>7.17.18 strtotime()</u> : Transforme un texte anglais en timestamp.....	297
<u>7.17.19 time()</u> : Retourne le timestamp UNIX actuel.....	298
<u>7.18 dBase</u>	299
<u>7.18.1 Introduction</u>	299
<u>7.18.2 Installation</u>	299
<u>7.18.3 Configuration à l'exécution</u>	299
<u>7.18.4 Types de ressources</u>	299
<u>7.18.5 Constantes prédefinies</u>	299
<u>7.18.7 dbase_close()</u> : Ferme une base dBase.....	300
<u>7.18.8 dbase_create()</u> : Crée une base de données dBase.....	300
<u>7.18.9 dbase_delete_record()</u> : Efface un enregistrement dans une base dBase.....	301
<u>7.18.10 dbase_get_header_info()</u> : Lit les informations d'entête d'une base dBase.....	301
<u>7.18.11 dbase_get_record_with_names()</u> : Lit un enregistrement dans une base, sous la forme d'un tableau associatif.....	302
<u>7.18.12 dbase_get_record()</u> : Lit un enregistrement dans une base dBase.....	302
<u>7.18.13 dbase_numfields()</u> : Compte le nombre de champs d'une base dBase.....	302
<u>7.18.14 dbase_numrecords()</u> : Compte le nombre d'enregistrements dans une base dBase.....	302
<u>7.18.15 dbase_open()</u> : Ouvre une base dBase.....	303
<u>7.18.16 dbase_pack()</u> : Compacte une base dBase.....	303
<u>7.18.17 dbase_replace_record()</u> : Remplace un enregistrement dans une base dBase.....	303
<u>7.19 DBM (Obsolètes)</u>	304
<u>7.19.1 Introduction</u>	304
<u>7.19.2 Pré-requis</u>	304
<u>7.19.3 Installation</u>	304

Sommaire

7 Index des fonctions

<u>7.19.4 Configuration à l'exécution</u>	304
<u>7.19.5 Types de ressources</u>	304
<u>7.19.6 Constantes prédefinies</u>	304
<u>7.19.7 Exemples</u>	304
<u>7.19.9 dbmclose() : Ferme une base de données dbm</u>	305
<u>7.19.10 dbmdelete() : Efface une valeur</u>	305
<u>7.19.11 dbmexists() : Indique si une valeur existe</u>	305
<u>7.19.12 dbmfetch() : Lit une valeur</u>	305
<u>7.19.13 dbmfirstkey() : Lit la première clé</u>	306
<u>7.19.14 dbminsert() : Insère une valeur</u>	306
<u>7.19.15 dbmnextkey() : Lit la clé suivante</u>	306
<u>7.19.16 dbmopen() : Ouvre une base de données dbm</u>	306
<u>7.19.17 dbmreplace() : Remplace une valeur</u>	307
<u>7.20 dbx</u>	308
<u>7.20.1 Introduction</u>	308
<u>7.20.2 Pré-requis</u>	308
<u>7.20.3 Installation</u>	308
<u>7.20.4 Configuration à l'exécution</u>	308
<u>7.20.5 Types de ressources</u>	309
<u>7.20.6 Constantes prédefinies</u>	309
<u>7.20.8 dbx_compare() : Compare deux lignes afin de les trier</u>	310
<u>7.20.9 dbx_connect() : Ouvre une connexion à une base de données</u>	310
<u>7.20.10 dbx_error() : Rapporte le message d'erreur du dernier appel de fonction</u>	311
<u>7.20.11 dbx_escape_string() : Protège une chaîne de caractères pour l'utiliser dans une requête</u>	312
<u>7.20.12 dbx_fetch_row() : Lit une ligne dans un résultat DBX ayant l'option DBX_RESULT_UNBUFFERED activée</u>	312
<u>7.20.13 dbx_query() : Envoie une requête et lit tous les résultats</u>	313
<u>7.20.14 dbx_sort() : Tri un résultat avec une fonction utilisateur</u>	315
<u>7.21 DB++</u>	316
<u>7.21.1 Introduction</u>	316
<u>7.21.2 Pré-requis</u>	316
<u>7.21.3 Installation</u>	316
<u>7.21.4 Configuration à l'exécution</u>	316
<u>7.21.5 Types de ressources</u>	316
<u>7.21.6 Constantes prédefinies</u>	317
<u>7.21.8 dbplus_aql() : Exécute une requête AQL</u>	318
<u>7.21.9 dbplus_chdir() : Lit/modifie le chemin courant de base</u>	319
<u>7.21.10 dbplus_close() : Ferme une relation</u>	319
<u>7.21.11 dbplus_curr() : Lit la ligne courante dans une relation</u>	319
<u>7.21.12 dbplus_errcode() : Obtenir un message d'erreur</u>	320
<u>7.21.13 dbplus_errno() : Lit le code d'erreur généré par la dernière opération</u>	320
<u>7.21.14 dbplus_find() : Ajoute une contrainte sur une relation</u>	320
<u>7.21.15 dbplus_first() : Lit la première ligne d'une relation</u>	321
<u>7.21.16 dbplus_flush() : Ecris toutes les modifications apportées à une relation</u>	321
<u>7.21.17 dbplus_freealllocks() : Libère tous les verrous posés par le client</u>	321
<u>7.21.18 dbplus_freelock() : Lève un verrou en écriture sur une ligne</u>	322
<u>7.21.19 dbplus_freerlocks() : Lève tous les verrous sur les lignes d'une relation</u>	322
<u>7.21.20 dbplus_getlock() : Pose un verrou sur une ligne</u>	322
<u>7.21.21 dbplus_getunique() : Lit un identifiant unique pour une relation</u>	322
<u>7.21.22 dbplus_info() : Encore non implémentée</u>	323

Sommaire

7 Index des fonctions

<u>7.21.23 dbplus_last()</u> : Lit la dernière ligne d'une relation.....	323
<u>7.21.24 dbplus_lockrel()</u> : Pose un verrou en écriture sur une relation.....	323
<u>7.21.25 dbplus_next()</u> : Lit la ligne suivante dans la relation.....	324
<u>7.21.26 dbplus_open()</u> : Ouvre un fichier de relation.....	324
<u>7.21.27 dbplus_prev()</u> : Lit la ligne précédente dans la relation.....	324
<u>7.21.28 dbplus_rchperm()</u> : Modifie les droits sur la relation.....	325
<u>7.21.29 dbplus_rcreate()</u> : Crée une nouvelle relation DB++.....	325
<u>7.21.30 dbplus_rcreatex()</u> : Crée une copie exacte mais vide d'une relation.....	325
<u>7.21.31 dbplus_rcreatelike()</u> : Crée une copie exacte d'une relation, sans les indexes.....	326
<u>7.21.32 dbplus_resolve()</u> : Résoud les informations d'hôte pour une relation.....	326
<u>7.21.33 dbplus_restorepos()</u> : Pas encore implémenté.....	326
<u>7.21.34 dbplus_rkeys()</u> : Spécifie la nouvelle clé primaire d'une relation.....	326
<u>7.21.35 dbplus_ropen()</u> : Ouvre un fichier de relation local.....	327
<u>7.21.36 dbplus_rquery()</u> : Performe une requête AOL locale.....	327
<u>7.21.37 dbplus_rename()</u> : Renomme une relation.....	327
<u>7.21.38 dbplus_rsecindex()</u> : Crée un nouvel index secondaire pour une relation.....	327
<u>7.21.39 dbplus_runlink()</u> : Détruit une relation.....	328
<u>7.21.40 dbplus_rzap()</u> : Supprime toutes les lignes d'une relation.....	328
<u>7.21.41 dbplus_savepos()</u> : Pas encore implémentée.....	328
<u>7.21.42 dbplus_setindex()</u> : Pas encore implémentée.....	328
<u>7.21.43 dbplus_setindexbynumber()</u> : Pas encore implémentée.....	329
<u>7.21.44 dbplus_sql()</u> : Exécute une requête SQL.....	329
<u>7.21.45 dbplus_tcl()</u> : Exécute du code TCL coté serveur.....	329
<u>7.21.46 dbplus_remove()</u> : Supprime la ligne courante et retourne la nouvelle ligne courante.....	330
<u>7.21.47 dbplus_undo()</u> : Pas encore implémentée.....	330
<u>7.21.48 dbplus_undoprepere()</u> : Pas encore implémentée.....	330
<u>7.21.49 dbplus_unlockrel()</u> : Libère un verrou en écriture posé sur une relation.....	330
<u>7.21.50 dbplus_unselect()</u> : Retire une contrainte d'une relation.....	331
<u>7.21.51 dbplus_update()</u> : Modifie une ligne dans une relation.....	331
<u>7.21.52 dbplus_xlockrel()</u> : Pose un verrou exclusif sur une relation.....	331
<u>7.21.53 dbplus_xunlockrel()</u> : Lève un verrou exclusif posé sur une relation.....	331
<u>7.22 Direct IO.....</u>	332
<u>7.22.1 Introduction.....</u>	332
<u>7.22.2 Pré-requis.....</u>	332
<u>7.22.3 Installation.....</u>	332
<u>7.22.4 Configuration à l'exécution.....</u>	332
<u>7.22.5 Types de ressources.....</u>	332
<u>7.22.6 Constantes prédefinies.....</u>	332
<u>7.22.8 dio_fcntl()</u> : Exécute une fonction fichier de langage C.....	333
<u>7.22.9 dio_open()</u> : Ouvre un nouveau fichier.....	333
<u>7.22.10 dio_read()</u> : Lit des octets dans un fichier.....	333
<u>7.22.11 dio_seek()</u> : Déplace le pointeur interne de fichier.....	334
<u>7.22.12 dio_stat()</u> : Lit des informations sur le fichier.....	334
<u>7.22.13 dio_tcsetattr()</u> : Modifie les attributs du terminal et le taux de baud du port série.....	334
<u>7.22.14 dio_truncate()</u> : Tronque un fichier.....	335
<u>7.22.15 dio_write()</u> : Ecrit des données dans le fichier.....	335
<u>7.23 Accès aux dossiers.....</u>	336
<u>7.23.1 Introduction.....</u>	336
<u>7.23.2 Pré-requis.....</u>	336

Sommaire

7 Index des fonctions

<u>7.23.3 Installation</u>	336
<u>7.23.4 Configuration à l'exécution</u>	336
<u>7.23.5 Types de ressources</u>	336
<u>7.23.6 Constantes prédefinies</u>	336
<u>7.23.7 Voir aussi</u>	336
<u>7.23.9 chroot() : Change le dossier racine</u>	337
<u>7.23.10 closedir() : Ferme le pointeur sur le dossier</u>	337
<u>7.23.11 dir() : Classe dossier</u>	337
<u>7.23.12 getcwd() : Retourne le dossier de travail</u>	337
<u>7.23.13 opendir() : Ouvre un dossier, et récupère un pointeur dessus</u>	338
<u>7.23.14 readdir() : Lit une entrée du dossier</u>	338
<u>7.23.15 rewinddir() : Retourne à la première entrée du dossier</u>	339
<u>7.23.16 scandir() : Liste les fichiers et dossier dans un dossier</u>	339
<u>7.24 DOM XML</u>	341
<u>7.24.1 Introduction</u>	341
<u>7.24.2 Pré-requis</u>	341
<u>7.24.3 Installation</u>	341
<u>7.24.4 Deprecated functions</u>	341
<u>7.24.5 Constantes prédefinies</u>	342
<u>7.24.6 Classes</u>	343
<u>7.24.7 Exemples</u>	344
<u>7.24.9 domattribute->specified() : Vérifie si un attribut est spécifié</u>	346
<u>7.24.10 domattribute->value() : Retourne la valeur d'un attribut</u>	346
<u>7.24.11 domdocument->add_root [obsolète]() : Ajoute un noeud</u>	347
<u>7.24.12 domdocument->create_attribute() : Create new attribute</u>	347
<u>7.24.13 domdocument->create_cdata_section() : Crée un nouveau noeud CDATA</u>	347
<u>7.24.14 domdocument->create_comment() : Crée un nouveau noeud de commentaire</u>	348
<u>7.24.15 domdocument->create_element_ns() : Crée un nouvel élément avec un espace de noms associé</u>	348
<u>7.24.16 domdocument->create_element() : Crée un nouveau noeud élément</u>	348
<u>7.24.17 domdocument->create_entity_reference() : Crée un objet DomEntityReference</u>	348
<u>7.24.18 domdocument->create_processing_instruction() : Crée un nouveau noeud de traitement d'insctructions (PI)</u>	349
<u>7.24.19 domdocument->create_text_node() : Crée un nouveau noeud de texte</u>	349
<u>7.24.20 domdocument->doctype() : Retourne le type de document</u>	349
<u>7.24.21 domdocument->document_element() : Retourne le noeud racine d'un document DOM XML</u>	350
<u>7.24.22 domdocument->dump_file() : Convertit un document DOM XML en fichier</u>	350
<u>7.24.23 domdocument->dump_mem() : Convertit l'arbre XML interne en une chaîne de caractères</u>	350
<u>7.24.24 domdocument->get_element_by_id() : Recherche un élément avec son identifiant</u>	351
<u>7.24.25 domdocument->get_elements_by_tagname() : get_elements_by_tagname()</u>	351
<u>7.24.26 domdocument->html_dump_mem() : Convertit l'arbre XML interne en une chaîne de caractères au format HTML</u>	351
<u>7.24.27 domdocument->xinclude() : Substitute XIncludes dans un objet domdocument</u>	352

Sommaire

7 Index des fonctions

<u>7.24.28 domdocumenttype->entities()</u> : Retourne la liste des entités.....	352
<u>7.24.29 domdocumenttype->internal_subset()</u> : Retourne le sous ensemble interne (internal subset).....	352
<u>7.24.30 domdocumenttype->name()</u> : Retourne le nom du type de document.....	352
<u>7.24.31 domdocumenttype->notations()</u> : Retourne la liste des notations.....	353
<u>7.24.32 domdocumenttype->public_id()</u> : Retourne l'identifiant public du type de document.....	353
<u>7.24.33 domdocumenttype->system_id()</u> : Retourne l'identifiant système du type de document.....	353
<u>7.24.34 domelement->get_attribute_node()</u> : Retourne la valeur d'un attribut.....	354
<u>7.24.35 domelement->get_attribute()</u> : Retourne la valeur d'un attribut.....	354
<u>7.24.36 domelement->get_elements_by_tagname()</u> : Lit les éléments par nom de balise.....	354
<u>7.24.37 domelement->has_attribute()</u> : Vérifie si un attribut existe.....	354
<u>7.24.38 domelement->remove_attribute()</u> : Supprime un attribut.....	354
<u>7.24.39 domelement->set_attribute()</u> : Ajoute un nouvel attribut.....	354
<u>7.24.40 domelement->>tagname()</u> : Retourne le nom d'un élément.....	355
<u>7.24.41 domnode->add_namespace()</u> : Ajoute une déclaration d'espace de nom à un noeud.....	355
<u>7.24.42 domnode->append_child()</u> : Ajoute un nouveau fils new child at the end of the children.....	355
<u>7.24.43 domnode->append_sibling()</u> : Ajoute un frère à un noeud.....	356
<u>7.24.44 domnode->attributes()</u> : Retourne la liste des attributs.....	357
<u>7.24.45 domnode->child_nodes()</u> : Retourne les fils d'un noeud.....	357
<u>7.24.46 domnode->clone_node()</u> : Clone un noeud.....	357
<u>7.24.47 domnode->dump_node()</u> : Convertit un noeud en chaîne.....	357
<u>7.24.48 domnode->first_child()</u> : Retourne le premier fils du noeud.....	357
<u>7.24.49 domnode->get_content()</u> : Lit le contenu du noeud.....	357
<u>7.24.50 domnode->has_attributess()</u> : Vérifie si un noeud a des attributs.....	358
<u>7.24.51 domnode->has_child_nodes()</u> : Vérifie si un noeud a des fils.....	358
<u>7.24.52 domnode->insert_before()</u> : Insère un nouveau noeud fils.....	358
<u>7.24.53 domnode->is_blank_node()</u> : Vérifie si un noeud est blanc.....	359
<u>7.24.54 domnode->last_child()</u> : Retourne le dernier fils du noeud.....	359
<u>7.24.55 domnode->next_sibling()</u> : Retourne le noeud frère suivant.....	359
<u>7.24.56 domnode->node_name()</u> : Retourne le nom du noeud.....	359
<u>7.24.57 domnode->node_type()</u> : Retourne le type de noeud.....	360
<u>7.24.58 domnode->node_value()</u> : Retourne la valeur d'un noeud.....	360
<u>7.24.59 domnode->owner_document()</u> : Retourne le document auquel appartient ce noeud DOM XML.....	360
<u>7.24.60 domnode->parent_node()</u> : Retourne le père d'un noeud DOM XML.....	361
<u>7.24.61 domnode->prefix()</u> : Retourne le préfixe d'espace de nom d'un noeud.....	361
<u>7.24.62 domnode->previous_sibling()</u> : Retourne le frère précédent d'un noeud.....	361
<u>7.24.63 domnode->remove_child()</u> : Supprime un fils de la liste des noeuds fils.....	361
<u>7.24.64 domnode->replace_child()</u> : Remplace un noeud fils.....	362
<u>7.24.65 domnode->replace_node()</u> : Remplace un noeud DomXML.....	362
<u>7.24.66 domnode->set_content()</u> : Modifie le contenu d'un noeud.....	362
<u>7.24.67 domnode->set_name()</u> : Modifie le nom d'un noeud.....	363
<u>7.24.68 domnode->set_namespace()</u> : Modifie l'espace de noms d'un noeud.....	363
<u>7.24.69 domnode->unlink_node()</u> : Efface un noeud.....	363
<u>7.24.70 domprocessinginstruction->data()</u> : Retourne les données du script.....	363
<u>7.24.71 domprocessinginstruction->target()</u> : Retourne la cible du script.....	363

Sommaire

7 Index des fonctions

<u>7.24.72 domxsltstylesheet->process()</u> : Applique une transformation XSLT à un objet domdocument.....	364
<u>7.24.73 domxsltstylesheet->result_dump_file()</u> : Ecrit le résultat d'une transformation XSLT dans un fichier.....	364
<u>7.24.74 domxsltstylesheet->result_dump_mem()</u> : Ecrit le résultat d'une transformation XSLT dans une chaîne.....	364
<u>7.24.75 domxml_new_doc()</u> : Crée un document XML vide.....	365
<u>7.24.76 domxml_open_file()</u> : Crée un objet DOM à partir d'un fichier XML.....	365
<u>7.24.77 domxml_open_mem()</u> : Crée un objet DOM pour un document XML.....	366
<u>7.24.78 domxml_version()</u> : Lit le numéro de version de la librairie XML.....	366
<u>7.24.79 domxml_xmltree()</u> : Crée un arbre d'objets PHP à partir d'un document XML.....	366
<u>7.24.80 domxml_xslt_stylesheet_doc()</u> : Crée un objet domxsltstylesheet à partir d'un objet domdocument.....	366
<u>7.24.81 domxml_xslt_stylesheet_file()</u> : Crée un objet domxsltstylesheet à partir d'un document xsl dans un fichier.....	367
<u>7.24.82 domxml_xslt_stylesheet()</u> : Crée un objet domxsltstylesheet à partir d'un document xml.....	367
<u>7.24.83 xpath_eval_expression()</u> : Calcule un chemin XPath à partir d'une chaîne.....	367
<u>7.24.84 xpath_eval()</u> : Calcule un chemin XPath à partir d'une chaîne.....	368
<u>7.24.85 xpath_new_context()</u> : Crée un nouveau contexte xpath.....	368
<u>7.24.86 xpath_eval()</u> : Calcul un chemin XPath à partir d'une chaîne.....	368
<u>7.24.87 xpath_new_context()</u> : Crée un nouveau contexte XPath.....	368
<u>7.25 .NET</u>	369
<u>7.25.1 Introduction</u>	369
<u>7.26 Gestion des erreurs</u>	370
<u>7.26.1 Introduction</u>	370
<u>7.26.2 Pré-requis</u>	370
<u>7.26.3 Installation</u>	370
<u>7.26.4 Configuration à l'exécution</u>	370
<u>7.26.5 Constantes prédefinies</u>	372
<u>7.26.6 Exemples</u>	373
<u>7.26.7 Voir aussi</u>	374
<u>7.26.9 debug_print_backtrace()</u> : Affiche une backtrace.....	374
<u>7.26.10 error_log()</u> : Stocke un message d'erreur.....	375
<u>7.26.11 error_reporting()</u> : Fixe le niveau de rapport d'erreurs PHP.....	375
<u>7.26.12 restore_error_handler()</u> : Réactive l'ancienne fonction de gestion des erreurs.....	376
<u>7.26.13 set_error_handler()</u> : Choisi une fonction utilisateur comme gestionnaire d'erreurs.....	376
<u>7.26.14 trigger_error()</u> : Déclenche une erreur utilisateur.....	378
<u>7.26.15 user_error()</u> : Alias de trigger_error.....	379
<u>7.27 File alteration monitor functions</u>	380
<u>7.27.1 Introduction</u>	380
<u>7.27.2 Pré-requis</u>	380
<u>7.27.3 Installation</u>	380
<u>7.27.4 Configuration à l'exécution</u>	380
<u>7.27.5 Types de ressources</u>	380
<u>7.27.6 Constantes prédefinies</u>	380
<u>7.27.8 fam_close()</u> : Close FAM connection.....	381
<u>7.27.9 fam_monitor_collection()</u> : Monitor a collection of files in a directory for	

Sommaire

7 Index des fonctions

<u>changes</u>	381
<u>7.27.10 fam_monitor_directory()</u> : Monitor a directory for changes.....	381
<u>7.27.11 fam_monitor_file()</u> : Monitor a regular file for changes.....	381
<u>7.27.12 fam_next_event()</u> :	382
<u>7.27.13 fam_open()</u> : Open connection to FAM daemon.....	382
<u>7.27.14 fam_pending()</u> : Check for pending FAM events.....	382
<u>7.27.15 fam_resume_monitor()</u> : Resume suspended monitoring.....	382
<u>7.27.16 fam_suspend_monitor()</u> : Temporarily suspend monitoring.....	382
<u>7.28 FrontBase</u>	383
<u>7.28.1 Introduction</u>	383
<u>7.28.2 Pré-requis</u>	383
<u>7.28.3 Installation</u>	383
<u>7.28.4 Configuration à l'exécution</u>	383
<u>7.28.5 Types de ressources</u>	384
<u>7.28.6 Constantes prédefinies</u>	384
<u>7.28.8 fbsql_autocommit()</u> : Active ou désactive la validation automatique.....	385
<u>7.28.9 fbsql_change_user()</u> : Change le nom d'utilisateur de la session active.....	385
<u>7.28.10 fbsql_close()</u> : Ferme la connexion FrontBase.....	386
<u>7.28.11 fbsql_commit()</u> : Valide une transaction.....	386
<u>7.28.12 fbsql_connect()</u> : Ouvre une connexion à un serveur FrontBase.....	386
<u>7.28.13 fbsql_create_blob()</u> : Crée un BLOB.....	387
<u>7.28.14 fbsql_create_clob()</u> : Crée un CLOB.....	387
<u>7.28.15 fbsql_create_db()</u> : Crée une base de données.....	388
<u>7.28.16 fbsql_data_seek()</u> : Déplace le pointeur interne de résultat FBSQL.....	388
<u>7.28.17 fbsql_database_password()</u> : Modifie/lit le mot de passe dans une base FBSQL.....	389
<u>7.28.18 fbsql_database()</u> : Lit ou définit le nom de la base de données utilisée avec une connexion.....	389
<u>7.28.19 fbsql_db_query()</u> : Envoie une requête à la base FrontBase.....	389
<u>7.28.20 fbsql_db_status()</u> : Lit le statut courant d'une base de données.....	390
<u>7.28.21 fbsql_drop_db()</u> : Supprime une base de données FrontBase.....	390
<u>7.28.22 fbsql_errno()</u> : Retourne le code d'erreur FrontBase.....	390
<u>7.28.23 fbsql_error()</u> : Retourne le message d'erreur FrontBase.....	391
<u>7.28.24 fbsql_fetch_array()</u> : Lit toute une ligne de résultat dans un tableau.....	391
<u>7.28.25 fbsql_fetch_assoc()</u> : Lit toute une ligne de résultat dans un tableau associatif.....	392
<u>7.28.26 fbsql_fetch_field()</u> : Lit des informations sur une colonne dans un résultat, et retourne un objet.....	392
<u>7.28.27 fbsql_fetch_lengths()</u> : Lit la taille de chaque colonne d'un résultat.....	393
<u>7.28.28 fbsql_fetch_object()</u> : Lit une ligne de résultat sous forme d'objet.....	393
<u>7.28.29 fbsql_fetch_row()</u> : Lit une ligne de résultat sous forme de tableau numérique.....	394
<u>7.28.30 fbsql_field_flags()</u> : Lit les options associé à une colonne de résultat.....	394
<u>7.28.31 fbsql_field_len()</u> : Retourne la taille d'un champ.....	394
<u>7.28.32 fbsql_field_name()</u> : Lit le nom d'un champ.....	394
<u>7.28.33 fbsql_field_seek()</u> : Déplace le pointeur de résultat.....	395
<u>7.28.34 fbsql_field_table()</u> : Lit le nom de la table d'origine d'un champ.....	395
<u>7.28.35 fbsql_field_type()</u> : Lit le type d'une colonne.....	395
<u>7.28.36 fbsql_free_result()</u> : Libère le résultat de la mémoire.....	396
<u>7.28.37 fbsql_get_autostart_info()</u> : Pas de documentation.....	396
<u>7.28.38 fbsql_hostname()</u> : Lit ou définit le nom de l'hôte à utiliser avec une	

Sommaire

7 Index des fonctions

<u>connexion</u>	396
<u>7.28.39 fbsql_insert_id()</u> : Lit le dernier identifiant généré par une requête INSERT.....	396
<u>7.28.40 fbsql_list_dbs()</u> : Liste les bases de données FBSQL.....	397
<u>7.28.41 fbsql_list_fields()</u> : Liste les champs d'un résultat FrontBase.....	397
<u>7.28.42 fbsql_list_tables()</u> : Liste les tables dans une base de données FrontBase.....	398
<u>7.28.43 fbsql_next_result()</u> : Déplace le pointeur interne vers le résultat suivant.....	398
<u>7.28.44 fbsql_num_fields()</u> : Lit le nombre de champs dans un résultat.....	398
<u>7.28.45 fbsql_num_rows()</u> : Lit le nombre de lignes dans un résultat.....	398
<u>7.28.46 fbsql_password()</u> : Lit ou définit le mot de passe à utiliser avec une connexion.....	399
<u>7.28.47 fbsql_pconnect()</u> : Ouvre une connexion persistante à un serveur FrontBase.....	399
<u>7.28.48 fbsql_query()</u> : Exécute une requête sur un serveur FrontBase.....	399
<u>7.28.49 fbsql_read_blob()</u> : Lit un BLOB dans une base de données.....	400
<u>7.28.50 fbsql_read_clob()</u> : Lit un CLOB dans une base de données.....	401
<u>7.28.51 fbsql_result()</u> : Lit des données dans un résultat.....	401
<u>7.28.52 fbsql_rollback()</u> : Annule une transaction FBSQL.....	402
<u>7.28.53 fbsql_select_db()</u> : Sélectionne une base de données FrontBase.....	402
<u>7.28.54 fbsql_set_lob_mode()</u> : Modifie le mode de lecture des LOB.....	402
<u>7.28.55 fbsql_set_transaction()</u> : Configure le verrouillage et l'isolation de transactions.....	403
<u>7.28.56 fbsql_start_db()</u> : Démarre une base de données FBSQL.....	403
<u>7.28.57 fbsql_stop_db()</u> : Stoppe une base de données FBSQL.....	403
<u>7.28.58 fbsql_tablename()</u> : Lit le nom de la table d'un champ.....	403
<u>7.28.59 fbsql_username()</u> : Lit ou définit le nom de login à utiliser avec une connexion.....	404
<u>7.28.60 fbsql_warnings()</u> : Active ou désactive les alertes FrontBase.....	404
<u>7.29 FilePro</u>	405
<u>7.29.3 filepro_fieldcount()</u> : Retourne le nombre de champs dans une base filePro.....	405
<u>7.29.5 filepro_fieldtype()</u> : Retourne le type d'un champ Filepro.....	405
<u>7.29.6 filepro_fieldwidth()</u> : Retourne la taille d'un champ Filepro.....	405
<u>7.29.7 filepro_retrieve()</u> : Retourne la valeur d'un champ Filepro.....	405
<u>7.29.8 filepro_rowcount()</u> : Retourne le nombre de champs dans une base filePro.....	405
<u>7.29.9 filepro()</u> : Lit et vérifie un fichier.....	406
<u>7.30 Système de fichiers</u>	407
<u>7.30.1 Introduction</u>	407
<u>7.30.2 Pré-requis</u>	407
<u>7.30.3 Installation</u>	407
<u>7.30.4 Configuration à l'exécution</u>	407
<u>7.30.5 Types de ressources</u>	408
<u>7.30.6 Constantes prédefinies</u>	408
<u>7.30.7 Voir aussi</u>	408
<u>7.30.9 chgrp()</u> : Change le groupe propriétaire d'un fichier.....	410
<u>7.30.10 chmod()</u> : Change le mode du fichier.....	410
<u>7.30.11 chown()</u> : Change le groupe propriétaire du fichier.....	411
<u>7.30.12 clearstatcache()</u> : Efface le cache de stat.....	411
<u>7.30.13 copy()</u> : Copie un fichier.....	411
<u>7.30.14 delete()</u> : Effacer.....	412
<u>7.30.15 dirname()</u> : Renvoie le nom du dossier.....	412
<u>7.30.16 disk_free_space()</u> : Renvoie l'espace disque disponible dans le répertoire.....	413

Sommaire

7 Index des fonctions

<u>7.30.17 disk_total_space()</u> : Retourne la taille d'un dossier.....	413
<u>7.30.18 diskfreespace()</u> : Alias de disk_free_space.....	413
<u>7.30.19 fclose()</u> : Ferme un fichier.....	413
<u>7.30.20 feof()</u> : Teste la fin du fichier.....	414
<u>7.30.21 fflush()</u> : Envoi tout le contenu généré dans un fichier.....	414
<u>7.30.22 fgetc()</u> : Lit un caractère dans un fichier.....	414
<u>7.30.23 fgetcsv()</u> : Renvoie la ligne courante et cherche les champs CSV.....	415
<u>7.30.24 fgets()</u> : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier.....	415
<u>7.30.25 fgets()</u> : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier et élimine les balises HTML.....	416
<u>7.30.26 file_exists()</u> : Vérifie si un fichier existe.....	417
<u>7.30.27 file_get_contents()</u> : Lit tout un fichier dans une chaîne.....	417
<u>7.30.28 file_put_contents()</u> : Ecrit une chaîne dans un fichier.....	417
<u>7.30.29 file()</u> : Lit le fichier et renvoie le résultat dans un tableau.....	418
<u>7.30.30 fileatime()</u> : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois.....	419
<u>7.30.31 filectime()</u> : Renvoie l'heure à laquelle l'inode a été accédé pour la dernière fois.....	419
<u>7.30.32 filegroup()</u> : Lire le nom du groupe.....	419
<u>7.30.33 fileinode()</u> : Renvoie le numéro d'inode du fichier.....	420
<u>7.30.34 filemtime()</u> : Renvoie la date de dernière modification du fichier.....	420
<u>7.30.35 fileowner()</u> : Renvoie le nom du propriétaire du fichier.....	420
<u>7.30.36 fileperms()</u> : Renvoie les permissions affectées au fichier.....	421
<u>7.30.37 filesize()</u> : Renvoie la taille du fichier.....	421
<u>7.30.38 filetype()</u> : Retourne le type de fichier.....	421
<u>7.30.39 flock()</u> : Verrouille le fichier.....	421
<u>7.30.40 fnmatch()</u> : Repère un fichier à partir d'un masque.....	422
<u>7.30.41 fopen()</u> : Ouverture d'un fichier ou d'une URL.....	423
<u>7.30.42 fpassthru()</u> : Affiche le reste du fichier.....	424
<u>7.30.43 fputs()</u> : Ecrit dans un fichier.....	424
<u>7.30.44 fread()</u> : Lecture du fichier en mode binaire.....	425
<u>7.30.45 fscanf()</u> : Analyse un fichier en fonction d'un format.....	425
<u>7.30.46 fseek()</u> : Modifie le pointeur de fichier.....	426
<u>7.30.47 fstat()</u> : Lit les informations sur un fichier à partir d'un pointeur de fichier.....	426
<u>7.30.48 ftell()</u> : Renvoie la position du pointeur du fichier.....	427
<u>7.30.49 ftruncate()</u> : Tronque un fichier.....	427
<u>7.30.50 fwrite()</u> : Ecriture du fichier en mode binaire.....	427
<u>7.30.51 glob()</u> : Recherche des chemins qui vérifie un masque.....	428
<u>7.30.52 is_dir()</u> : Indique si le nom de fichier est un dossier.....	429
<u>7.30.53 is_executable()</u> : Indique si le fichier est exécutable.....	429
<u>7.30.54 is_file()</u> : Indique si le fichier est un véritable fichier.....	429
<u>7.30.55 is_link()</u> : Indique si le fichier est un lien symbolique.....	429
<u>7.30.56 is_readable()</u> : Indique un fichier est autorisé en lecture.....	430
<u>7.30.57 is_uploaded_file()</u> : Indique si le fichier a été téléchargé par HTTP POST.....	430
<u>7.30.58 is_writable()</u> : Indique si un fichier est autorisé en écriture.....	431
<u>7.30.59 is_writeable()</u> : Indique si un fichier est autorisé en écriture.....	431
<u>7.30.60 link()</u> : Crée un lien.....	431
<u>7.30.61 linkinfo()</u> : Renvoie les informations à propos d'un lien.....	432
<u>7.30.62 lstat()</u> : Retourne les informations à propos d'un fichier ou d'un lien symbolique.....	432

Sommaire

7 Index des fonctions

<u>7.30.63 mkdir()</u> : Crée un dossier.....	432
<u>7.30.64 move_uploaded_file()</u> : Déplace un fichier téléchargé.....	433
<u>7.30.65 parse_ini_file()</u> : Traite un fichier de configuration.....	433
<u>7.30.66 pathinfo()</u> : Retourne des informations sur un chemin système.....	434
<u>7.30.67 pclose()</u> : Ferme un processus de pointeur de fichier.....	435
<u>7.30.68 popen()</u> : Crée un processus de pointeur de fichier.....	435
<u>7.30.69 readfile()</u> : Affiche un fichier.....	436
<u>7.30.70 readlink()</u> : Renvoie le nom du fichier vers lequel pointe un lien symbolique.....	436
<u>7.30.71 realpath()</u> : Retourne le chemin canonique absolu.....	436
<u>7.30.72 rename()</u> : Renomme un fichier.....	437
<u>7.30.73 rewind()</u> : Remplace le pointeur de fichier au début.....	437
<u>7.30.74 rmdir()</u> : Efface un dossier.....	437
<u>7.30.75 set_file_buffer()</u> : Alias de stream_set_write_buffer.....	438
<u>7.30.76 stat()</u> : Renvoie les informations à propos d'un fichier.....	438
<u>7.30.77 symlink()</u> : Crée un lien symbolique.....	438
<u>7.30.78 tempnam()</u> : Crée un fichier avec un nom unique.....	439
<u>7.30.79 tmpfile()</u> : Crée un fichier temporaire.....	439
<u>7.30.80 touch()</u> : Modifie la date de modification et de dernier accès d'un fichier.....	440
<u>7.30.81 umask()</u> : Change le "umask" courant.....	440
<u>7.30.82 unlink()</u> : Efface un fichier.....	440
<u>7.31 Forms Data Format</u>	441
<u>7.31.1 Introduction</u>	441
<u>7.31.2 Pré-requis</u>	441
<u>7.31.3 Installation</u>	441
<u>7.31.4 Configuration à l'exécution</u>	442
<u>7.31.5 Types de ressources</u>	442
<u>7.31.6 Constantes prédefinies</u>	442
<u>7.31.7 Exemples</u>	442
<u>7.31.9 fdf_add_template()</u> : Ajoute un template dans le document FDF.....	444
<u>7.31.10 fdf_close()</u> : Ferme un document FDF.....	444
<u>7.31.11 fdf_create()</u> : Crée un nouveau document FDF.....	444
<u>7.31.12 fdf_enum_values()</u> : Appelle une fonction utilisateur à chaque valeur FDF.....	444
<u>7.31.13 fdf_erro()</u> : Retourne le code d'erreur de la dernière opération FDF.....	445
<u>7.31.14 fdf_error()</u> : Retourne le message d'erreur FDF.....	445
<u>7.31.15 fdf_get_ap()</u> : Lit l'apparence d'un champ.....	445
<u>7.31.16 fdf_get_attachment()</u> : Extrait un fichier intégré dans un document FDF.....	445
<u>7.31.17 fdf_get_encoding()</u> : Lit la valeur de la clé /Encoding.....	446
<u>7.31.18 fdf_get_file()</u> : Lit la valeur de la clé /F.....	446
<u>7.31.19 fdf_get_flags()</u> : Lit les attributs d'un champ FDF.....	446
<u>7.31.20 fdf_get_opt()</u> : Lit une valeur dans un tableau de valeur d'un champ FDF.....	446
<u>7.31.21 fdf_get_status()</u> : Lit la valeur de la clé /STATUS.....	446
<u>7.31.22 fdf_get_value()</u> : Retourne la valeur d'un champ FDF.....	447
<u>7.31.23 fdf_get_version()</u> : Lit le numéro de version de l'API FDF.....	447
<u>7.31.24 fdf_header()</u> : Emet les en-têtes HTTP spécifiques à FDF.....	447
<u>7.31.25 fdf_next_field_name()</u> : Lit le nom du champ FDF suivant.....	447
<u>7.31.26 fdf_open_string()</u> : Lit un document FDF à partir d'une chaîne de caractères.....	448
<u>7.31.27 fdf_open()</u> : Ouvre un document FDF.....	448
<u>7.31.28 fdf_remove_item()</u> : Configure le cadre FDF de destination pour le formulaire.....	448

Sommaire

7 Index des fonctions

<u>7.31.29 fdf_save_string()</u> : Retourne un document FDF sous forme de chaîne.....	449
<u>7.31.30 fdf_save()</u> : Sauver un document FDF.....	449
<u>7.31.31 fdf_set_ap()</u> : Fixe l'apparence d'un champ FDF.....	449
<u>7.31.32 fdf_set_encoding()</u> : Modifie l'encodage des caractères.....	450
<u>7.31.33 fdf_set_file()</u> : Assigne une valeur à un champ FDF.....	450
<u>7.31.34 fdf_set_flags()</u> : Modifie une option d'un champ.....	450
<u>7.31.35 fdf_set_javascript_action()</u> : Modifie l'action javascript d'un champ.....	450
<u>7.31.36 fdf_set_opt()</u> : Modifie une option d'un champ.....	451
<u>7.31.37 fdf_set_status()</u> : Fixe la valeur de la clé /STATUS.....	451
<u>7.31.38 fdf_set_submit_form_action()</u> : Modifie l'action d'un formulaire.....	451
<u>7.31.39 fdf_set_target_frame()</u> : Configure le cadre de destination pour l'affichage du formulaire.....	451
<u>7.31.40 fdf_set_value()</u> : Modifie la valeur d'un champ FDF.....	451
<u>7.31.41 fdf_set_version()</u> : Modifie le numéro de version du fichier FDF.....	452
<u>7.32 FriBiDi</u>	453
<u>7.32.1 Introduction</u>	453
<u>7.32.2 Pré-requis</u>	453
<u>7.32.3 Installation</u>	453
<u>7.32.4 Configuration à l'exécution</u>	453
<u>7.32.5 Types de ressources</u>	453
<u>7.32.6 Constantes prédefinies</u>	453
<u>7.33 FTP</u>	454
<u>7.33.1 Introduction</u>	454
<u>7.33.2 Pré-requis</u>	454
<u>7.33.3 Installation</u>	454
<u>7.33.4 Configuration à l'exécution</u>	454
<u>7.33.5 Types de ressources</u>	454
<u>7.33.6 Constantes prédefinies</u>	454
<u>7.33.7 Exemples</u>	455
<u>7.33.9 ftp_chdir()</u> : Change le dossier FTP courant.....	456
<u>7.33.10 ftp_chmod()</u> : Modifie les droits d'un fichier via FTP.....	456
<u>7.33.11 ftp_close()</u> : Ferme une connexion FTP.....	457
<u>7.33.12 ftp_connect()</u> : Ouvre une connexion FTP.....	457
<u>7.33.13 ftp_delete()</u> : Efface un fichier sur un serveur FTP.....	457
<u>7.33.14 ftp_exec()</u> : Exécute une commande sur le serveur FTP.....	457
<u>7.33.15 ftp_fget()</u> : Télécharge un fichier via FTP dans un flôt local.....	458
<u>7.33.16 ftp_fput()</u> : Charge un fichier sur un serveur FTP.....	458
<u>7.33.17 ftp_get_option()</u> : Lit différentes options pour la connexion FTP courante....	458
<u>7.33.18 ftp_get()</u> : Télécharge un fichier depuis un serveur FTP.....	459
<u>7.33.19 ftp_login()</u> : Identification avec une connexion FTP.....	459
<u>7.33.20 ftp_mdtm()</u> : Retourne la date de dernière modification d'un fichier sur un serveur FTP.....	459
<u>7.33.21 ftp_mkdir()</u> : Crée un dossier sur un serveur FTP.....	460
<u>7.33.22 ftp_nb_continue()</u> : Reprend le téléchargement d'un fichier (non bloquant)...	460
<u>7.33.23 ftp_nb_fget()</u> : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant).....	460
<u>7.33.24 ftp_nb_fput()</u> : Écrit un fichier sur un serveur FTP, et le lit depuis un fichier (non bloquant).....	460
<u>7.33.25 ftp_nb_get()</u> : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant).....	461
<u>7.33.26 ftp_nb_put()</u> : Envoie un fichier sur le serveur FTP (non-bloquant).....	462

Sommaire

7 Index des fonctions

<u>7.33.27 ftp_nlist()</u> : Retourne la liste des fichiers dans un dossier.....	462
<u>7.33.28 ftp_pasv()</u> : Active ou désactive le mode passif.....	463
<u>7.33.29 ftp_put()</u> : Charge un fichier sur un serveur FTP.....	463
<u>7.33.30 ftp_pwd()</u> : Retourne le nom du dossier courant.....	463
<u>7.33.31 ftp_quit()</u> : Ferme une connexion FTP.....	463
<u>7.33.32 ftp_raw()</u> : Envoie une commande FTP brute.....	464
<u>7.33.33 ftp_rawlist()</u> : Fait une liste détaillée de fichiers dans un dossier.....	464
<u>7.33.34 ftp_rename()</u> : Renomme un fichier sur un serveur FTP.....	464
<u>7.33.35 ftp_rmdir()</u> : Efface un dossier FTP.....	464
<u>7.33.36 ftp_set_option()</u> : Modifie les options de la connexion FTP.....	465
<u>7.33.37 ftp_site()</u> : Envoie la commande SITE au serveur.....	465
<u>7.33.38 ftp_size()</u> : Retourne la taille d'un fichier.....	465
<u>7.33.39 ftp_ssl_connect()</u> : Ouvre une connexion FTP sécurisée avec SSL.....	465
<u>7.33.40 ftp_systype()</u> : Retourne un identifiant de type de serveur FTP.....	466
<u>7.34 Fonctions</u>	467
<u>7.34.1 Introduction</u>	467
<u>7.34.2 Pré-requis</u>	467
<u>7.34.3 Installation</u>	467
<u>7.34.4 Configuration à l'exécution</u>	467
<u>7.34.5 Constantes prédefinies</u>	467
<u>7.34.7 call_user_func()</u> : Appelle une fonction utilisateur.....	467
<u>7.34.8 create_function()</u> : Crée une fonction anonyme (style lambda).....	468
<u>7.34.9 func_get_arg()</u> : Retourne un élément de la liste des arguments.....	469
<u>7.34.10 func_get_args()</u> : Retourne les arguments d'une fonction sous forme de tableau.....	470
<u>7.34.11 func_num_args()</u> : Retourne le nombre d'arguments passé à la fonction.....	470
<u>7.34.12 function_exists()</u> : Indique si une fonction est définie.....	471
<u>7.34.13 get_defined_functions()</u> : Liste toutes les fonctions définies.....	471
<u>7.34.14 register_shutdown_function()</u> : Enregistre une fonction pour exécution à l'extinction.....	472
<u>7.34.15 register_tick_function()</u> : Enregistre une fonction exécutée à chaque tick.....	472
<u>7.34.16 unregister_tick_function()</u> : Annule la fonction exécutée à chaque tick.....	473
<u>7.35 Gettext (GNU)</u>	474
<u>7.35.1 Introduction</u>	474
<u>7.35.2 Pré-requis</u>	474
<u>7.35.3 Installation</u>	474
<u>7.35.4 Configuration à l'exécution</u>	474
<u>7.35.5 Types de ressources</u>	474
<u>7.35.6 Constantes prédefinies</u>	474
<u>7.35.8 bindtextdomain()</u> : Fixe le chemin d'un domaine.....	475
<u>7.35.9 dcgettext()</u> : Remplace le domaine lors d'une recherche.....	475
<u>7.35.10 dcngettext()</u> : Version plurielle de dcgettext.....	475
<u>7.35.11 dgettext()</u> : Remplace le domaine courant.....	475
<u>7.35.12 dngettext()</u> : Version plurielle de dgettext.....	475
<u>7.35.13 gettext()</u> : Recherche un message dans le domaine courant.....	475
<u>7.35.14 ngettext()</u> : Version plurielle de gettext.....	476
<u>7.35.15 textdomain()</u> : Fixe le domaine par défaut.....	476
<u>7.36 GMP</u>	477
<u>7.36.1 Introduction</u>	477
<u>7.36.2 Pré-requis</u>	477
<u>7.36.3 Installation</u>	477

Sommaire

7 Index des fonctions

<u>7.36.4 Configuration à l'exécution</u>	477
<u>7.36.5 Types de ressources</u>	477
<u>7.36.6 Constantes prédefinies</u>	478
<u>7.36.7 Exemples</u>	478
<u>7.36.8 Voir aussi</u>	478
<u>7.36.10 gmp_add() : Addition de 2 nombres GMP</u>	479
<u>7.36.11 gmp_and() : ET logique</u>	479
<u>7.36.12 gmp_clrbit() : Annule un bit</u>	479
<u>7.36.13 gmp_cmp() : Compare des nombres GMP</u>	479
<u>7.36.14 gmp_com() : Calcule le complémentaire d'un nombre</u>	480
<u>7.36.15 gmp_div_q() : Divisions de 2 nombres GMP</u>	480
<u>7.36.16 gmp_div_qr() : Divise deux nombres GMP</u>	480
<u>7.36.17 gmp_div_r() : Reste de la division de deux nombres GMP</u>	480
<u>7.36.18 gmp_div() : Divise deux nombres GMP</u>	481
<u>7.36.19 gmp_divexact() : Division exacte de nombres GMP</u>	481
<u>7.36.20 gmp_fact() : Factorielle GMP</u>	481
<u>7.36.21 gmp_gcd() : PGCD</u>	481
<u>7.36.22 gmp_gcdext() : PGCD étendu</u>	481
<u>7.36.23 gmp_hamdist() : Distance de Hamming</u>	482
<u>7.36.24 gmp_init() : Crée un nombre GMP</u>	482
<u>7.36.25 gmp_intval() : Convertit un nombre GMP en entier</u>	482
<u>7.36.26 gmp_invert() : Inverse modulo</u>	482
<u>7.36.27 gmp_jacobi() : Symbole de Jacobi</u>	483
<u>7.36.28 gmp_legendre() : Symbole de Legendre</u>	483
<u>7.36.29 gmp_mod() : Modulo GMP</u>	483
<u>7.36.30 gmp_mul() : Multiplication de 2 nombres GMP</u>	483
<u>7.36.31 gmp_neg() : Opposé de nombre GMP</u>	483
<u>7.36.32 gmp_or() : OU logique</u>	483
<u>7.36.33 gmp_perfect_square() : Carré parfait GMP</u>	483
<u>7.36.34 gmp_popcount() : Compte de population</u>	484
<u>7.36.35 gmp_pow() : Puissance</u>	484
<u>7.36.36 gmp_powm() : Puissance et modulo</u>	484
<u>7.36.37 gmp_prob_prime() : Nombre GMP probablement premier</u>	484
<u>7.36.38 gmp_random() : Nombre GMP aléatoire</u>	484
<u>7.36.39 gmp_scan0() : Recherche 0</u>	484
<u>7.36.40 gmp_scan1() : Recherche 1</u>	484
<u>7.36.41 gmp_setbit() : Modifie un bit</u>	485
<u>7.36.42 gmp_sign() : Signe du nombre GMP</u>	485
<u>7.36.43 gmp_sqrt() : Racine carrée GMP</u>	485
<u>7.36.44 gmp_sqrtrem() : Racine carrée avec reste GMP</u>	485
<u>7.36.45 gmp_strval() : Convertit un nombre GMP en chaîne</u>	485
<u>7.36.46 gmp_sub() : Soustraction de 2 nombres GMP</u>	485
<u>7.36.47 gmp_xor() : OU exclusif logique</u>	486
<u>7.37 HTTP</u>	487
<u>7.37.1 Introduction</u>	487
<u>7.37.2 Pré-requis</u>	487
<u>7.37.3 Installation</u>	487
<u>7.37.4 Configuration à l'exécution</u>	487
<u>7.37.5 Types de ressources</u>	487
<u>7.37.6 Constantes prédefinies</u>	487
<u>7.37.8 headers_sent() : Indique si les en-têtes HTTP ont déjà été envoyés</u>	487

Sommaire

7 Index des fonctions

7.37.9 <u>setcookie()</u> : Envoie un cookie.....	488
7.38 <u>Hyperwave</u>	491
7.38.1 <u>Introduction</u>	491
7.38.2 <u>Pré-requis</u>	493
7.38.3 <u>Installation</u>	493
7.38.4 <u>Intégration avec Apache</u>	493
7.38.5 <u>Configuration à l'exécution</u>	494
7.38.6 <u>Types de ressources</u>	494
7.38.7 <u>Constantes prédefinies</u>	494
7.38.8 <u>A faire</u>	495
7.38.10 <u>hw_changeobject()</u> : Change les attributs d'un objet (obsolète).....	496
7.38.11 <u>hw_children()</u> : Liste des object ids des objets fils.....	496
7.38.12 <u>hw_childrenobj()</u> : Liste des object records des objets fils.....	497
7.38.13 <u>hw_close()</u> : Ferme la connexion Hyperwave.....	497
7.38.14 <u>hw_connect()</u> : Ouvre une connexion Hyperwave.....	497
7.38.15 <u>hw_connection_info()</u> : Affiche des informations sur la connexion au serveur Hyperwave.....	497
7.38.16 <u>hw_cp()</u> : Copie des objets HyperWave.....	497
7.38.17 <u>hw_deleteobject()</u> : Efface des objets.....	498
7.38.18 <u>hw_docbyanchor()</u> : Identifiant d'objet de l'objet dans l'ancrage.....	498
7.38.19 <u>hw_docbyanchorobj()</u> : Attributs de l'objet dans l'ancrage.....	498
7.38.20 <u>hw_document_attributes()</u> : Object record de hw_document.....	498
7.38.21 <u>hw_document_bodytag()</u> : Balise de corps d'un document.....	498
7.38.22 <u>hw_document_content()</u> : Contenu d'un document.....	498
7.38.23 <u>hw_document_setcontent()</u> : Modifie/remplace le contenu d'un document.....	499
7.38.24 <u>hw_document_size()</u> : Taille d'un document.....	499
7.38.25 <u>hw_dummy()</u> : Fonction sans objet pour Hyperwave.....	499
7.38.26 <u>hw_edittest()</u> : Retourne un document texte.....	499
7.38.27 <u>hw_error()</u> : Retourne le code d'erreur.....	500
7.38.28 <u>hw_errormsg()</u> : Retourne un message d'erreur.....	500
7.38.29 <u>hw_free_document()</u> : Détruit un document.....	500
7.38.30 <u>hw_getanchors()</u> : Identifiants des ancrages d'un document.....	500
7.38.31 <u>hw_getanchorsobj()</u> : Attributs des ancrages d'un document.....	500
7.38.32 <u>hw_getandlock()</u> : Retourne les attributs, et verrouille l'objet.....	500
7.38.33 <u>hw_getchildcoll()</u> : Identifiant d'objet des groupes fils.....	501
7.38.34 <u>hw_getchildcollobj()</u> : object records d'un groupe d'enfants.....	501
7.38.35 <u>hw_getchilddoccoll()</u> : ids des documents fils d'un groupe.....	501
7.38.36 <u>hw_getchilddoccollobj()</u> : Attributs des documents fils d'un groupe.....	501
7.38.37 <u>hw_getobject()</u> : Lit les attributs d'un objet Hyperwave.....	501
7.38.38 <u>hw_getobjectbyquery()</u> : Recherche un objet.....	502
7.38.39 <u>hw_getobjectbyquerycoll()</u> : Recherche un objet dans un groupe.....	502
7.38.40 <u>hw_getobjectbyquerycollobj()</u> : Recherche un objet dans un groupe.....	502
7.38.41 <u>hw_getobjectbyqueryobj()</u> : Recherche un objet.....	503
7.38.42 <u>hw_getparents()</u> : Identifiant d'objet des parents.....	503
7.38.43 <u>hw_getparentsobj()</u> : Attributs des parents.....	503
7.38.44 <u>hw_getrellink()</u> : Crée le lien depuis la source vers le destinataire, relativement à rootid.....	503
7.38.45 <u>hw_getremote()</u> : Retourne un document distant.....	503
7.38.46 <u>hw_getremotechildren()</u> : Retourne les fils d'un document distant.....	504
7.38.47 <u>hw_getsrcbydestobj()</u> : Retourne les ancrages qui pointent sur un objet.....	504
7.38.48 <u>hw_gettext()</u> : Retourne un document texte.....	504

Sommaire

7 Index des fonctions

<u>7.38.49 hw_username()</u> : Nom de l'utilisateur actuellement identifié.....	505
<u>7.38.50 hw_identify()</u> : Identifie un utilisateur.....	505
<u>7.38.51 hw_incollections()</u> : Vérifie qu'un identifiant d'objet est dans un groupe.....	505
<u>7.38.52 hw_info()</u> : Informations à propos d'une connexion.....	505
<u>7.38.53 hw_inscoll()</u> : Insère un groupe.....	506
<u>7.38.54 hw_insdock()</u> : Insère un document.....	506
<u>7.38.55 hw_insertanchors()</u> : Insère uniquement des ancrs dans du texte.....	506
<u>7.38.56 hw_insertdocument()</u> : Insère un document dans un groupe.....	506
<u>7.38.57 hw_insertobject()</u> : Insère un objet record.....	506
<u>7.38.58 hw_mapid()</u> : Représente un id global en un id virtuel local.....	507
<u>7.38.59 hw_modifyobject()</u> : Modifie les attributs d'objet record.....	507
<u>7.38.60 hw_mv()</u> : Déplace un objet.....	508
<u>7.38.61 hw_new_document()</u> : Crée un nouveau document.....	509
<u>7.38.62 hw_objrec2array()</u> : Convertit les attributs d'un objet en tableau.....	509
<u>7.38.63 hw_output_document()</u> : Affiche hw_document.....	509
<u>7.38.64 hw_pconnect()</u> : Crée une connexion persistante.....	509
<u>7.38.65 hw_pipedocument()</u> : Retourne un document.....	510
<u>7.38.66 hw_root()</u> : Object id de la racine.....	510
<u>7.38.67 hw_setlinkroot()</u> : Configure l'identifiant vers lequel les liens sont calculés.....	510
<u>7.38.68 hw_stat()</u> : Retourne le statut de la chaîne.....	510
<u>7.38.69 hw_unlock()</u> : Déverrouille un objet.....	510
<u>7.38.70 hw_who()</u> : Liste des utilisateurs actuellement identifiés.....	511
<u>7.39 Hyperwave API.....</u>	512
<u>7.39.1 Introduction.....</u>	512
<u>7.39.2 Pré-requis.....</u>	512
<u>7.39.3 Installation.....</u>	512
<u>7.39.4 intégration dans Apache.....</u>	512
<u>7.39.5 Configuration à l'exécution.....</u>	512
<u>7.39.6 Types de ressources.....</u>	513
<u>7.39.7 Constantes prédefinies.....</u>	513
<u>7.39.8 Classes.....</u>	513
<u>7.39.10 hw_api_attribute->langdepvalue()</u> : Retourne la valeur d'un attribut dans une langue.....	515
<u>7.39.11 hw_api_attribute->value()</u> : Retourne la valeur de l'attribut HW.....	515
<u>7.39.12 hw_api_attribute->values()</u> : Retourne toutes les valeurs de l'attribut sous la forme d'un tableau de chaînes.....	515
<u>7.39.13 hw_api_attribute()</u> : Crée une nouvelle instance de la classe hw_api_attribute.....	515
<u>7.39.14 hw_api->checkin()</u> : Archive un objet.....	515
<u>7.39.15 hw_api->checkout()</u> : Extrait un objet.....	516
<u>7.39.16 hw_api->children()</u> : Retourne le fils d'un objet.....	516
<u>7.39.17 hw_api_content->mimetype()</u> : Retourne le type MIME.....	516
<u>7.39.18 hw_api_content->read()</u> : Lit le contenu.....	516
<u>7.39.19 hw_api->content()</u> : Retourne le contenu d'un objet.....	517
<u>7.39.20 hw_api->copy()</u> : Copies physically.....	517
<u>7.39.21 hw_api->dbstat()</u> : Retourne des statistiques sur le serveur de bases.....	517
<u>7.39.22 hw_api->dcstat()</u> : Retourne des statistiques sur le serveur de cache de document.....	517
<u>7.39.23 hw_api->dstanchors()</u> : Liste toutes les ancrs de destination.....	517
<u>7.39.24 hw_api->dstofsrcanchors()</u> : Retourne la destination d'une ancre source.....	518
<u>7.39.25 hw_api_error->count()</u> : Retourne le nombre de raisons d'erreurs.....	518

Sommaire

7 Index des fonctions

7.39.26 hw_api_error->reason()	: Retourne la première raison d'erreur.....	518
7.39.27 hw_api->find()	: Recherche des objets.....	518
7.39.28 hw_api->ftstat()	: Retourne des statistiques sur le serveur de textes.....	518
7.39.29 hwapi_hgcsp()	: Retourne un objet de classe hw_api.....	519
7.39.30 hw_api->hwstat()	: Retourne les statistiques sur le serveur Hyperwave.....	519
7.39.31 hw_api->identify()	: S'identifie auprès du serveur Hyperwave.....	519
7.39.32 hw_api->info()	: Retourne des information sur la configuration du serveur.....	519
7.39.33 hw_api->insert()	: Insère un nouvel objet.....	519
7.39.34 hw_api->insertanchor()	: Insère un nouvel objet de type ancre.....	520
7.39.35 hw_api->insertcollection()	: Insère un nouvel objet de type collection.....	520
7.39.36 hw_api->insertdocument()	: Insère un nouvel objet de type document.....	520
7.39.37 hw_api->link()	: Crée un lien avec un objet.....	521
7.39.38 hw_api->lock()	: Verrouille un objet.....	521
7.39.39 hw_api->move()	: Déplace un objet d'une collection à l'autre.....	521
7.39.40 hw_api->content()	: Crée une nouvelle instance de la classe hw_api_content.....	521
7.39.41 hw_api_object->assign()	: Clone un objet.....	521
7.39.42 hw_api_object->attreditable()	: Vérifie si un attribut est éditable.....	522
7.39.43 hw_api_object->count()	: Retourne le nombre d'attributs.....	522
7.39.44 hw_api_object->insert()	: Insère un nouvel attribut HyperWave.....	522
7.39.45 hw_api_object()	: Crée une nouvelle instance de la classe hw_api_object.....	522
7.39.46 hw_api_object->remove()	: Supprime un attribut HyperWave.....	522
7.39.47 hw_api_object->title()	: Retourne le titre de l'attribut.....	522
7.39.48 hw_api_object->value()	: Retourne la valeur de l'attribut.....	522
7.39.49 hw_api->object()	: Lit les informations d'attribut.....	523
7.39.50 hw_api->objectbyanchor()	: Retourne l'objet propriétaire d'une ancre.....	523
7.39.51 hw_api->parents()	: Retourne le parent d'un objet.....	524
7.39.52 hw_api_reason->description()	: Retourne la description d'une raison.....	524
7.39.53 hw_api_reason->type()	: Retourne le type de raison.....	524
7.39.54 hw_api->remove()	: Efface un objet.....	524
7.39.55 hw_api->replace()	: Remplace un objet.....	524
7.39.56 hw_api->setcommittedversion()	: Valide une version autre que la dernière.....	525
7.39.57 hw_api->srcanchors()	: Retourne la liste de toutes les ancrs.....	525
7.39.58 hw_api->srcsofdst()	: Retourne les sources d'un objet de destination.....	525
7.39.59 hw_api->unlock()	: Déverrouille un objet HyperWave verrouillé.....	526
7.39.60 hw_api->user()	: Retourne le propriétaire d'un objet.....	526
7.39.61 hw_api->userlist()	: Retourne la liste des utilisateurs connectés.....	526
7.40 Iconv.....		527
7.40.1 Introduction.....		527
7.40.2 Pré-requis.....		527
7.40.3 Installation.....		527
7.40.4 Configuration à l'exécution.....		527
7.40.5 Types de ressources.....		527
7.40.6 Constantes prédefinies.....		527
7.40.7 Voir aussi.....		528
7.40.9 iconv_set_encoding()	: Modifie le jeu courant de caractères courant.....	528
7.40.10 iconv()	: Convertit une chaîne dans un jeu de caractères.....	528
7.40.11 ob_iconv_handler()	: Gestionnaire de sortie pour maîtriser le jeu de caractères de sortie.....	529
7.41 Images.....		530
7.41.1 Introduction.....		530

Sommaire

7 Index des fonctions

<u>7.41.2 Pré-requis.....</u>	530
<u>7.41.3 Installation.....</u>	531
<u>7.41.4 Configuration à l'exécution.....</u>	531
<u>7.41.5 Types de ressources.....</u>	532
<u>7.41.6 Constantes prédefinies.....</u>	532
<u>7.41.7 Exemples.....</u>	533
<u>7.41.9 exif_read_data() : Lit les en-têtes EXIF des images JPEG et TIFF.....</u>	535
<u>7.41.10 exif_thumbnail() : Récupère la miniature d'une image TIFF ou JPEG.....</u>	537
<u>7.41.11 gd_info() : Retourne des informations à propos de la librairie GD installée.....</u>	538
<u>7.41.12 getimagesize() : Retourne la taille d'une image.....</u>	539
<u>7.41.13 image_type_to_mime_type() : Lit le Mime-Type d'un type d'image.....</u>	540
<u>7.41.14 image2wbmp() : Crée une image WBMP.....</u>	541
<u>7.41.15 imagealphablending() : Modifie le mode de blending d'une image.....</u>	541
<u>7.41.16 imageantialias() : Utiliser ou non les fonctions d'antialias.....</u>	541
<u>7.41.17 imagearc() : Dessine une ellipse partielle.....</u>	541
<u>7.41.18 imagechar() : Dessine un caractère horizontalement.....</u>	542
<u>7.41.19 imagecharup() : Dessine un caractère verticalement.....</u>	542
<u>7.41.20 imagecolorallocate() : Alloue une couleur pour une image.....</u>	542
<u>7.41.21 imagecolorallocatealpha() : Alloue une couleur à une image.....</u>	543
<u>7.41.22 imagecolorat() : Retourne l'index de la couleur d'un pixel donné.....</u>	543
<u>7.41.23 imagecolorclosest() : Retourne l'index de la couleur la plus proche d'une couleur donnée.....</u>	543
<u>7.41.24 imagecolorclosestalpha() : Retourne la couleur la plus proche, en tenant compte du canal alpha.....</u>	544
<u>7.41.25 imagecolorclosesthwb() : Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir (hue, white and blackness).....</u>	544
<u>7.41.26 imagecolordeallocate() : Supprime une couleur d'une image.....</u>	544
<u>7.41.27 imagecolorexact() : Retourne l'index de la couleur donnée.....</u>	544
<u>7.41.28 imagecolorexactalpha() : Retourne l'index d'une couleur avec son canal alpha.....</u>	545
<u>7.41.29 imagecolormatch() : Fait correspondre un peu plus les couleurs de la version palette d'une image aux couleurs de sa version truecolor.....</u>	545
<u>7.41.30 imagecolorresolve() : Retourne l'index de la couleur donnée, ou la plus proche possible.....</u>	545
<u>7.41.31 imagecolorresolvealpha() : Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha.....</u>	545
<u>7.41.32 imagecolorset() : Change la couleur dans une palette à l'index donné.....</u>	546
<u>7.41.33 imagecolorsforindex() : Retourne la couleur associée à un index.....</u>	546
<u>7.41.34 imagecolorstotal() : Calcule le nombre de couleurs d'une palette.....</u>	546
<u>7.41.35 imagecolortransparent() : Définit la couleur transparente.....</u>	546
<u>7.41.36 imagecopy() : Copie une partie d'une image.....</u>	546
<u>7.41.37 imagecopymerge() : Copie et fusionne une partie d'une image.....</u>	547
<u>7.41.38 imagecopymergegray() : Copie et fusionne une partie d'une image en niveaux de gris.....</u>	547
<u>7.41.39 imagecopyresampled() : Copie, redimensionne, rééchantillonne une image.....</u>	547
<u>7.41.40 imagecopyresized() : Copie et redimensionne une partie d'une image.....</u>	548
<u>7.41.41 imagecreate() : Crée une nouvelle image à palette.....</u>	548
<u>7.41.42 imagecreatefromgd() : Crée une nouvelle image à partir d'un fichier GD ou d'une URL.....</u>	549
<u>7.41.43 imagecreatefromgd2() : Crée une nouvelle image à partir d'un fichier GD2</u>	

Sommaire

7 Index des fonctions

<u>ou d'une URL.....</u>	549
<u>7.41.44 imagecreatefromgd2part() : Crée une nouvelle image à partir d'une partie de fichier GD2 ou d'une URL.....</u>	549
<u>7.41.45 imagecreatefromgif() : Crée une nouvelle image à partir d'un fichier ou d'une URL.....</u>	549
<u>7.41.46 imagecreatefromjpeg() : Crée une nouvelle image JPEG à partir d'un fichier ou d'une URL.....</u>	550
<u>7.41.47 imagecreatefrompng() : Crée une nouvelle image PNG à partir d'un fichier ou d'une URL.....</u>	550
<u>7.41.48 imagecreatefromstring() : Crée une image à partir d'une chaîne.....</u>	551
<u>7.41.49 imagecreatefromwbmp() : Crée une image depuis un fichier WBMP.....</u>	551
<u>7.41.50 imagecreatefromxbm() : Crée une image à partir d'un fichier XBM.....</u>	552
<u>7.41.51 imagecreatefromxpm() : Crée une image à partir d'un fichier XPM.....</u>	552
<u>7.41.52 imagecreatetruecolor() : Crée une nouvelle image en vraies couleurs.....</u>	552
<u>7.41.53 imagedashedline() : Dessine une ligne pointillée.....</u>	553
<u>7.41.54 imagedestroy() : détruit une image.....</u>	553
<u>7.41.55 imageellipse() : Dessine une ellipse.....</u>	553
<u>7.41.56 imagefill() : Remplit.....</u>	553
<u>7.41.57 imagefilledarc() : Dessine une ellipse partielle et la remplit.....</u>	553
<u>7.41.58 imagefilledellipse() : Dessine une ellipse pleine.....</u>	554
<u>7.41.59 imagefilledpolygon() : Dessine un polygone rempli.....</u>	554
<u>7.41.60 imagefilledrectangle() : Dessine un rectangle rempli.....</u>	554
<u>7.41.61 imagefilltoborder() : Remplit avec une région avec une couleur spécifique.....</u>	554
<u>7.41.62 imagefontheight() : Retourne la hauteur de la police.....</u>	555
<u>7.41.63 imagefontwidth() : Retourne la largeur de la police.....</u>	555
<u>7.41.64 imageftbbox() : Calcule le rectangle d'encadrement pour un texte, en utilisant la police courante et freetype2.....</u>	555
<u>7.41.65 imagefttext() : Ecrit du texte dans une image avec la police courante et FreeType 2.....</u>	555
<u>7.41.66 imagegammacorrect() : Applique une correction gamma à l'image.....</u>	555
<u>7.41.67 imagegd() : Génère une image au format GD, vers le navigateur ou un fichier.....</u>	555
<u>7.41.68 imagegd2() : Génère une image au format GD2, vers le navigateur ou un fichier.....</u>	556
<u>7.41.69 imagegif() : Envoie une image GIF vers un navigateur ou un fichier.....</u>	556
<u>7.41.70 imageinterlace() : Active ou désactive l'entrelacement.....</u>	557
<u>7.41.71 imageistruecolor() : Détermine si une image est une image truecolor.....</u>	557
<u>7.41.72 imagejpeg() : Envoie une image JPEG vers un navigateur ou un fichier.....</u>	557
<u>7.41.73 imageline() : Dessine une ligne.....</u>	558
<u>7.41.74 imageloadfont() : Charge une nouvelle police.....</u>	558
<u>7.41.75 imagepalettecopy() : Copie la palette d'une image à l'autre.....</u>	558
<u>7.41.76 imagepng() : Envoie une image PNG vers un navigateur ou un fichier.....</u>	558
<u>7.41.77 imagepolygon() : Dessine un polygone.....</u>	559
<u>7.41.78 imagesbbox() : Retourne le rectangle entourant un texte et dessiné avec une police PostScript Type1.....</u>	559
<u>7.41.79 imagepscopyfont() : Fait une copie d'une police chargée pour modification ultérieure.....</u>	560
<u>7.41.80 imagepsencodefont() : Change le codage vectoriel d'un caractère dans une police.....</u>	560
<u>7.41.81 imagepsextendfont() : Etend ou condense une police de caractères.....</u>	560
<u>7.41.82 imagepsfreefont() : Libère la mémoire occupée par une police PostScript</u>	

Sommaire

7 Index des fonctions

<u>Type 1</u>	560
<u>7.41.83 imagepsloadfont() : Charge une police PostScript Type 1 depuis un fichier</u>	561
<u>7.41.84 imagepsslantfont() : Incline une police de caractères PostScript</u>	561
<u>7.41.85 imagepstext() : Dessine un texte sur une image avec une police</u> <u>PostScript Type1</u>	561
<u>7.41.86 imagerectangle() : Dessine un rectangle</u>	562
<u>7.41.87 imagerotate() : Fait tourner une image d'un angle</u>	562
<u>7.41.88 imagesavealpha() : Définit l'option pour enregistrer les informations</u> <u>complètes du canal alpha lors de la sauvegarde d'images PNG</u>	562
<u>7.41.89 imagesetbrush() : Modifie la brosse pour le dessin des lignes</u>	563
<u>7.41.90 imagesetpixel() : Dessine un pixel</u>	563
<u>7.41.91 imagesetstyle() : Configure le style pour le dessin des lignes</u>	563
<u>7.41.92 imagesetthickness() : Modifie l'épaisseur d'un trait</u>	564
<u>7.41.93 imagesettile() : Modifie l'image utilisée pour le carrelage</u>	564
<u>7.41.94 imagestring() : Dessine une chaîne horizontale</u>	564
<u>7.41.95 imagestringup() : Dessine une chaîne verticale</u>	565
<u>7.41.96 imagesx() : Retourne la largeur d'une image</u>	565
<u>7.41.97 imagesy() : Retourne la hauteur de l'image</u>	565
<u>7.41.98 imagetruecolortopalette() : Convertit une image en vraies couleurs en</u> <u>image à palette</u>	565
<u>7.41.99 imagettfbbox() : Retourne le rectangle entourant un texte et dessiné avec</u> <u>une police TrueType</u>	566
<u>7.41.100 imagetfttext() : Dessine un texte avec une police TrueType</u>	566
<u>7.41.101 imagetypes() : Retourne les types d'images supportés par la version</u> <u>courante de PHP</u>	567
<u>7.41.102 imagewbmp() : Affiche une image WBMP</u>	568
<u>7.41.103 iptcembed() : Intègre des données binaires IPTC dans une image JPEG</u>	568
<u>7.41.104 iptcparse() : Analyse un bloc binaire IPTC http://www.iptc.org/ et</u> <u>recherche les balises simples</u>	568
<u>7.41.105 jpeg2wbmp() : Convertit une image JPEG en image WBMP</u>	568
<u>7.41.106 png2wbmp() : Convertit une image PNG en image WBMP</u>	569
<u>7.41.107 read_exif_data() : Alias de exif_read_data</u>	569
<u>7.42 IMAP</u>	570
<u>7.42.1 Introduction</u>	570
<u>7.42.2 Pré-requis</u>	570
<u>7.42.3 Installation</u>	570
<u>7.42.4 Configuration à l'exécution</u>	571
<u>7.42.5 Types de ressources</u>	571
<u>7.42.6 Constantes prédefinies</u>	571
<u>7.42.7 Voir aussi</u>	572
<u>7.42.9 imap_alerts() : Retourne toutes les alertes</u>	574
<u>7.42.10 imap_append() : Ajoute une chaîne dans une boîte aux lettres</u>	574
<u>7.42.11 imap_base64() : Décode un texte encodé en BASE64</u>	574
<u>7.42.12 imap_binary() : Convertit une chaîne à 8 bits en une chaîne à base64</u>	574
<u>7.42.13 imap_body() : Lit le corps d'un message</u>	575
<u>7.42.14 imap_bodystruct() : Lit la structure d'une section du corps d'un mail</u>	575
<u>7.42.15 imap_check() : Vérifie le courrier de la boîte aux lettres courante</u>	575
<u>7.42.16 imap_clearflag_full() : Supprime un flag sur un message</u>	576
<u>7.42.17 imap_close() : Termine un flût IMAP</u>	576
<u>7.42.18 imap_createmailbox() : Crée une nouvelle boîte aux lettres</u>	576
<u>7.42.19 imap_delete() : Marque le fichier pour l'effacement, dans la boîte aux</u>	

Sommaire

7 Index des fonctions

<u>lettres courante</u>	577
<u>7.42.20 imap_deletemailbox() : Efface une boîte aux lettres</u>	577
<u>7.42.21 imap_errors() : Retourne toutes les erreurs</u>	577
<u>7.42.22 imap_expunge() : Efface tous les messages marqués pour l'effacement</u>	577
<u>7.42.23 imap_fetch_overview() : Lit un sommaire des en-têtes de messages</u>	578
<u>7.42.24 imap_fetchbody() : Retourne une section extraite du corps d'un message</u>	578
<u>7.42.25 imap_fetchheader() : Retourne l'en-tête d'un message</u>	579
<u>7.42.26 imap_fetchstructure() : Lit la structure d'un message</u>	579
<u>7.42.27 imap_get_quota() : Lit les quotas des boîtes aux lettres</u>	580
<u>7.42.28 imap_get_quotaroot() : Lit les quotas de chaque utilisateur</u>	581
<u>7.42.29 imap_getmailboxes() : Liste les boîtes aux lettres, et retourne le détail pour chacune</u>	581
<u>7.42.30 imap_getsubscribed() : Liste toutes les boîtes aux lettres souscrites</u>	582
<u>7.42.31 imap_header() : Lit l'en-tête d'un message</u>	582
<u>7.42.32 imap_headerinfo() : Lit l'en-tête du message</u>	583
<u>7.42.33 imap_headers() : Retourne les en-têtes de tous les messages d'une boîte aux lettres</u>	584
<u>7.42.34 imap_last_error() : Retourne la dernière erreur (si elle existe) qui est survenu lors de la dernière requête</u>	584
<u>7.42.35 imap_list() : Lit la liste des boîtes aux lettres</u>	584
<u>7.42.36 imap_listmailbox() : Liste les boîtes aux lettres</u>	585
<u>7.42.37 imap_listscan() : Lit la liste des boîtes aux lettres, et y recherche une chaîne</u>	585
<u>7.42.38 imap_listsubscribed() : Liste les boîtes aux lettres souscrites</u>	585
<u>7.42.39 imap_lsub() : Liste toutes les boîtes aux lettres enregistrées</u>	585
<u>7.42.40 imap_mail_compose() : Crée un message MIME</u>	585
<u>7.42.41 imap_mail_copy() : Copie les messages spécifiés dans une boîte aux lettres</u>	586
<u>7.42.42 imap_mail_move() : Déplace des messages dans une boîte aux lettres</u>	586
<u>7.42.43 imap_mail() : Envoie un message mail</u>	586
<u>7.42.44 imap_mailboxmsginfo() : Lit les informations à propos de la boîte aux lettres courante</u>	586
<u>7.42.45 imap_mime_header_decode() : Décode les éléments MIME d'une en-tête</u>	587
<u>7.42.46 imap_msgno() : Retourne le numéro de séquence de message pour un UID donné</u>	587
<u>7.42.47 imap_num_msg() : Retourne le nombre de message dans la boîte aux lettres courante</u>	588
<u>7.42.48 imap_num_recent() : Retourne le nombre de messages récents dans la boîte aux lettres courante</u>	588
<u>7.42.49 imap_open() : Ouvre un flôt IMAP vers une boîte aux lettres</u>	588
<u>7.42.50 imap_ping() : Vérifie que le flôt IMAP est toujours actif</u>	589
<u>7.42.51 imap_qprint() : Convertit une chaîne à guillemets en une chaîne à 8 bits</u>	590
<u>7.42.52 imap_renamemailbox() : Renomme une boîte aux lettres</u>	590
<u>7.42.53 imap_reopen() : Ouvre un flôt IMAP vers une nouvelle boîte aux lettres</u>	590
<u>7.42.54 imap_rfc822_parse_adrlist() : Analyse une adresse email</u>	590
<u>7.42.55 imap_rfc822_parse_headers() : Analyse un en-tête mail</u>	591
<u>7.42.56 imap_rfc822_write_address() : Retourne une adresse email formatée correctement</u>	591
<u>7.42.57 imap_scanmailbox() : Lit la liste des boîtes aux lettres, et y recherche une chaîne</u>	591
<u>7.42.58 imap_search() : Retourne un tableau de message après recherche</u>	592

Sommaire

7 Index des fonctions

7.42.59	<u>imap_set_quota()</u> : Modifie le quota d'une boîte aux lettres.....	592
7.42.60	<u>imap_setacl()</u> : Modifie le ACL de la boîte aux lettres.....	593
7.42.61	<u>imap_setflag_full()</u> : Positionne un flag sur un message.....	593
7.42.62	<u>imap_sort()</u> : Trie des messages.....	594
7.42.63	<u>imap_status()</u> : Retourne les informations de statut sur une boîte aux lettres autres que la boîte courante.....	594
7.42.64	<u>imap_subscribe()</u> : Souscrit à une boîte aux lettres.....	595
7.42.65	<u>imap_thread()</u> : Retourne l'arbre de REFERENCE organisé par thread.....	595
7.42.66	<u>imap_uid()</u> : Retourne l'UID d'un message.....	595
7.42.67	<u>imap_undelete()</u> : Enlève la marque d'effacement d'un message.....	595
7.42.68	<u>imap_unsubscribe()</u> : Termine la souscription à une boîte aux lettres.....	595
7.42.69	<u>imap_utf7_decode()</u> : Décode une chaîne modifiée UTF-7.....	596
7.42.70	<u>imap_utf7_encode()</u> : Convertit des données 8bit en texte UTF-7.....	596
7.42.71	<u>imap_utf8()</u> : Convertit du texte au format MIME en UTF8.....	596
7.43	<u>Informix</u>	597
7.43.1	<u>Introduction</u>	597
7.43.2	<u>Pré-requis</u>	597
7.43.3	<u>Installation</u>	597
7.43.4	<u>Configuration à l'exécution</u>	597
7.43.5	<u>Types de ressources</u>	600
7.43.6	<u>Constantes prédefinies</u>	600
7.43.8	<u>ifx_blobinfile_mode()</u> : Choisit le mode par défaut des objets BLOB pour toutes les requêtes SELECT.....	601
7.43.9	<u>ifx_bytestvarchar()</u> : Choisit le mode par défaut des objets BYTE.....	601
7.43.10	<u>ifx_close()</u> : Ferme une connexion à un serveur Informix.....	601
7.43.11	<u>ifx_connect()</u> : Ouvre une connexion à un serveur Informix.....	602
7.43.12	<u>ifx_copy_blob()</u> : Duplique un objet BLOB.....	602
7.43.13	<u>ifx_create_blob()</u> : Crée un objet BLOB.....	602
7.43.14	<u>ifx_create_char()</u> : Crée un objet char.....	602
7.43.15	<u>ifx_do()</u> : Exécute une requête SQL préparée.....	603
7.43.16	<u>ifx_error()</u> : Retourne le code d'erreur de la dernière requête Informix.....	603
7.43.17	<u>ifx_errormsg()</u> : Retourne le message d'erreur de la dernière requête Informix.....	604
7.43.18	<u>ifx_fetch_row()</u> : Retourne une ligne sous la forme d'un tableau énuméré.....	604
7.43.19	<u>ifx_fieldproperties()</u> : Liste les propriétés des champs SQL.....	605
7.43.20	<u>ifx_fieldtypes()</u> : Liste les champs Informix SQL.....	605
7.43.21	<u>ifx_free_blob()</u> : Supprime un objet BLOB.....	605
7.43.22	<u>ifx_free_char()</u> : Supprime un objet char.....	606
7.43.23	<u>ifx_free_result()</u> : Libère les ressources prises par un résultat.....	606
7.43.24	<u>ifx_get_blob()</u> : Retourne le contenu d'un objet BLOB.....	606
7.43.25	<u>ifx_get_char()</u> : Retourne le contenu d'un objet char.....	606
7.43.26	<u>ifx_getsqlca()</u> : Retourne le contenu de la variable sqlca.sqlerrd[0..5] après une requête.....	606
7.43.27	<u>ifx_htmltbl_result()</u> : Lit toutes les lignes d'un tableau, et la met sous la forme d'un tableau HTML.....	607
7.43.28	<u>ifx_nullformat()</u> : Modifie le mode par défaut de lecture des valeurs.....	607
7.43.29	<u>ifx_num_fields()</u> : Retourne le nombre de colonnes dans une requête.....	607
7.43.30	<u>ifx_num_rows()</u> : Compte le nombre de ligne déjà lues dans un résultat.....	608
7.43.31	<u>ifx_pconnect()</u> : Ouvre une connexion persistante à un serveur Informix.....	608
7.43.32	<u>ifx_prepare()</u> : Prépare une requête SQL pour l'exécution.....	608
7.43.33	<u>ifx_query()</u> : Envoie une requête Informix.....	609

Sommaire

7 Index des fonctions

<u>7.43.34 ifx_textasvarchar()</u> : Choisit le mode par défaut des objets text.....	610
<u>7.43.35 ifx_update_blob()</u> : Modifie le contenu d'un objet BLOB.....	610
<u>7.43.36 ifx_update_char()</u> : Modifie le contenu d'un objet char.....	610
<u>7.43.37 ifxus_close_slob()</u> : Ferme un objet SLOB.....	610
<u>7.43.38 ifxus_create_slob()</u> : Crée un objet SLOB et l'ouvre.....	610
<u>7.43.39 ifxus_free_slob()</u> : Supprime un objet SLOB.....	611
<u>7.43.40 ifxus_open_slob()</u> : Ouvre un objet SLOB.....	611
<u>7.43.41 ifxus_read_slob()</u> : Lit n bytes d'un objet SLOB.....	611
<u>7.43.42 ifxus_seek_slob()</u> : Fixe le fichier courant, ou la position courante.....	611
<u>7.43.43 ifxus_tell_slob()</u> : Retourne le fichier courant, ou la position courante.....	611
<u>7.43.44 ifxus_write_slob()</u> : Ecrit une chaîne dans un objet SLOB.....	611
<u>7.44 InterBase</u>	613
<u>7.44.1 Introduction</u>	613
<u>7.44.2 Pré-requis</u>	613
<u>7.44.3 Installation</u>	613
<u>7.44.4 Configuration à l'exécution</u>	613
<u>7.44.5 Types de ressources</u>	614
<u>7.44.6 Constantes prédefinies</u>	614
<u>7.44.8 ibase_affected_rows()</u> : Retourne le nombre de lignes affectées par la dernière requête iBase.....	615
<u>7.44.9 ibase_blob_add()</u> : Ajoute des données dans un BLOB iBase fraîchement crée.....	615
<u>7.44.10 ibase_blob_cancel()</u> : Annule la création d'un BLOB iBase.....	615
<u>7.44.11 ibase_blob_close()</u> : Ferme un BLOB iBase.....	615
<u>7.44.12 ibase_blob_create()</u> : Crée un BLOB iBase pour ajouter des données.....	616
<u>7.44.13 ibase_blob_echo()</u> : Affiche le contenu d'un BLOB iBase au navigateur.....	616
<u>7.44.14 ibase_blob_get()</u> : Lit len octets de données dans un BLOB iBase ouvert....	616
<u>7.44.15 ibase_blob_import()</u> : Créé un BLOB iBase, y copie un fichier et le referme.....	617
<u>7.44.16 ibase_blob_info()</u> : Retourne la taille d'un BLOB iBase et d'autres informations utiles.....	617
<u>7.44.17 ibase_blob_open()</u> : Ouvre un BLOB iBase pour récupérer des parties de données.....	617
<u>7.44.18 ibase_close()</u> : Ferme une connexion à une base de données Interbase.....	617
<u>7.44.19 ibase_commit_ret()</u> : Valide une transaction iBase sans la refermer.....	617
<u>7.44.20 ibase_commit()</u> : Valide une transaction iBase.....	618
<u>7.44.21 ibase_connect()</u> : Ouvre une connexion à une base de données InterBase.....	618
<u>7.44.22 ibase_delete_user()</u> : Efface un utilisateur d'une base de données de sécurité (uniquement pour IB6 ou plus récent).....	619
<u>7.44.23 ibase_drop_db()</u> : Supprime une base de données iBase.....	619
<u>7.44.24 ibase_errcode()</u> : Retourne le code d'erreur iBase.....	619
<u>7.44.25 ibase_errmsg()</u> : Retourne un message d'erreur.....	619
<u>7.44.26 ibase_execute()</u> : Exécute une requête iBase préparée.....	619
<u>7.44.27 ibase_fetch_assoc()</u> : Récupère une ligne du résultat d'une requête dans un tableau associatif.....	620
<u>7.44.28 ibase_fetch_object()</u> : Lit une ligne dans une base Interbase dans un objet..	620
<u>7.44.29 ibase_fetch_row()</u> : Lit une ligne dans une base Interbase.....	620
<u>7.44.30 ibase_field_info()</u> : Lit les informations sur un champ iBase.....	621
<u>7.44.31 ibase_free_event_handler()</u> : Libère un gestionnaire d'événements iBase....	621
<u>7.44.32 ibase_free_query()</u> : Libère la mémoire réservée par une requête	

Sommaire

7 Index des fonctions

<u>préparée</u>	621
<u>7.44.33 ibase_free_result() : Libère un résultat iBase</u>	622
<u>7.44.34 ibase_modify_user() : Modifie un utilisateur dans une base de données de sécurité (uniquement pour InterBase6 ou plus récent)</u>	622
<u>7.44.35 ibase_name_result() : Assigne un nom à un jeu de résultats iBase</u>	622
<u>7.44.36 ibase_num_fields() : Retourne le nombre de colonnes dans un résultat iBase</u>	622
<u>7.44.37 ibase_num_params() : Retourne le nombre de paramètres dans une requête préparée iBase</u>	623
<u>7.44.38 ibase_param_info() : Retourne des informations à propos d'un paramètre dans une requête préparée iBase</u>	623
<u>7.44.39 ibase_pconnect() : Ouvre une connexion persistante à une base de données InterBase</u>	623
<u>7.44.40 ibase_prepare() : Prépare une requête iBase pour lier les paramètres et l'exécuter ultérieurement</u>	624
<u>7.44.41 ibase_query() : Exécute une requête sur une base iBase</u>	624
<u>7.44.42 ibase_rollback_ret() : Annule une transaction sans la refermer</u>	624
<u>7.44.43 ibase_rollback() : Annule une transaction iBase</u>	625
<u>7.44.44 ibase_set_event_handler() : Enregistre une fonction de callback sur événement iBase</u>	625
<u>7.44.45 ibase_timefmt() : Fixe le format de date pour les prochaines requêtes</u>	625
<u>7.44.46 ibase_trans() : Prépare une transaction iBase</u>	626
<u>7.44.47 ibase_wait_event() : Attend un événement iBase</u>	626
<u>7.45 Ingres II</u>	627
<u>7.45.1 Introduction</u>	627
<u>7.45.2 Pré-requis</u>	627
<u>7.45.3 Installation</u>	627
<u>7.45.4 Configuration à l'exécution</u>	627
<u>7.45.5 Types de ressources</u>	628
<u>7.45.6 Constantes prédefinies</u>	628
<u>7.45.8 ingres_close() : Ferme une connexion à un serveur Ingres</u>	628
<u>7.45.9 ingres_commit() : Valide une transaction</u>	629
<u>7.45.10 ingres_connect() : Ouvre une connexion à un serveur Ingres</u>	629
<u>7.45.11 ingres_fetch_array() : Récupère une ligne de résultat Ingres dans un tableau</u>	630
<u>7.45.12 ingres_fetch_object() : Récupère une ligne de résultat dans un objet</u>	631
<u>7.45.13 ingres_fetch_row() : Récupère une ligne de résultat dans un tableau énuméré</u>	631
<u>7.45.14 ingres_field_length() : Retourne la taille d'un champ</u>	632
<u>7.45.15 ingres_field_name() : Retourne le nom d'un champ dans le résultat d'une requête</u>	632
<u>7.45.16 ingres_field_nullable() : Teste si un champ est annulable</u>	632
<u>7.45.17 ingres_field_precision() : Retourne la précision d'un champ</u>	633
<u>7.45.18 ingres_field_scale() : Retourne l'échelle d'un champ</u>	633
<u>7.45.19 ingres_field_type() : Retourne le type d'un champ dans le résultat d'une requête</u>	633
<u>7.45.20 ingres_num_fields() : Retourne le nombre de champs renvoyés par la dernière requête</u>	634
<u>7.45.21 ingres_num_rows() : Retourne le nombre de lignes affectées ou retournées par la dernière requête</u>	634
<u>7.45.22 ingres_pconnect() : Ouvre une connexion persistante à un serveur Ingres</u>	635

Sommaire

7 Index des fonctions

7.45.23 <u>ingres_query()</u> : Envoie une requête SQL à un serveur Ingres II.....	635
7.45.24 <u>ingres_rollback()</u> : Annule une transaction.....	636
<u>7.46 IRC</u>	637
7.46.1 <u>Introduction</u>	637
7.46.2 <u>Installation</u>	637
7.46.3 <u>Configuration à l'exécution</u>	637
7.46.4 <u>Types de ressources</u>	637
7.46.5 <u>Constantes prédefinies</u>	637
7.46.7 <u>ircg_disconnect()</u> : Ferme la connexion avec un serveur.....	638
7.46.8 <u>ircg_fetch_error_msg()</u> : Retourne le message d'erreur de la dernière opération IRCG.....	638
7.46.9 <u>ircg_get_username()</u> : Lit le nom de l'utilisateur de la connexion.....	638
7.46.10 <u>ircg_html_encode()</u> : Prépare l'affichage pour le HTML.....	639
7.46.11 <u>ircg_ignore_add()</u> : Ajoute un utilisateur sur la liste des utilisateurs indésirables.....	639
7.46.12 <u>ircg_ignore_del()</u> : Supprime un utilisateur de la liste des utilisateurs indésirables.....	639
7.46.13 <u>ircg_is_conn_alive()</u> : Vérifie l'état de la connexion.....	639
7.46.14 <u>ircg_join()</u> : Rejoint un canal IRC.....	639
7.46.15 <u>ircg_kick()</u> : Expulse un utilisateur d'un canal.....	639
7.46.16 <u>ircg_lookup_format_messages()</u> : Sélectionne un format d'affichage pour les messages IRC.....	640
7.46.17 <u>ircg_msg()</u> : Envoie un message à un canal ou un utilisateur.....	640
7.46.18 <u>ircg_nick()</u> : Change de nom sur le serveur.....	640
7.46.19 <u>ircg_nickname_escape()</u> : Prépare les caractères spéciaux d'un nick pour être compatible IRC.....	640
7.46.20 <u>ircg_nickname_unescape()</u> : Decodes encoded nickname.....	640
7.46.21 <u>ircg_notice()</u> : Envoie une note (notice) à un utilisateur.....	641
7.46.22 <u>ircg_part()</u> : Quitte le canal.....	641
7.46.23 <u>ircg_pconnect()</u> : Connecte à un serveur IRC.....	641
7.46.24 <u>ircg_register_format_messages()</u> : Enregistre un nouveau format d'affichage des messages IRC.....	641
7.46.25 <u>ircg_set_current()</u> : Prépare la connexion courante pour l'affichage.....	642
7.46.26 <u>ircg_set_file()</u> : Configure le fichier d'historique de la connexion.....	642
7.46.27 <u>ircg_set_on_die()</u> : Configure les opérations à exécuter lorsque la connexion s'interrompt.....	643
7.46.28 <u>ircg_topic()</u> : Modifie le sujet (topic) d'un canal.....	643
7.46.29 <u>ircg_whois()</u> : Requiert les informations sur un utilisateur.....	643
<u>7.47 Java</u>	644
7.47.1 <u>Introduction</u>	644
7.47.2 <u>Pré-requis</u>	644
7.47.3 <u>Installation</u>	644
7.47.4 <u>Configuration à l'exécution</u>	644
7.47.5 <u>Types de ressources</u>	644
7.47.6 <u>Constantes prédefinies</u>	645
7.47.7 <u>Exemples</u>	645
7.47.8 <u>Java Servlet SAPI</u>	646
7.47.10 <u>java_last_exception_get()</u> : Lit la dernière exception Java.....	646
<u>7.48 LDAP</u>	648
7.48.1 <u>Introduction</u>	648
7.48.2 <u>Pré-requis</u>	649

Sommaire

7 Index des fonctions

<u>7.48.3 Installation</u>	649
<u>7.48.4 Configuration à l'exécution</u>	649
<u>7.48.5 Types de ressources</u>	649
<u>7.48.6 Constantes prédéfinies</u>	649
<u>7.48.7 Exemples</u>	650
<u>7.48.8 Utiliser les fonctions LDAP de PHP</u>	650
<u>7.48.10 ldap_add() : Ajoute une entrée dans un dossier LDAP</u>	652
<u>7.48.11 ldap_bind() : Authentification au serveur LDAP</u>	652
<u>7.48.12 ldap_close() : Ferme une connexion au serveur LDAP</u>	653
<u>7.48.13 ldap_compare() : Compare une entrée avec des valeurs d'attributs</u>	653
<u>7.48.14 ldap_connect() : Se connecte à un serveur LDAP</u>	654
<u>7.48.15 ldap_count_entries() : Compte le nombre d'entrées après une recherche</u>	655
<u>7.48.16 ldap_delete() : Efface une entrée dans un dossier</u>	655
<u>7.48.17 ldap_dn2ufn() : Convertit un DN en format UFN (User Friendly Naming)</u>	655
<u>7.48.18 ldap_err2str() : Convertit un numéro d'erreur LDAP en message d'erreur</u>	655
<u>7.48.19 ldap_errno() : Retourne le numéro d'erreur LDAP de la dernière commande exécutée</u>	656
<u>7.48.20 ldap_error() : Retourne le message LDAP de la dernière commande LDAP</u>	656
<u>7.48.21 ldap_explode_dn() : Sépare les différents composants d'un DN</u>	657
<u>7.48.22 ldap_first_attribute() : Return first attribute</u>	657
<u>7.48.23 ldap_first_entry() : Retourne la première entrée</u>	657
<u>7.48.24 ldap_first_reference() : Return first reference</u>	657
<u>7.48.25 ldap_free_result() : Free result memory</u>	658
<u>7.48.26 ldap_get_attributes() : Lit les attributs d'une entrée</u>	658
<u>7.48.27 ldap_get_dn() : Lit le DN d'une entrée</u>	659
<u>7.48.28 ldap_get_entries() : Lit toutes les entrées du résultat</u>	659
<u>7.48.29 ldap_get_option() : Lit/écrit la valeur courante d'une option</u>	659
<u>7.48.30 ldap_get_values_len() : Lit toutes les valeurs binaires d'une entrée</u>	660
<u>7.48.31 ldap_get_values() : Lit toutes les valeurs d'une entrée LDAP</u>	660
<u>7.48.32 ldap_list() : Recherche dans un niveau</u>	661
<u>7.48.33 ldap_mod_add() : Ajoute un attribut à l'entrée courante</u>	661
<u>7.48.34 ldap_mod_del() : Efface un attribut à l'entrée courante</u>	662
<u>7.48.35 ldap_mod_replace() : Remplace un attribut dans l'entrée courante</u>	662
<u>7.48.36 ldap_modify() : Modifie une entrée LDAP</u>	662
<u>7.48.37 ldap_next_attribute() : Lit l'attribut suivant</u>	662
<u>7.48.38 ldap_next_entry() : Lit la prochaine entrée</u>	662
<u>7.48.39 ldap_next_reference() : Lit la référence suivante</u>	663
<u>7.48.40 ldap_parse_reference() : Extrait les informations d'une référence d'entrée</u>	663
<u>7.48.41 ldap_parse_result() : Extrait des informations d'un résultat</u>	663
<u>7.48.42 ldap_read() : Lit une entrée</u>	663
<u>7.48.43 ldap_rename() : Modifie le nom d'une entrée</u>	664
<u>7.48.44 ldap_search() : Recherche sur le serveur LDAP</u>	664
<u>7.48.45 ldap_set_option() : Modifie la valeur d'une option LDAP</u>	665
<u>7.48.46 ldap_set_rebind_proc() : Configure une fonction de callback pour refaire des liaisons, lors de recherche de référents</u>	666
<u>7.48.47 ldap_sort() : Trie les entrées d'un résultat LDAP</u>	666
<u>7.48.48 ldap_start_tls() : Démarre TLS</u>	667
<u>7.48.49 ldap_t61_to_8859() : Convertit les caractères t6 en caractères 8859</u>	667
<u>7.48.50 ldap_unbind() : Déconnecte d'un serveur LDAP</u>	667
<u>7.49 Mail</u>	668

Sommaire

7 Index des fonctions

<u>7.49.1 Introduction</u>	668
<u>7.49.2 Pré-requis</u>	668
<u>7.49.3 Installation</u>	668
<u>7.49.4 Configuration à l'exécution</u>	668
<u>7.49.5 Types de ressources</u>	669
<u>7.49.6 Constantes prédefinies</u>	669
<u>7.49.8 mail() : Envoie de mail</u>	669
<u>7.50 Traitement de email</u>	672
<u>7.50.1 Introduction</u>	672
<u>7.50.2 Installation</u>	672
<u>7.50.4 mailparse_msg_create() : Retourne un analyseur destiné au traitement d'un message</u>	672
<u>7.50.5 mailparse_msg_extract_part_file() : Extrait et décode une section de message</u>	672
<u>7.50.6 mailparse_msg_extract_part() : Extrait et décode une section de message. Si la fonction de callback n'est pas spécifiée, le contenu sera envoyé à la sortie standard</u>	673
<u>7.50.7 mailparse_msg_free() : Détruit un analyseur créé par mailparse_msg_create</u>	673
<u>7.50.8 mailparse_msg_get_part_data() : Retourne un tableau associatif avec des informations sur le message</u>	673
<u>7.50.9 mailparse_msg_get_part() : Retourne une ressource sur une section d'un message MIME</u>	674
<u>7.50.10 mailparse_msg_get_structure() : Retourne un tableau avec les différentes sections MIME du message</u>	674
<u>7.50.11 mailparse_msg_parse_file() : Analyse un fichier et retourne une ressource représentant la structure du fichier</u>	674
<u>7.50.12 mailparse_msg_parse() : Analyse incrémentalement des données dans un buffer</u>	674
<u>7.50.13 mailparse_rfc822_parse_addresses() : Analyse les adresses et retourne un tableau contenant ces données</u>	675
<u>7.50.14 mailparse_stream_encode() : Lit les données dans un fichier, applique l'encodage et envoie le résultat à destfp</u>	675
<u>7.50.15 mailparse_uudecode_all() : Scanne les données du fichier et extrait tous les fichiers encodés qui s'y trouvent. Retourne un tableau contenant la liste des fichiers</u>	675
<u>7.51 Mathématiques</u>	676
<u>7.51.1 Introduction</u>	676
<u>7.51.2 Pré-requis</u>	676
<u>7.51.3 Installation</u>	676
<u>7.51.4 Configuration à l'exécution</u>	676
<u>7.51.5 Types de ressources</u>	676
<u>7.51.6 Constantes prédefinies</u>	676
<u>7.51.8 acos() : arc cosinus</u>	678
<u>7.51.9 acosh() : Arc cosinus hyperbolique</u>	678
<u>7.51.10 asin() : arc sinus</u>	678
<u>7.51.11 asinh() : Arc sinus hyperbolique</u>	678
<u>7.51.12 atan() : arc tangent</u>	678
<u>7.51.13 atan2() : arc tangent de deux variables</u>	679
<u>7.51.14 atanh() : Arc tangeant hyperbolique</u>	679
<u>7.51.15 base_convert() : Convertit un nombre entre des bases arbitraires</u>	679

Sommaire

7 Index des fonctions

<u>7.51.16 bindec() : Convertit de binaire en décimal</u>	679
<u>7.51.17 ceil() : Arrondit au nombre supérieur</u>	680
<u>7.51.18 cos() : cosinus</u>	680
<u>7.51.19 cosh() : Cosinus hyperbolic</u>	680
<u>7.51.20 decbin() : Convertit de décimal en binaire</u>	680
<u>7.51.21 dechex() : Convertit de décimal en hexadécimal</u>	680
<u>7.51.22 decoct() : Convertit de décimal en octal</u>	681
<u>7.51.23 deg2rad() : Convertit un nombre de degrés en radians</u>	681
<u>7.51.24 exp() : Exponentielle</u>	681
<u>7.51.25 expm1() : Retourne le résultat de $\exp(\text{number}) - 1$, calculé de manière précise, même si val est proche de 0</u>	681
<u>7.51.26 floor() : Arrondit à l'entier inférieur</u>	681
<u>7.51.27 fmod() : Retourne le reste de la division</u>	682
<u>7.51.28 getrandmax() : Plus grande valeur aléatoire possible</u>	682
<u>7.51.29 hexdec() : Convertit de hexadécimal en décimal</u>	682
<u>7.51.30 hypot() : Retourne le résultat de $\sqrt{\text{num1}^2 + \text{num2}^2}$</u>	683
<u>7.51.31 is_finite() : Indique si un nombre est fini</u>	683
<u>7.51.32 is_infinite() :</u>	683
<u>7.51.33 is_nan() : Indique si une valeur est un nombre</u>	683
<u>7.51.34 lcg_value() : Générateur de congruence combinée linéaire</u>	683
<u>7.51.35 log() : Logarithme naturel (népérien)</u>	684
<u>7.51.36 log10() : logarithme en base 10</u>	684
<u>7.51.37 log1p() : Retourne le résultat de $\log(1 + \text{number})$, calculé de manière précise, même si val est proche de zéro</u>	684
<u>7.51.38 max() : La plus grande valeur</u>	684
<u>7.51.39 min() : La plus petite valeur</u>	685
<u>7.51.40 mt_getrandmax() : La plus grand valeur aléatoire possible</u>	685
<u>7.51.41 mt_rand() : Génère une valeur aléatoire (meilleure méthode)</u>	686
<u>7.51.42 mt_srand() : Initialise une valeur aléatoire (meilleure méthode)</u>	686
<u>7.51.43 octdec() : Convertit d'octal en décimal</u>	686
<u>7.51.44 pi() : Retourne la valeur de pi</u>	687
<u>7.51.45 pow() : Expression exponentielle</u>	687
<u>7.51.46 rad2deg() : Convertit de radians en degrés</u>	687
<u>7.51.47 rand() : Génère une valeur aléatoire</u>	688
<u>7.51.48 round() : Arrondi</u>	688
<u>7.51.49 sin() : Sinus</u>	688
<u>7.51.50 sinh() : Sinyus hyperbolique</u>	689
<u>7.51.51 sqrt() : Racine carrée</u>	689
<u>7.51.52 srand() : Initialise le générateur de nombres aléatoires</u>	689
<u>7.51.53 tan() : Tangente</u>	689
<u>7.51.54 tanh() : Tangente hyperbolique</u>	690
<u>7.52 Chaînes de caractères multi-octets</u>	691
<u>7.52.1 Introduction</u>	691
<u>7.52.2 Installation</u>	692
<u>7.52.3 Configuration à l'exécution</u>	692
<u>7.52.4 Types de ressources</u>	694
<u>7.52.5 Constantes prédefinies</u>	694
<u>7.52.6 Entrées/Sorties HTTP</u>	694
<u>7.52.7 Jeux de caractères supportés</u>	695
<u>7.52.8 Exploitation des chaînes multi-octets en PHP</u>	696
<u>7.52.9 Cas des caractères japonais</u>	696

Sommaire

7 Index des fonctions

<u>7.52.10 Références</u>	697
<u>7.52.12 mb_convert_encoding() : Conversion d'encodage</u>	698
<u>7.52.13 mb_convert_kana() : Convertit entre les différents "kana"</u>	699
<u>7.52.14 mb_convert_variables() : Convertit l'encodage de variables</u>	699
<u>7.52.15 mb_decode_mimeheader() : Décode une en-tête MIME</u>	700
<u>7.52.16 mb_decode_numericentity() : Décode les entités HTML en caractères</u>	700
<u>7.52.17 mb_detect_encoding() : Détecte un encodage</u>	700
<u>7.52.18 mb_detect_order() : Lit/modifie l'ordre de détection des encodages</u>	701
<u>7.52.19 mb_encode_mimeheader() : Encode une chaîne pour une en-tête MIME</u>	701
<u>7.52.20 mb_encode_numericentity() : Encode des entités HTML</u>	702
<u>7.52.21 mb_ereg_match() : Expression régulière POSIX pour les chaînes multi-octets</u>	703
<u>7.52.22 mb_ereg_replace() : Remplace des segments de chaînes, avec le support des expressions régulières multi-octet</u>	703
<u>7.52.23 mb_ereg_search_getpos() : Retourne l'offset du début du prochain segment repéré par une expression régulière</u>	704
<u>7.52.24 mb_ereg_search_getregs() : Lit le dernier segment de chaîne multi-octets qui correspond au masque</u>	704
<u>7.52.25 mb_ereg_search_init() : Configure les chaînes et les expressions régulières pour le support des caractères multi-octets</u>	704
<u>7.52.26 mb_ereg_search_pos() : Retourne la position et la longueur du segment de chaîne qui vérifie le masque de l'expression régulière</u>	705
<u>7.52.27 mb_ereg_search_regs() : Retourne le segment de chaîne trouvé par une expression régulière multi-octets</u>	705
<u>7.52.28 mb_ereg_search_setpos() : Choisit le point de départ de la recherche par expression régulière</u>	706
<u>7.52.29 mb_ereg_search() : Recherche par expression régulière multi-octets</u>	706
<u>7.52.30 mb_ereg() : Recherche par expression régulière avec support des caractères multi-octets</u>	707
<u>7.52.31 mb_eregi_replace() : Expression régulière avec support des caractères multi-octets, sans tenir compte de la casse</u>	707
<u>7.52.32 mb_eregi() : Regular expression match ignoring case with multibyte support</u>	707
<u>7.52.33 mb_get_info() : Lit la configuration interne de l'extension mbstring</u>	708
<u>7.52.34 mb_http_input() : Détecte le type d'encodage d'un caractère HTTP</u>	708
<u>7.52.35 mb_http_output() : Lit/modifie l'encodage d'affichage</u>	709
<u>7.52.36 mb_internal_encoding() : Lit/modifie l'encodage interne</u>	709
<u>7.52.37 mb_language() : Lit/modifie le langage courant</u>	709
<u>7.52.38 mb_output_handler() : Fonction de traitement des affichages web</u>	710
<u>7.52.39 mb_parse_str() : Analyse les données HTTP GET/POST/COOKIE et assigne les variables globales</u>	710
<u>7.52.40 mb_preferred_mime_name() : Détecte l'encodage MIME</u>	711
<u>7.52.41 mb_regex_encoding() : Retourne le jeu de caractères courant pour les expressions régulières</u>	711
<u>7.52.42 mb_regex_set_options() : Lit et modifie les options des fonctions d'expression régulières à support de caractères multi-octets</u>	711
<u>7.52.43 mb_send_mail() : Envoie un mail encodé ISO-2022-JP (mail japonais)</u>	712
<u>7.52.44 mb_split() : Scinde une chaîne en tableau avec une expression régulière multi-octets</u>	712
<u>7.52.45 mb_strcut() : Coupe une partie de chaîne</u>	713
<u>7.52.46 mb_strimwidth() : Tronque une chaîne</u>	713

Sommaire

7 Index des fonctions

<u>7.52.47 mb_strlen()</u> : Retourne la taille d'une chaîne.....	713
<u>7.52.48 mb_strpos()</u> : Repère la première occurrence d'un caractère dans une chaîne.....	713
<u>7.52.49 mb_strrpos()</u> : Repère la dernière occurrence d'un caractère dans une chaîne.....	714
<u>7.52.50 mb_strtolower()</u> : Met tous les caractères en minuscules.....	714
<u>7.52.51 mb_strtoupper()</u> : Met tous les caractères en majuscules.....	715
<u>7.52.52 mb_strwidth()</u> : Retourne la largeur d'une chaîne.....	715
<u>7.52.53 mb_strtolower()</u> : Lit/modifie les caractères de substitution.....	716
<u>7.52.54 mb_substr_count()</u> : Compte le nombre d'occurrences d'une sous-chaîne.....	716
<u>7.52.55 mb_substr()</u> : Lit une sous-chaîne.....	716
<u>7.53 MCAL</u>	718
<u>7.53.1 Introduction</u>	718
<u>7.53.2 Pré-requis</u>	718
<u>7.53.3 Installation</u>	718
<u>7.53.4 Configuration à l'exécution</u>	718
<u>7.53.5 Types de ressources</u>	718
<u>7.53.6 Constantes prédefinies</u>	719
<u>7.53.8 mcal_close()</u> : Ferme une connexion MCAL.....	720
<u>7.53.9 mcal_create_calendar()</u> : Crée un nouveau calendrier.....	720
<u>7.53.10 mcal_date_compare()</u> : Compare deux dates.....	720
<u>7.53.11 mcal_date_valid()</u> : Valide une date.....	721
<u>7.53.12 mcal_day_of_week()</u> : Le jour de la semaine.....	721
<u>7.53.13 mcal_day_of_year()</u> : Le jour de l'année.....	721
<u>7.53.14 mcal_days_in_month()</u> : Retourne le nombre de jour dans un mois.....	721
<u>7.53.15 mcal_delete_calendar()</u> : Efface un calendrier.....	721
<u>7.53.16 mcal_delete_event()</u> : Efface un événement dans un calendrier MCAL.....	721
<u>7.53.17 mcal_event_add_attribute()</u> : Ajoute un attribut et une valeur à la structure globale.....	722
<u>7.53.18 mcal_event_init()</u> : Initialise la structure globale d'un flôt MCAL.....	722
<u>7.53.19 mcal_event_set_alarm()</u> : Fixe l'alarme de la structure globale.....	722
<u>7.53.20 mcal_event_set_category()</u> : Fixe la catégorie de la structure globale.....	722
<u>7.53.21 mcal_event_set_class()</u> : Fixe la classe de la structure globale.....	722
<u>7.53.22 mcal_event_set_description()</u> : Fixe la description de la structure globale.....	722
<u>7.53.23 mcal_event_set_end()</u> : Fixe la date de fin de la structure globale.....	723
<u>7.53.24 mcal_event_set_recur_daily()</u> : Fixe la récurrence quotidienne.....	723
<u>7.53.25 mcal_event_set_recur_monthly_mday()</u> : Fixe la récurrence.....	723
<u>7.53.26 mcal_event_set_recur_monthly_wday()</u> : Fixe la récurrence mensuelle.....	723
<u>7.53.27 mcal_event_set_recur_none()</u> : Supprime la récurrence de la structure globale.....	723
<u>7.53.28 mcal_event_set_recur_weekly()</u> : Fixe la récurrence hebdomadaire.....	724
<u>7.53.29 mcal_event_set_recur_yearly()</u> : Fixe la récurrence annuelle.....	724
<u>7.53.30 mcal_event_set_start()</u> : Fixe la date de début de la structure globale.....	724
<u>7.53.31 mcal_event_set_title()</u> : Fixe le titre de la structure globale.....	724
<u>7.53.32 mcal_expunge()</u> : Supprime tous les événements marqués pour l'effacement.....	724
<u>7.53.33 mcal_fetch_current_stream_event()</u> : Retourne un objet contenant la structure de date.....	724
<u>7.53.34 mcal_fetch_event()</u> : Recherche un événement dans le calendrier.....	725
<u>7.53.35 mcal_is_leap_year()</u> : Vérifie que l'année est bissextile.....	726
<u>7.53.36 mcal_list_alarms()</u> : Retourne une liste d'événements qui ont une alarme	

Sommaire

7 Index des fonctions

<u>prévue à une date</u>	726
<u>7.53.37 mcal_list_events() : Retourne une liste d'événement entre deux dates</u>	726
<u>7.53.38 mcal_next_recurrence() : Retourne la prochaine occurrence d'un événement</u>	727
<u>7.53.39 mcal_open() : Ouvre une connexion MCAL</u>	727
<u>7.53.40 mcal_popen() : Ouvre une connexion persistante MCAL</u>	727
<u>7.53.41 mcal_rename_calendar() : Renomme un calendrier</u>	727
<u>7.53.42 mcal_reopen() : Réouvre une connexion MCAL</u>	727
<u>7.53.43 mcal_snooze() : Eteind l'alarme d'un événement</u>	727
<u>7.53.44 mcal_store_event() : Modifie un événement dans un calendrier MCAL</u>	728
<u>7.53.45 mcal_time_valid() : Valide une heure</u>	728
<u>7.53.46 mcal_week_of_year() : Retourne le numéro de semaine dans l'année pour une date donnée</u>	728
<u>7.54 Chiffage mcrypt</u>	729
<u>7.54.1 Introduction</u>	729
<u>7.54.2 Pré-requis</u>	729
<u>7.54.3 Installation</u>	729
<u>7.54.4 Configuration à l'exécution</u>	729
<u>7.54.5 Types de ressources</u>	729
<u>7.54.6 Constantes prédefinies</u>	729
<u>7.54.7 Mcrypt ciphers</u>	730
<u>7.54.8 Exemples</u>	731
<u>7.54.10 mcrypt_cfb() : Chiffre/déchiffre des données en mode CFB</u>	732
<u>7.54.11 mcrypt_create_iv() : Crée un vecteur d'initialisation à partir d'une source aléatoire</u>	733
<u>7.54.12 mcrypt_decrypt() : Déchiffre un texte</u>	733
<u>7.54.13 mcrypt_ecb() : Chiffre/déchiffre des données en mode ECB</u>	734
<u>7.54.14 mcrypt_enc_get_algorithms_name() : Retourne le nom de l'algorithme</u>	734
<u>7.54.15 mcrypt_enc_get_block_size() : Retourne la taille de blocs d'un algorithme</u>	734
<u>7.54.16 mcrypt_enc_get_iv_size() : Retourne la taille du VI d'un algorithme</u>	734
<u>7.54.17 mcrypt_enc_get_key_size() : Retourne la taille maximale de la clé pour un mode</u>	734
<u>7.54.18 mcrypt_enc_get_modes_name() : Retourne le nom du mode</u>	735
<u>7.54.19 mcrypt_enc_get_supported_key_sizes() : Retourne un tableau contenant les tailles de clés acceptées par un algorithme</u>	735
<u>7.54.20 mcrypt_enc_is_block_algorithm_mode() : Teste le chiffage par blocs d'un mode</u>	735
<u>7.54.21 mcrypt_enc_is_block_algorithm() : Teste le chiffage par blocs d'un algorithme</u>	735
<u>7.54.22 mcrypt_enc_is_block_mode() : Teste si le mode retourne les données par blocs</u>	736
<u>7.54.23 mcrypt_enc_self_test() : Teste un module ouvert</u>	736
<u>7.54.24 mcrypt_encrypt() : Chiffre un texte</u>	736
<u>7.54.25 mcrypt_generic_deinit() : Prépare le module pour le déchargement</u>	737
<u>7.54.26 mcrypt_generic_end() : Termine un chiffage</u>	737
<u>7.54.27 mcrypt_generic_init() : Initialise tous les buffers nécessaires</u>	737
<u>7.54.28 mcrypt_generic() : Chiffre</u>	737
<u>7.54.29 mcrypt_get_block_size() : Retourne la taille de blocs d'un chiffement</u>	738
<u>7.54.30 mcrypt_get_cipher_name() : Lit le nom du chiffement utilisé</u>	738
<u>7.54.31 mcrypt_get_iv_size() : Retourne la taille du VI utilisé par un couple chiffement/mode</u>	738

Sommaire

7 Index des fonctions

<u>7.54.32 mcrypt_get_key_size()</u> : Retourne la taille de la clé d'un chiffrement.....	739
<u>7.54.33 mcrypt_list_algorithms()</u> : Liste tous les algorithmes de chiffrement supportés.....	739
<u>7.54.34 mcrypt_list_modes()</u> : Liste tous les modes de chiffrement supportés.....	740
<u>7.54.35 mcrypt_module_close()</u> : Décharge le module de chiffrement.....	740
<u>7.54.36 mcrypt_module_get_algo_block_size()</u> : Retourne la taille de blocs d'un algorithme.....	740
<u>7.54.37 mcrypt_module_get_algo_key_size()</u> : Retourne la taille maximale de clé.....	740
<u>7.54.38 mcrypt_module_get_supported_key_sizes()</u> : Retourne un tableau contenant les tailles de clés supportée par un algorithme de chiffrement.....	741
<u>7.54.39 mcrypt_module_is_block_algorithm_mode()</u> : Indique si un mode fonctionne par blocs.....	741
<u>7.54.40 mcrypt_module_is_block_algorithm()</u> : Indique si un algorithme fonctionne par blocs.....	741
<u>7.54.41 mcrypt_module_is_block_mode()</u> : Indique si un mode travaille par blocs.....	741
<u>7.54.42 mcrypt_module_open()</u> : Ouvre le module de l'algorithme et le mode à utiliser.....	741
<u>7.54.43 mcrypt_module_self_test()</u> : Teste un mode.....	742
<u>7.54.44 mcrypt_ofb()</u> : Chiffre/déchiffre des données en mode OFB.....	743
<u>7.54.45 mdecrypt_generic()</u> : Déchiffre.....	743
<u>7.55 MCVE (Paiement)</u>	744
<u>7.55.1 Introduction</u>	744
<u>7.55.2 Installation</u>	744
<u>7.55.3 Voir aussi</u>	744
<u>7.55.4 Constantes prédefinies</u>	744
<u>7.55.6 mcve_adduserarg()</u> : Ajoute une valeur dans une structure de configuration d'utilisateur.....	750
<u>7.55.7 mcve_bt()</u> : Lit le total des paiements non finalisés en mode batch.....	750
<u>7.55.8 mcve_checkstatus()</u> : Vérifie si une transaction a été complétée.....	750
<u>7.55.9 mcve_chkpwd()</u> : Verifie le mot de passe.....	751
<u>7.55.10 mcve_chngpwd()</u> : Modifie le mot de passe administrateur.....	751
<u>7.55.11 mcve_completeauthorizations()</u> : Retourne le nombre de transactions dans la queue d'autorisation, ainsi que la liste des identifiants.....	751
<u>7.55.12 mcve_connect()</u> : Etablit la connexion avec le système MCVE.....	751
<u>7.55.13 mcve_connectionerror()</u> : Lit un message représentant la raison de l'échec de la connexion.....	751
<u>7.55.14 mcve_deleterespone()</u> : Efface une transaction spécifiée dans une structure MCVE_CONN.....	751
<u>7.55.15 mcve_deletetrans()</u> : Efface une transaction spécifiée dans une structure MCVE_CONN.....	752
<u>7.55.16 mcve_deleteusersetup()</u> : Supprime une structure usersetup.....	752
<u>7.55.17 mcve_deluser()</u> : Efface un compte d'utilisateur MCVE.....	752
<u>7.55.18 mcve_destroyconn()</u> : Termine une connexion MCVE_CONN.....	752
<u>7.55.19 mcve_destroyengine()</u> : Libère la mémoire associée à la connectivité IP/SSL.....	752
<u>7.55.20 mcve_disableuser()</u> : Désactive un utilisateur MCVE actif.....	752
<u>7.55.21 mcve_edituser()</u> : Modifie la structure usersetup d'un utilisateur.....	753
<u>7.55.22 mcve_enableuser()</u> : Active un utilisateur MCVE inactif.....	753
<u>7.55.23 mcve_force()</u> : Envoie une commande FORCE à MCVE. (généralement une autorisation par téléphone).....	753
<u>7.55.24 mcve_getcell()</u> : Lit une cellule spécifique dans une réponse au format	

Sommaire

7 Index des fonctions

<u>CSV, en utilisant le nom de colonne.....</u>	753
<u>7.55.25 mcve_getcellbynum() : Lit une cellule spécifique dans une réponse au format CSV, en utilisant le numéro de colonne.....</u>	753
<u>7.55.26 mcve_getcommadelimited() : Lit une ligne brute, au format CSV retournée par MCVE.....</u>	753
<u>7.55.27 mcve_getheader() : Lit le nom d'une colonne dans une réponse au format CSV.....</u>	754
<u>7.55.28 mcve_getuserarg() : Lit une valeur dans une structure usersetup.....</u>	754
<u>7.55.29 mcve_getuserparam() : Lit un paramètre de réponse utilisateur.....</u>	754
<u>7.55.30 mcve_gft() : Audite MCVE pour les transactions échouées.....</u>	754
<u>7.55.31 mcve_gl() : Audite MCVE pour les transactions finalisées.....</u>	754
<u>7.55.32 mcve_gut() : Audite MCVE pour les transactions non conclues.....</u>	754
<u>7.55.33 mcve_initconn() : Crée et initialise une structure MCVE_CONN.....</u>	755
<u>7.55.34 mcve_initengine() : Prépare le client pour la connexion IP/SSL.....</u>	755
<u>7.55.35 mcve_initusersetup() : Initialise une structure pour stocker des données utilisateurs.....</u>	755
<u>7.55.36 mcve_iscommadelimited() : Vérifie qu'une réponse est au format MVCE.....</u>	755
<u>7.55.37 mcve_liststats() : Liste les statistiques de tous les utilisateurs MCVE.....</u>	755
<u>7.55.38 mcve_listusers() : Liste de tous les utilisateurs du système MCVE.....</u>	755
<u>7.55.39 mcve_maxconntimeout() : La durée maximum de tentative de connexion à MCVE.....</u>	756
<u>7.55.40 mcve_monitor() : Communique avec MCVE (envoie et reçoit des données) sans blocage.....</u>	756
<u>7.55.41 mcve_numcolumns() : Nombre de colonnes retournées dans une ligne au format CSV.....</u>	756
<u>7.55.42 mcve_numrows() : Nombre de lignes retournées dans une réponse au format CSV.....</u>	756
<u>7.55.43 mcve_override() : Envoie une commande OVERRIDE à MCVE.....</u>	756
<u>7.55.44 mcve_parsecommadelimited() : Analyse une au format CSV pour que mcve_getcell et consorts fonctionnent.....</u>	756
<u>7.55.45 mcve_ping() : Envoie une requête de ping à MCVE.....</u>	757
<u>7.55.46 mcve_preauth() : Envoie une commande PREAUTHORIZATION à MCVE.....</u>	757
<u>7.55.47 mcve_preauthcompletion() : Complète une PREAUTHORIZATION et la prépare pour la finalisation.....</u>	757
<u>7.55.48 mcve_gc() : Audite MCVE pour une liste de transactions dans la queue de sortie.....</u>	757
<u>7.55.49 mcve_responseparam() : Lit un paramètre de réponse personnalisé.....</u>	757
<u>7.55.50 mcve_return() : Emet une commande RETURN ou CREDIT vers MCVE.....</u>	758
<u>7.55.51 mcve_returncode() : Lit le code retourné exact de la transaction.....</u>	758
<u>7.55.52 mcve_returnstatus() : Vérifie si la transaction a réussi.....</u>	758
<u>7.55.53 mcve_sale() : Envoie une commande de vente (SALE) à MCVE.....</u>	758
<u>7.55.54 mcve_setblocking() : Configure la connexion en mode bloquant/non bloquant.....</u>	758
<u>7.55.55 mcve_setdropfile() : Configure la connexion en mode Drop-File.....</u>	758
<u>7.55.56 mcve_setip() : Configure la connexion en mode IP.....</u>	759
<u>7.55.57 mcve_setssl_files() : Spécifie le fichier de clé de certificats si le serveur requiert la vérification des certificats du client.....</u>	759
<u>7.55.58 mcve_setssl() : Configure la connexion en mode SSL.....</u>	759
<u>7.55.59 mcve_settimeout() : Configure le temps maximum d'une transaction (par transaction).....</u>	759
<u>7.55.60 mcve_settle() : Emet une résolution pour faire un dépôt batch.....</u>	759

Sommaire

7 Index des fonctions

<u>7.55.61 mcve_text_avs()</u> : Lit une représentation textuelle de return_avs.....	759
<u>7.55.62 mcve_text_code()</u> : Lit une représentation textuelle de return_code.....	760
<u>7.55.63 mcve_text_cv()</u> : Lit une représentation textuelle de return_cv.....	760
<u>7.55.64 mcve_transactionauth()</u> : Lit le numéro d'autorisation de la transaction (alpha-numérique).....	760
<u>7.55.65 mcve_transactionavs()</u> : Lit l'adresse de vérification (Address Verification) dans le statut retourné.....	760
<u>7.55.66 mcve_transactionbatch()</u> : Lit le numéro de batch associé à la transaction.....	760
<u>7.55.67 mcve_transactioncv()</u> : Lit le statut de CVC2/CVV2/CID.....	760
<u>7.55.68 mcve_transactionid()</u> : Lit l'identifiant système unique d'une transaction.....	761
<u>7.55.69 mcve_transactionitem()</u> : Lit le numéro d'ITEM d'une transaction dans un batch.....	761
<u>7.55.70 mcve_transactionssent()</u> : Vérifie si le buffer de sortie est vide ou pas.....	761
<u>7.55.71 mcve_transactiontext()</u> : Lit le message retourné par MCVE ou par l'institution de traitement.....	761
<u>7.55.72 mcve_transinqueue()</u> : Nombre de transactions dans la queue client.....	761
<u>7.55.73 mcve_transnew()</u> : Démarre une nouvelle transaction.....	761
<u>7.55.74 mcve_transparam()</u> : Ajoute un paramètre à une transaction.....	762
<u>7.55.75 mcve_transsend()</u> : Finalise et envoie une transaction.....	762
<u>7.55.76 mcve_ub()</u> : Lit une liste des paiements en attente en mode batch.....	762
<u>7.55.77 mcve_uwait()</u> : Attend x microsecondes.....	762
<u>7.55.78 mcve_verifyconnection()</u> : Configure MCVE pour qu'il pingue ou pas la connexion, afin d'en vérifier l'existence.....	762
<u>7.55.79 mcve_verifysslcert()</u> : Configure MCVE pour qu'il vérifie ou pas le certificat SSL.....	762
<u>7.55.80 mcve_void()</u> : Annule (rend VOID) une transaction dans la queue d'attente.....	763
<u>7.56 Hash</u>	764
<u>7.56.1 Introduction</u>	764
<u>7.56.2 Pré-requis</u>	764
<u>7.56.3 Installation</u>	764
<u>7.56.4 Configuration à l'exécution</u>	764
<u>7.56.5 Types de ressources</u>	764
<u>7.56.6 Constantes prédefinies</u>	764
<u>7.56.7 Exemples</u>	765
<u>7.56.9 mhash_get_block_size()</u> : Retourne la taille de bloc du hash.....	765
<u>7.56.10 mhash_get_hash_name()</u> : Retourne le nom du hash.....	765
<u>7.56.11 mhash_keygen_s2k()</u> : Génère une clé.....	766
<u>7.56.12 mhash()</u> : Calcule un hash.....	766
<u>7.57 Mimetype</u>	767
<u>7.57.1 Introduction</u>	767
<u>7.57.2 Pré-requis</u>	767
<u>7.57.3 Installation</u>	767
<u>7.57.4 Configuration à l'exécution</u>	767
<u>7.57.5 Types de ressources</u>	768
<u>7.57.6 Constantes prédefinies</u>	768
<u>7.58 Microsoft SQL Server</u>	769
<u>7.58.1 Introduction</u>	769
<u>7.58.2 Pré-requis</u>	769
<u>7.58.3 Installation</u>	769
<u>7.58.4 Configuration à l'exécution</u>	769

Sommaire

7 Index des fonctions

<u>7.58.5 Types de ressources.....</u>	770
<u>7.58.6 Constantes prédefinies.....</u>	770
<u>7.58.8 mssql_close() : Ferme une connexion MS SQL Server.....</u>	771
<u>7.58.9 mssql_connect() : Ouvre une connexion à un serveur MS SQL server.....</u>	771
<u>7.58.10 mssql_data_seek() : Déplace le pointeur interne de ligne.....</u>	771
<u>7.58.11 mssql_execute() : Exécute une procédure stockée sur un serveur MS SQL.....</u>	772
<u>7.58.12 mssql_fetch_array() : Lit une ligne de résultat MSSQL dans un tableau.....</u>	772
<u>7.58.13 mssql_fetch_assoc() : Retourne un tableau associatif pour la ligne courant de résultat MS SQL Server.....</u>	772
<u>7.58.14 mssql_fetch_batch() : Retourne le prochain lot de lignes MS SQL Server.....</u>	772
<u>7.58.15 mssql_fetch_field() : Lit les informations sur le champ.....</u>	773
<u>7.58.16 mssql_fetch_object() : Retourne une ligne de résultat MS SQL Server sous la forme d'un objet.....</u>	773
<u>7.58.17 mssql_fetch_row() : Lit une ligne de résultat MSSQL dans un tableau numérique.....</u>	773
<u>7.58.18 mssql_field_length() : Lit la longueur d'un champ MS SQL Server.....</u>	774
<u>7.58.19 mssql_field_name() : Lit le nom d'un champ MS SQL Server.....</u>	774
<u>7.58.20 mssql_field_seek() : Fixe l'offset du pointeur de champ MS SQL Server.....</u>	774
<u>7.58.21 mssql_field_type() : Lit le nom d'un champ MS SQL Server.....</u>	774
<u>7.58.22 mssql_free_result() : Libère la mémoire des ressources MS SQL Server.....</u>	774
<u>7.58.23 mssql_free_statement() : Libère une commande MS SQL Server de la mémoire.....</u>	775
<u>7.58.24 mssql_get_last_message() : Retourne le dernier message d'erreur du serveur.....</u>	775
<u>7.58.25 mssql_guid_string() : Convertit le GUID binaire de 16 octets en une chaîne de caractères.....</u>	775
<u>7.58.26 mssql_init() : Initialise une procédure stockée MS SQL Server locale ou distante.....</u>	775
<u>7.58.27 mssql_min_error_severity() : Fixe le niveau de sévérité des erreurs MS SQL Server.....</u>	775
<u>7.58.28 mssql_min_message_severity() : Fixe le niveau de sévérité des messages d'erreurs MS SQL Server.....</u>	776
<u>7.58.29 mssql_next_result() : Déplace le pointeur interne MS SQL Server au le résultat suivant.....</u>	776
<u>7.58.30 mssql_num_fields() : Retourne le nombre de champs dans un résultat MS SQL Server.....</u>	776
<u>7.58.31 mssql_num_rows() : Retourne le nombre de lignes dans un résultat MSSQL.....</u>	776
<u>7.58.32 mssql_pconnect() : Ouvre une connexion persistante à un serveur MS SQL.....</u>	777
<u>7.58.33 mssql_query() : Envoie une requête SQL au serveur MSSQL.....</u>	777
<u>7.58.34 mssql_result() : Lit les données d'un résultat.....</u>	777
<u>7.58.35 mssql_rows_affected() : Retourne le nombre de ligne affectées par une requête MS SQL Server.....</u>	778
<u>7.58.36 mssql_select_db() : Sélectionne la base de données MS SQL.....</u>	778
<u>7.59 Ming pour Flash.....</u>	779
<u>7.59.1 Introduction.....</u>	779
<u>7.59.2 Pré-requis.....</u>	779
<u>7.59.3 Installation.....</u>	779
<u>7.59.4 Configuration à l'exécution.....</u>	780

Sommaire

7 Index des fonctions

<u>7.59.5 Types de ressources.....</u>	780
<u>7.59.6 Constantes prédefinies.....</u>	780
<u>7.59.7 Classes prédefinies.....</u>	781
<u>7.59.9 <code>ming_setscale()</code> : Configure l'échelle (??).....</u>	783
<u>7.59.10 <code>ming_useswfversion()</code> : Utilise la version de SWF (??).....</u>	783
<u>7.59.11 <code>swfaction()</code> : Crée une nouvelle action.....</u>	783
<u>7.59.12 <code>swfbitmap->getheight()</code> : Retourne la hauteur d'une bitmap.....</u>	789
<u>7.59.13 <code>swfbitmap->getwidth()</code> : Retourne la largeur d'une bitmap.....</u>	789
<u>7.59.14 <code>swfbitmap()</code> : Crée un objet bitmap.....</u>	789
<u>7.59.15 <code>swfbutton_keypress()</code> : Retourne l'option d'actino pour keyPress (char).....</u>	790
<u>7.59.16 <code>swfbutton->addaction()</code> : Ajoute une action au bouton.....</u>	790
<u>7.59.17 <code>swfbutton->addshape()</code> : Ajoute une forme à un bouton.....</u>	791
<u>7.59.18 <code>swfbutton->setaction()</code> : Assigne l'action du bouton.....</u>	791
<u>7.59.19 <code>swfbutton->setdown()</code> : Alias de <code>addShape(shape, SWFBUTTON_DOWN)</code>.....</u>	791
<u>7.59.20 <code>swfbutton->sethit()</code> : Alias de <code>addShape(shape, SWFBUTTON_HIT)</code>.....</u>	792
<u>7.59.21 <code>swfbutton->setover()</code> : Alias de <code>addShape(shape, SWFBUTTON_OVER)</code>.....</u>	792
<u>7.59.22 <code>swfbutton->setup()</code> : Alias de <code>SWFbutton->addShape(shape, SWFBUTTON_UP)</code>.....</u>	792
<u>7.59.23 <code>swfbutton()</code> : Crée un nouveau bouton.....</u>	792
<u>7.59.24 <code>swfdisplayitem->addcolor()</code> : Ajoute une couleur à une transoformation.....</u>	794
<u>7.59.25 <code>swfdisplayitem->move()</code> : Déplace un objet en coordonnées relatives.....</u>	794
<u>7.59.26 <code>swfdisplayitem->moveto()</code> : Déplace un objet en coordonnées globales.....</u>	795
<u>7.59.27 <code>swfdisplayitem->multicolor()</code> : Multiplie la couleur de transformation.....</u>	795
<u>7.59.28 <code>swfdisplayitem->remove()</code> : Supprime un objet d'une animation.....</u>	796
<u>7.59.29 <code>swfdisplayitem->rotate()</code> : Fait tourner une forme relativement.....</u>	796
<u>7.59.30 <code>swfdisplayitem->rotateto()</code> : Tourne un objet en angle absolu.....</u>	796
<u>7.59.31 <code>swfdisplayitem->scale()</code> : Etire un objet relativement.....</u>	797
<u>7.59.32 <code>swfdisplayitem->scaletto()</code> : Etire un objet en coordonnées globales.....</u>	798
<u>7.59.33 <code>swfdisplayitem->setdepth()</code> : Modifie la place en profondeur (z-order).....</u>	798
<u>7.59.34 <code>swfdisplayitem->setname()</code> : Nomme un objet.....</u>	798
<u>7.59.35 <code>swfdisplayitem->setratio()</code> : Modifie le ratio de l'objet.....</u>	799
<u>7.59.36 <code>swfdisplayitem->skewx()</code> : Incline suivant les X relativement.....</u>	799
<u>7.59.37 <code>swfdisplayitem->skewxto()</code> : Incline suivant les X.....</u>	800
<u>7.59.38 <code>swfdisplayitem->skewy()</code> : Incline suivant les Y relativement.....</u>	800
<u>7.59.39 <code>swfdisplayitem->skewyto()</code> : Incline suivant les Y.....</u>	800
<u>7.59.40 <code>swfdisplayitem()</code> : Crée un nouvel objet d'affichage <code>displayitem</code>.....</u>	801
<u>7.59.41 <code>swffill->moveto()</code> : Déplace l'origine de l'objet <code>SWFFill</code>.....</u>	801
<u>7.59.42 <code>swffill->rotateto()</code> : Tourne la forme.....</u>	801
<u>7.59.43 <code>swffill->scaletto()</code> : Modifie l'échelle de la forme.....</u>	802
<u>7.59.44 <code>swffill->skewxto()</code> : Incline (abscisses).....</u>	802
<u>7.59.45 <code>swffill->skewyto()</code> : Incline (ordonnées).....</u>	802
<u>7.59.46 <code>swffill()</code> : Crée un objet de remplissage.....</u>	802
<u>7.59.47 <code>swffont->getwidth()</code> : Retourne la taille de la chaîne.....</u>	803
<u>7.59.48 <code>swffont()</code> : Charge une police.....</u>	803
<u>7.59.49 <code>swfgradient->addentry()</code> : Ajoute une couleur à la liste du gradient.....</u>	803
<u>7.59.50 <code>swfgradient()</code> : Crée un objet gradient.....</u>	804
<u>7.59.51 <code>swfmorph->getshape1()</code> : Sélectionne la forme de départ.....</u>	804
<u>7.59.52 <code>swfmorph->getshape2()</code> : Sélectionne la forme de fin.....</u>	805
<u>7.59.53 <code>swfmorph()</code> : Crée un morphing.....</u>	805
<u>7.59.54 <code>swfmovie->add()</code> : Ajoute un objet dans une animation.....</u>	806

Sommaire

7 Index des fonctions

<u>7.59.55 swfmovie->nextframe()</u> : Passe à l'image suivante.....	806
<u>7.59.56 swfmovie->output()</u> : Envoie votre animation au navigateur.....	806
<u>7.59.57 swfmovie->remove()</u> : Supprime un objet d'une animation.....	807
<u>7.59.58 swfmovie->save()</u> : Sauve dans un fichier.....	807
<u>7.59.59 swfmovie->setbackground()</u> : Modifie la couleur de fond.....	807
<u>7.59.60 swfmovie->setdimension()</u> : Modifie les dimensions de l'animation.....	808
<u>7.59.61 swfmovie->setframes()</u> : Modifie le nombre total d'images dans l'animation.....	808
<u>7.59.62 swfmovie->setrate()</u> : Modifie la vitesse de l'animation.....	808
<u>7.59.63 swfmovie->streammp3()</u> : Envoie un fichier MP3 en streaming.....	808
<u>7.59.64 swfmovie()</u> : Crée un objet 'animation'.....	809
<u>7.59.65 swfshape->addfill()</u> : Ajoute un remplissage plein à la forme Flash.....	809
<u>7.59.66 swfshape->drawcurve()</u> : Dessine une courbe relativement.....	811
<u>7.59.67 swfshape->drawcurveto()</u> : Dessine une courbe.....	811
<u>7.59.68 swfshape->drawline()</u> : Dessine une ligne relativement.....	811
<u>7.59.69 swfshape->drawlineto()</u> : Dessine une ligne.....	812
<u>7.59.70 swfshape->movepen()</u> : Déplace le stylo relativement.....	812
<u>7.59.71 swfshape->movepento()</u> : Déplace le stylo.....	812
<u>7.59.72 swfshape->setleftfill()</u> : Modifie la couleur de rastérisation de gauche.....	812
<u>7.59.73 swfshape->setline()</u> : Modifie le style de ligne de la forme.....	813
<u>7.59.74 swfshape->setrightfill()</u> : Modifie la couleur de rastérisation de droite.....	814
<u>7.59.75 swfshape()</u> : Crée une nouvelle forme.....	814
<u>7.59.76 swfsprite->add()</u> : Ajoute un objet à un sprite.....	815
<u>7.59.77 swfsprite->nextframe()</u> : Va à la prochaine image du sprite.....	815
<u>7.59.78 swfsprite->remove()</u> : Supprime un objet dans un sprite.....	815
<u>7.59.79 swfsprite->setframes()</u> : Fixe le nombre maximum d'image dans le sprite.....	816
<u>7.59.80 swfsprite()</u> : Crée un sprite.....	816
<u>7.59.81 swftext->addstring()</u> : Ajoute du texte.....	817
<u>7.59.82 swftext->getwidth()</u> : Calcule la longueur d'une chaîne.....	817
<u>7.59.83 swftext->moveto()</u> : Déplace le stylo de texte.....	817
<u>7.59.84 swftext->setcolor()</u> : Modifie la couleur de la police.....	817
<u>7.59.85 swftext->setfont()</u> : Sélectionne la police courante.....	818
<u>7.59.86 swftext->setheight()</u> : Modifie la hauteur de la police courante.....	818
<u>7.59.87 swftext->setspacing()</u> : Modifie l'espacement de police.....	818
<u>7.59.88 swftext()</u> : Crée un nouvel objet texte.....	818
<u>7.59.89 swftextfield->addstring()</u> : Ajoute au texte.....	819
<u>7.59.90 swftextfield->align()</u> : Modifie l'alignement du texte.....	819
<u>7.59.91 swftextfield->setbounds()</u> : Sélectionne la largeur et hauteur du champs.....	819
<u>7.59.92 swftextfield->setcolor()</u> : Modifie la couleur du champs texte.....	820
<u>7.59.93 swftextfield->setfont()</u> : Modifie la police du champs.....	820
<u>7.59.94 swftextfield->setheight()</u> : Modifie la hauteur de la police du champs texte.....	820
<u>7.59.95 swftextfield->setindentation()</u> : Modifie l'indentation de la première ligne.....	820
<u>7.59.96 swftextfield->setleftmargin()</u> : Modifie la marge de gauche.....	821
<u>7.59.97 swftextfield->setlinespacing()</u> : Modifie l'espacement de lignes.....	821
<u>7.59.98 swftextfield->setmargins()</u> : Modifie les marges du champs texte Flash.....	821
<u>7.59.99 swftextfield->setname()</u> : Nomme le champs texte.....	821
<u>7.59.100 swftextfield->setrightmargin()</u> : Modifie la marge de droite.....	822
<u>7.59.101 swftextfield()</u> : Crée un nouveau champs texte.....	822
<u>7.60 Fonctions diverses</u>	823
<u>7.60.1 Introduction</u>	823
<u>7.60.2 Pré-requis</u>	823

Sommaire

7 Index des fonctions

<u>7.60.3 Installation</u>	823
<u>7.60.4 Configuration à l'exécution</u>	823
<u>7.60.5 Types de ressources</u>	823
<u>7.60.6 Constantes prédefinies</u>	824
<u>7.60.8 connection_status()</u> : Retourne les bits de status de la connexion HTTP.....	824
<u>7.60.9 connection_timeout()</u> : Indique si le script a expiré.....	824
<u>7.60.10 constant()</u> : Retourne la valeur d'une constante.....	825
<u>7.60.11 define()</u> : Définit une constante.....	825
<u>7.60.12 defined()</u> : Vérifie l'existence d'une constante.....	826
<u>7.60.13 die()</u> : Alias de la fonction exit.....	826
<u>7.60.14 eval()</u> : Exécute une chaîne comme un script PHP.....	826
<u>7.60.15 exit()</u> : Termine le script courant.....	827
<u>7.60.16 get_browser()</u> : Indique de quoi est capable le navigateur client.....	827
<u>7.60.17 highlight_file()</u> : Colorisation de la syntaxe d'un fichier.....	828
<u>7.60.18 highlight_string()</u> : Applique la syntaxe colorisée à du code PHP.....	829
<u>7.60.19 ignore_user_abort()</u> : Active l'interruption de script sur déconnexion du visiteur.....	829
<u>7.60.20 pack()</u> : Compacte des données dans une chaîne binaire.....	830
<u>7.60.21 show_source()</u> : Génère la syntaxe colorisée d'un fichier PHP.....	831
<u>7.60.22 sleep()</u> : Retarde l'exécution.....	831
<u>7.60.23 uniqid()</u> : Génère un identifiant unique.....	831
<u>7.60.24 unpack()</u> : Déconditionne des données depuis une chaîne binaire.....	832
<u>7.60.25 usleep()</u> : Retarde l'exécution en micro-secondes.....	832
<u>7.61 mnoGoSearch</u>	833
<u>7.61.1 Introduction</u>	833
<u>7.61.2 Pré-requis</u>	833
<u>7.61.3 Installation</u>	833
<u>7.61.4 Configuration à l'exécution</u>	833
<u>7.61.5 Types de ressources</u>	834
<u>7.61.6 Constantes prédefinies</u>	834
<u>7.61.8 udm_alloc_agent()</u> : Alloue une session mnoGoSearch.....	836
<u>7.61.9 udm_api_version()</u> : Lit la version des API mnoGoSearch.....	837
<u>7.61.10 udm_cat_list()</u> : Liste toutes les catégories soeurs d'une catégorie.....	837
<u>7.61.11 udm_cat_path()</u> : Lit le chemin de la catégorie courante.....	838
<u>7.61.12 udm_check_charset()</u> : Vérifie si un jeu de caractère est supporté par mnogosearch.....	838
<u>7.61.13 udm_check_stored()</u> : Vérifie la connexion MnoGoSearch avec document stocké.....	839
<u>7.61.14 udm_clear_search_limits()</u> : Annule toutes les limites de recherche MnoGoSearch.....	839
<u>7.61.15 udm_close_stored()</u> : Ferme la connexion MnoGoSearch avec le document enregistré.....	839
<u>7.61.16 udm_crc32()</u> : Retourne la somme de contrôle CRC32 d'une chaîne.....	839
<u>7.61.17 udm_errno()</u> : Numéro d'erreur mnoGoSearch.....	839
<u>7.61.18 udm_error()</u> : Message d'erreur mnoGoSearch.....	839
<u>7.61.19 udm_find()</u> : Effectue une recherche.....	840
<u>7.61.20 udm_free_agent()</u> : Détruit une session mnoGoSearch.....	840
<u>7.61.21 udm_free_ispell_data()</u> : Libère la mémoire allouée pour ispell par MnoGoSearch.....	841
<u>7.61.22 udm_free_res()</u> : Libère un résultat mnoGoSearch.....	841
<u>7.61.23 udm_get_doc_count()</u> : Lit le nombre total de documents dans les bases	

Sommaire

7 Index des fonctions

<u>MnoGoSearch</u>	841
<u>7.61.24 udm_get_res_field()</u> : Lit un champ de résultat mnoGoSearch.....	841
<u>7.61.25 udm_get_res_param()</u> : Lit les paramètres de résultats mnoGoSearch.....	842
<u>7.61.26 udm_load_ispell_data()</u> : Charge les données ispell.....	842
<u>7.61.27 udm_open_stored()</u> : Ouvre une connexion MnoGoSearch avec un document stocké.....	844
<u>7.61.28 udm_set_agent_param()</u> : Modifie les paramètres de l'agent MnoGoSearch.....	844
<u>7.62 mSQL</u>	848
<u>7.62.1 Introduction</u>	848
<u>7.62.2 Pré-requis</u>	848
<u>7.62.3 Installation</u>	848
<u>7.62.4 Configuration à l'exécution</u>	848
<u>7.62.5 Types de ressources</u>	848
<u>7.62.6 Constantes prédefinies</u>	848
<u>7.62.8 mysql_close()</u> : Ferme une connexion mSQL.....	850
<u>7.62.9 mysql_connect()</u> : Ouvre une connexion mSQL.....	850
<u>7.62.10 mysql_create_db()</u> : Crée une base de données mSQL.....	850
<u>7.62.11 mysql_createdb()</u> : Crée une base de données mSQL.....	850
<u>7.62.12 mysql_data_seek()</u> : Déplace le pointeur interne mSQL.....	851
<u>7.62.13 mysql_dbname()</u> : Lit le nom de la base de données courante.....	851
<u>7.62.14 mysql_drop_db()</u> : Efface une base de données mSQL.....	851
<u>7.62.15 mysql_dropdb()</u> : Efface une base de données mSQL.....	851
<u>7.62.16 mysql_error()</u> : Retourne le message d'erreur mSQL.....	851
<u>7.62.17 mysql_fetch_array()</u> : Lit une ligne sous la forme d'un tableau.....	851
<u>7.62.18 mysql_fetch_field()</u> : Lit la valeur d'un champ mSQL.....	852
<u>7.62.19 mysql_fetch_object()</u> : Lit une ligne sous la forme d'un objet.....	852
<u>7.62.20 mysql_fetch_row()</u> : Retourne une ligne sous la forme d'un objet.....	853
<u>7.62.21 mysql_field_seek()</u> : Fixe d'offset d'un champ mSQL.....	853
<u>7.62.22 mysql_fieldflags()</u> : Retourne le flag d'un champ mSQL.....	853
<u>7.62.23 mysql_fieldlen()</u> : Retourne la longueur d'un champ mSQL.....	853
<u>7.62.24 mysql_fieldname()</u> : Lit le nom d'un champ mSQL.....	853
<u>7.62.25 mysql_fieldtable()</u> : Retourne le nom d'une table à partir d'un nom de champ mSQL.....	854
<u>7.62.26 mysql_fieldtype()</u> : Retourne le type de champs mSQL.....	854
<u>7.62.27 mysql_free_result()</u> : Libère le résultat de la mémoire.....	854
<u>7.62.28 mysql_freeresult()</u> : Libère le résultat de la mémoire.....	854
<u>7.62.29 mysql_list_dbs()</u> : Liste les bases de données mSQL sur un serveur.....	854
<u>7.62.30 mysql_list_fields()</u> : Liste les champs d'une table.....	854
<u>7.62.31 mysql_list_tables()</u> : Liste les tables mSQL sur une base de données.....	855
<u>7.62.32 mysql_listdbs()</u> : Liste les bases de données mSQL sur un serveur.....	855
<u>7.62.33 mysql_listfields()</u> : Liste les champs d'une table.....	855
<u>7.62.34 mysql_listtables()</u> : Liste les tables mSQL sur une base de données.....	855
<u>7.62.35 mysql_num_fields()</u> : Retourne le nombre de champs dans un résultat.....	855
<u>7.62.36 mysql_num_rows()</u> : Retourne le nombre de lignes dans un résultat.....	855
<u>7.62.37 mysql_numfields()</u> : Retourne le nombre de champs dans un résultat.....	856
<u>7.62.38 mysql_numrows()</u> : Retourne le nombre de lignes dans un résultat.....	856
<u>7.62.39 mysql_pconnect()</u> : Ouvre une connexion persistante à un serveur mSQL.....	856
<u>7.62.40 mysql_query()</u> : Envoie une requête mSQL.....	856
<u>7.62.41 mysql_regcase()</u> : Prépare une chaîne pour une recherche par expression régulière insensible à la casse.....	857

Sommaire

7 Index des fonctions

<u>7.62.42 mysql_result()</u> : Retourne les données de résultat.....	857
<u>7.62.43 mysql_select_db()</u> : Sélectionne une base de données mSQL.....	857
<u>7.62.44 mysql_selectdb()</u> : Sélectionne une base de données mSQL.....	858
<u>7.62.45 mysql_tablename()</u> : Retourne le nom d'une table à partir d'un nom de champs.....	858
<u>7.62.46 mysql()</u> : Exécute une requête mSQL.....	858
<u>7.63 MySQL</u>	859
<u>7.63.1 Introduction</u>	859
<u>7.63.2 Pré-requis</u>	859
<u>7.63.3 Installation</u>	859
<u>7.63.4 Configuration à l'exécution</u>	859
<u>7.63.5 Types de ressources</u>	860
<u>7.63.6 Constantes prédefinies</u>	860
<u>7.63.7 Exemples</u>	861
<u>7.63.9 mysql_change_user()</u> : Change le nom de session de l'utilisateur actif.....	862
<u>7.63.10 mysql_client_encoding()</u> : Retourne le nom du jeu de caractères utilisé par le client MySQL.....	863
<u>7.63.11 mysql_close()</u> : Ferme la connexion MySQL.....	863
<u>7.63.12 mysql_connect()</u> : Ouvre une connexion à un serveur MySQL.....	863
<u>7.63.13 mysql_create_db()</u> : Crée une base de données MySQL.....	864
<u>7.63.14 mysql_data_seek()</u> : Déplace le pointeur interne de résultat.....	865
<u>7.63.15 mysql_db_name()</u> : Lit les noms des bases de données.....	865
<u>7.63.16 mysql_db_query()</u> : Envoie une requête MySQL à un serveur MySQL.....	866
<u>7.63.17 mysql_drop_db()</u> : Efface une base de données MySQL.....	866
<u>7.63.18 mysql_errno()</u> : Retourne le numéro d'erreur de la dernière commande MySQL.....	867
<u>7.63.19 mysql_error()</u> : Retourne le texte associé avec l'erreur générée lors de la dernière requête.....	867
<u>7.63.20 mysql_escape_string()</u> : Protège une chaîne pour la passer à mysql_query().....	868
<u>7.63.21 mysql_fetch_array()</u> : Retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux.....	868
<u>7.63.22 mysql_fetch_assoc()</u> : Lit une ligne de résultats dans un tableau associatif.....	869
<u>7.63.23 mysql_fetch_field()</u> : Retourne les données enregistrées dans une colonne sous forme d'objet.....	871
<u>7.63.24 mysql_fetch_lengths()</u> : Retourne la taille de chaque colonne d'une ligne de résultat.....	872
<u>7.63.25 mysql_fetch_object()</u> : Retourne une ligne de résultat MySQL sous la forme d'un objet.....	872
<u>7.63.26 mysql_fetch_row()</u> : Retourne une ligne de résultat MySQL sous la forme d'un tableau.....	873
<u>7.63.27 mysql_field_flags()</u> : Retourne des détails sur une colonne MySQL.....	873
<u>7.63.28 mysql_field_len()</u> : Retourne la taille d'un champ de résultat MySQL.....	873
<u>7.63.29 mysql_field_name()</u> : Retourne le nom d'une colonne.....	873
<u>7.63.30 mysql_field_seek()</u> : Déplace le pointeur de résultat.....	874
<u>7.63.31 mysql_field_table()</u> : Retourne le nom de la table où se trouve une colonne.....	874
<u>7.63.32 mysql_field_type()</u> : Retourne le type de la colonne courante.....	875
<u>7.63.33 mysql_free_result()</u> : Efface le résultat de la mémoire.....	875
<u>7.63.34 mysql_get_client_info()</u> : Lit les informations sur le client MySQL.....	875
<u>7.63.35 mysql_get_host_info()</u> : Lit les informations sur l'hôte MySQL.....	876

Sommaire

7 Index des fonctions

<u>7.63.36 mysql_get_proto_info()</u> : Lit les informations sur le protocole MySQL.....	876
<u>7.63.37 mysql_get_server_info()</u> : Lit les informations sur le serveur MySQL.....	877
<u>7.63.38 mysql_info()</u> : Lit des informations à propos de la dernière requête.....	877
<u>7.63.39 mysql_insert_id()</u> : Retourne l'identifiant généré par la dernière requête INSERT.....	878
<u>7.63.40 mysql_list_dbs()</u> : Liste les bases de données disponibles sur le serveur MySQL.....	878
<u>7.63.41 mysql_list_fields()</u> : Liste les champs du résultat MySQL.....	879
<u>7.63.42 mysql_list_processes()</u> : Liste les processus MySQL.....	879
<u>7.63.43 mysql_list_tables()</u> : Liste les tables d'une base de données.....	880
<u>7.63.44 mysql_num_fields()</u> : Retourne le nombre de champs d'un résultat MySQL.....	880
<u>7.63.45 mysql_num_rows()</u> : Retourne le nombre de lignes d'un résultat.....	881
<u>7.63.46 mysql_pconnect()</u> : Ouvre une connexion persistante à un serveur MySQL.....	881
<u>7.63.47 mysql_ping()</u> : Pingue le serveur, et se reconnecte au besoin.....	882
<u>7.63.48 mysql_query()</u> : Envoie une requête SQL à un serveur MySQL.....	882
<u>7.63.49 mysql_real_escape_string()</u> : Protège les caractères spéciaux d'une commande SQL.....	883
<u>7.63.50 mysql_result()</u> : Retourne un champ d'un résultat.....	883
<u>7.63.51 mysql_select_db()</u> : Sélectionne une base de données MySQL.....	884
<u>7.63.52 mysql_stat()</u> : Retourne le statut courant du serveur.....	884
<u>7.63.53 mysql_tablename()</u> : Lit le nom de la table qui contient un champ.....	885
<u>7.63.54 mysql_thread_id()</u> : Retourne l'identifiant du thread courant.....	885
<u>7.63.55 mysql_unbuffered_query()</u> : Exécute une requête SQL sans mobiliser les résultats.....	886
<u>7.64 Extension MySQL améliorée.....</u>	887
<u>7.64.1 Introduction.....</u>	887
<u>7.64.2 Pré-requis.....</u>	887
<u>7.64.3 Installation.....</u>	887
<u>7.64.4 Configuration à l'exécution.....</u>	887
<u>7.64.5 Types de ressources.....</u>	888
<u>7.64.6 Constantes prédefinies.....</u>	888
<u>7.64.8 mysqli_autocommit()</u> : Active ou désactive le mode auto-commit.....	891
<u>7.64.9 mysqli_bind_param()</u> : Lie des variables à une requête préparée.....	891
<u>7.64.10 mysqli_bind_result()</u> : Lie des variables à une requête préparée pour le stockage de résultat.....	891
<u>7.64.11 mysqli_change_user()</u> : Change l'utilisateur de la connexion spécifiée.....	892
<u>7.64.12 mysqli_character_set_name()</u> : Retourne le jeu de caractères courant pour la connexion.....	892
<u>7.64.13 mysqli_close()</u> : Ferme une connexion.....	893
<u>7.64.14 mysqli_commit()</u> : Valide la transaction courante.....	893
<u>7.64.15 mysqli_connect()</u> : Ouvre une connexion à un serveur MySQL.....	893
<u>7.64.16 mysqli_data_seek()</u> : Déplace le pointeur interne de résultat.....	894
<u>7.64.17 mysqli_debug()</u> : Effectue des actions de débogage.....	894
<u>7.64.18 mysqli_disable_reads_from_master()</u> :.....	894
<u>7.64.19 mysqli_disable_rpl_parse()</u> :.....	895
<u>7.64.20 mysqli_dump_debug_info()</u> : Ecrit les informations de débogage dans le log.....	895
<u>7.64.21 mysqli_enable_reads_from_master()</u> :.....	895
<u>7.64.22 mysqli_enable_rpl_parse()</u> :.....	895
<u>7.64.23 mysqli_errno()</u> : Retourne un code d'erreur pour la dernière commande	

Sommaire

7 Index des fonctions

<u>MySQL</u>	895
<u>7.64.24 mysqli_error() : Retourne une chaîne décrivant la dernière erreur</u>	896
<u>7.64.25 mysqli_execute() : Exécute une requête préparée</u>	896
<u>7.64.26 mysqli_fetch_array() : Retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux</u>	897
<u>7.64.27 mysqli_fetch_assoc() : Récupère une ligne de résultat sous forme de tableau associatif</u>	898
<u>7.64.28 mysqli_fetch_field_direct() : Récupère les données métas d'un champ unique</u>	899
<u>7.64.29 mysqli_fetch_field() : Retourne le prochain champ dans le jeu de résultats</u>	899
<u>7.64.30 mysqli_fetch_fields() : Retourne un tableau d'objets représentant les champs dans le résultat</u>	899
<u>7.64.31 mysqli_fetch_lengths() : Retourne les longueurs des colonnes de la ligne courante du jeu de résultats</u>	900
<u>7.64.32 mysqli_fetch_object() : Retourne la ligne courante d'un jeu de résultat sous forme d'objet</u>	900
<u>7.64.33 mysqli_fetch_row() : Récupère une ligne de résultat sous forme de tableau indexé</u>	900
<u>7.64.34 mysqli_fetch() : Récupère un résultat à partir d'une requête préparée dans les variables</u>	900
<u>7.64.35 mysqli_field_count() : Retourne le nombre de colonnes pour la dernière requête</u>	901
<u>7.64.36 mysqli_field_seek() : Déplace le pointeur de résultat sur le champ spécifié</u>	901
<u>7.64.37 mysqli_field_tell() : Retourne la position du champ courant dans un pointeur de résultat</u>	901
<u>7.64.38 mysqli_free_result() : Libère la mémoire associée à un résultat</u>	901
<u>7.64.39 mysqli_get_client_info() : Retourne une chaîne contenant la version du client MySQL</u>	901
<u>7.64.40 mysqli_get_host_info() : Retourne une chaîne contenant le type de connexion utilisé</u>	901
<u>7.64.41 mysqli_get_proto_info() : Retourne la version du protocole MySQL utilisé</u>	902
<u>7.64.42 mysqli_get_server_info() : Retourne la version du serveur MySQL</u>	902
<u>7.64.43 mysqli_get_server_version() : Retourne un entier représentant la version du serveur MySQL</u>	902
<u>7.64.44 mysqli_info() : Retourne des informations à propos de la dernière requête exécutée</u>	902
<u>7.64.45 mysqli_init() : Initialise MySQLi et retourne une ressource à utiliser avec mysqli_real_connect</u>	902
<u>7.64.46 mysqli_insert_id() : Retourne l'identifiant automatiquement généré par la dernière requête</u>	903
<u>7.64.47 mysqli_kill() : Demande au serveur de terminer un thread MySQL</u>	903
<u>7.64.48 mysqli_master_query() : Force l'exécution d'une requête sur le maître dans une configuration maître/esclave (master/slave)</u>	903
<u>7.64.49 mysqli_num_fields() : Retourne le nombre de champs dans un résultat</u>	903
<u>7.64.50 mysqli_num_rows() : Retourne le nombre de lignes dans un résultat</u>	903
<u>7.64.51 mysqli_options() : Modifie les options</u>	904
<u>7.64.52 mysqli_param_count() : Retourne le nombre de paramètres pour la requête donnée</u>	904
<u>7.64.53 mysqli_ping() : Ping la connexion au serveur et reconnecte si elle n'existe plus</u>	904
<u>7.64.54 mysqli_prepare_result() :</u>	904

Sommaire

7 Index des fonctions

<u>7.64.55 mysqli_prepare()</u> : Prépare une requête SQL pour l'exécution	904
<u>7.64.56 mysqli_profiler()</u> :	904
<u>7.64.57 mysqli_query()</u> : Performe une requête sur la base de données	905
<u>7.64.58 mysqli_read_query_result()</u> :	905
<u>7.64.59 mysqli_real_connect()</u> : Ouvre une connexion à un serveur MySQL	905
<u>7.64.60 mysqli_real_escape_string()</u> : 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	905
<u>7.64.61 mysqli_real_query()</u> : Exécute une requête SQL	905
<u>7.64.62 mysqli_reload()</u> :	906
<u>7.64.63 mysqli_rollback()</u> :	906
<u>7.64.64 mysqli_rpl_parse_enabled()</u> :	906
<u>7.64.65 mysqli_rpl_probe()</u> :	906
<u>7.64.66 mysqli_rpl_query_type()</u> :	906
<u>7.64.67 mysqli_select_db()</u> : Sélectionne une base de données par défaut pour les requêtes	906
<u>7.64.68 mysqli_send_long_data()</u> :	907
<u>7.64.69 mysqli_send_query()</u> :	907
<u>7.64.70 mysqli_slave_query()</u> : Force l'exécution d'une requête sur un esclave dans une configuration maître/esclave (master/slave)	907
<u>7.64.71 mysqli_ssl_set()</u> :	907
<u>7.64.72 mysqli_stat()</u> : Obtient le statut courant du système	907
<u>7.64.73 mysqli_stmt_affected_rows()</u> :	907
<u>7.64.74 mysqli_stmt_close()</u> : Ferme une requête	908
<u>7.64.75 mysqli_stmt_errno()</u> :	908
<u>7.64.76 mysqli_stmt_error()</u> :	908
<u>7.64.77 mysqli_stmt_store_result()</u> :	908
<u>7.64.78 mysqli_store_result()</u> : Transfère un jeu de résultats à partir de la dernière requête	908
<u>7.64.79 mysqli_thread_id()</u> : Retourne l'identifiant du thread pour la connexion courante	908
<u>7.64.80 mysqli_thread_safe()</u> : Indique si le thread safety est donné ou non	909
<u>7.64.81 mysqli_use_result()</u> : Initialise la récupération d'un jeu de résultats	909
<u>7.64.82 mysqli_warning_count()</u> : Retourne le nombre d'avertissements généré par la dernière requête	909
<u>7.65 Sessions Mohawk</u>	910
<u>7.65.1 Introduction</u>	910
<u>7.65.2 Pré-requis</u>	910
<u>7.65.3 Installation</u>	910
<u>7.65.4 Configuration à l'exécution</u>	910
<u>7.65.5 Types de ressources</u>	910
<u>7.65.6 Constantes prédefinies</u>	910
<u>7.65.8 msession_count()</u> : Compte le nombre de sessions	911
<u>7.65.9 msession_create()</u> : Crée une session	911
<u>7.65.10 msession_destroy()</u> : Détruit la session	911
<u>7.65.11 msession_disconnect()</u> : Ferme la connexion au serveur msession	911
<u>7.65.12 msession_find()</u> : Trouve une valeur	911
<u>7.65.13 msession_get_array()</u> : Lit un tableau de ... (??)	911
<u>7.65.14 msession_get()</u> : Lit une valeur dans une session	912
<u>7.65.15 msession_getdata()</u> : Lit les données ... (??)	912
<u>7.65.16 msession_inc()</u> : Incrémente une valeur dans une session	912

Sommaire

7 Index des fonctions

<u>7.65.17 msession_list()</u> : Liste ... (??)	912
<u>7.65.18 msession_listvar()</u> : Liste les sessions contenant une variable	912
<u>7.65.19 msession_lock()</u> : Verrouille une session	912
<u>7.65.20 msession_plugin()</u> : Appele une fonction d'échappement dans les plugins de personnalité msession	913
<u>7.65.21 msession_randstr()</u> : Lit une chaîne aléatoire	913
<u>7.65.22 msession_set_array()</u> : Modifie un tableau de ... (??)	913
<u>7.65.23 msession_set()</u> : Modifie une valeur dans la session	913
<u>7.65.24 msession_setdata()</u> : Modifie les données (??)	913
<u>7.65.25 msession_timeout()</u> : Lit et/ou modifie la durée de vie de la session	913
<u>7.65.26 msession_uniq()</u> : Lit un identifiant unique	914
<u>7.65.27 msession_unlock()</u> : Déverrouille une session	914
<u>7.66 Muscat</u>	915
<u>7.66.1 Introduction</u>	915
<u>7.66.2 Installation</u>	915
<u>7.66.4 muscat_get()</u> : Lit la réponse de l'API muscat	915
<u>7.66.5 muscat_give()</u> : Envoie une chaîne à l'API de base muscat	915
<u>7.66.6 muscat_setup_net()</u> : Crée une nouvelle session muscat	916
<u>7.66.7 muscat_setup()</u> : Crée une nouvelle session	916
<u>7.67 Réseau</u>	917
<u>7.67.1 Introduction</u>	917
<u>7.67.2 Pré-requis</u>	917
<u>7.67.3 Installation</u>	917
<u>7.67.4 Configuration à l'exécution</u>	917
<u>7.67.5 Types de ressources</u>	917
<u>7.67.6 Constantes prédefinies</u>	917
<u>7.67.8 closelog()</u> : Ferme la connexion à l'historique système	918
<u>7.67.9 debugger_off()</u> : Désactive le debugger interne de PHP	919
<u>7.67.10 debugger_on()</u> : Active le debugger interne de PHP	919
<u>7.67.11 define_syslog_variables()</u> : Initialise toutes les constantes liées au syslog	919
<u>7.67.12 dns_check_record()</u> : Synonyme de checkdnrr	919
<u>7.67.13 dns_get_mx()</u> : Synonyme de getmxrr	919
<u>7.67.14 dns_get_record()</u> : Lit les données DNS associées à un hôte	919
<u>7.67.15 fsockopen()</u> : Ouvre une socket de connexion Internet ou Unix	922
<u>7.67.16 gethostbyaddr()</u> : Retourne le nom d'hôte correspondant à une IP	923
<u>7.67.17 gethostbyname()</u> : Retourne l'adresse IP correspondant à un hôte	924
<u>7.67.18 gethostbyname1()</u> : Retourne la liste d'IP correspondante à un hôte	924
<u>7.67.19 getmxrr()</u> : Retourne les enregistrements MX d'un hôte	925
<u>7.67.20 getprotobyname()</u> : Retourne le numéro de protocole associé à un nom de protocole	925
<u>7.67.21 getprotobynumber()</u> : Retourne le nom de protocole associé à un numéro de protocole	925
<u>7.67.22 getservbyname()</u> : Retourne le numéro de port associé à un service Internet et un protocole	925
<u>7.67.23 getservbyport()</u> : Retourne le service Internet qui correspond au port et protocole	926
<u>7.67.24 ip2long()</u> : Convertit une chaîne contenant une adresse (IPv4) IP numérique en adresse littérale	926
<u>7.67.25 long2ip()</u> : Convertit une adresse IP (IPv4) en adresse IP numérique	926
<u>7.67.26 openlog()</u> : Ouvre la connexion à l'historique système	926
<u>7.67.27 pfsockopen()</u> : Ouvre une socket de connexion Internet ou Unix	

Sommaire

7 Index des fonctions

<u>persistante</u>	927
<u>7.67.28 socket_get_status()</u> : Retourne les informations sur une socket.....	927
<u>7.67.29 socket_set_blocking()</u> : Active/désactive le mode bloquant d'une socket.....	927
<u>7.67.30 socket_set_timeout()</u> : Alias de stream_set_timeout.....	928
<u>7.67.31 syslog()</u> : Génère un message dans l'historique système.....	928
<u>7.68 Ncurses terminal screen control functions</u>	930
<u>7.68.1 Introduction</u>	930
<u>7.68.2 Pré-requis</u>	930
<u>7.68.3 Installation</u>	930
<u>7.68.4 Configuration à l'exécution</u>	930
<u>7.68.5 Types de ressources</u>	931
<u>7.68.6 Constantes prédefinies</u>	931
<u>7.68.8 ncurses_addchnstr()</u> : Ajoute une chaîne de taille donnée à la position courante.....	936
<u>7.68.9 ncurses_addchstr()</u> : Ajoute une chaîne à la position courante.....	936
<u>7.68.10 ncurses_addnstr()</u> : Ajoute une chaîne à la position courante.....	936
<u>7.68.11 ncurses_addstr()</u> : Affiche du texte à la position courante.....	936
<u>7.68.12 ncurses_assume_default_colors()</u> : Définit la couleur 0.....	937
<u>7.68.13 ncurses_attroff()</u> : Désactive les attributs donnés.....	937
<u>7.68.14 ncurses_attron()</u> : Active les attributs suivants.....	937
<u>7.68.15 ncurses_attrset()</u> : Modifie les attributs donnés.....	937
<u>7.68.16 ncurses_baudrate()</u> : Retourne le baudrate du terminal.....	938
<u>7.68.17 ncurses_beep()</u> : Fait beeper le terminal.....	938
<u>7.68.18 ncurses_bkgd()</u> : Définit les propriétés de fond d'écran pour le terminal.....	938
<u>7.68.19 ncurses_bkgdset()</u> : Définit le fond d'écran.....	938
<u>7.68.20 ncurses_border()</u> : Dessine un bord autour de l'écran avec les caractères donnés.....	939
<u>7.68.21 ncurses_bottom_panel()</u> : Place un conteneur visible en bas de la pile.....	939
<u>7.68.22 ncurses_can_change_color()</u> : Vérifie si le terminal peut changer de couleurs.....	939
<u>7.68.23 ncurses_cbreak()</u> : Change la bufferisation d'entrée.....	939
<u>7.68.24 ncurses_clear()</u> : Efface l'écran.....	940
<u>7.68.25 ncurses_clrtoobot()</u> : Efface l'écran depuis la position courante jusqu'au bas de l'écran.....	940
<u>7.68.26 ncurses_clrtoeol()</u> : Efface l'écran depuis la position courante jusqu'à la fin de la ligne.....	940
<u>7.68.27 ncurses_color_content()</u> : Lit la valeur RGB d'une couleur.....	941
<u>7.68.28 ncurses_color_set()</u> : Modifie la couleur de fond et de devant.....	941
<u>7.68.29 ncurses_curs_set()</u> : Modifie l'état du curseur.....	941
<u>7.68.30 ncurses_def_prog_mode()</u> : sauve le mode du terminal.....	941
<u>7.68.31 ncurses_def_shell_mode()</u> : Sauve le mode de terminal (shell).....	942
<u>7.68.32 ncurses_define_key()</u> : Définit un code de clé (keycode).....	942
<u>7.68.33 ncurses_del_panel()</u> : Retire un conteneur de la pile et l'efface (mais pas la fenêtre associée).....	942
<u>7.68.34 ncurses_delay_output()</u> : Retarde l'affichage sur les terminaux utilisant des caractères de remplissage.....	942
<u>7.68.35 ncurses_delch()</u> : Efface le caractère courant et décale le reste de la ligne vers la gauche.....	943
<u>7.68.36 ncurses_deleteln()</u> : Efface la ligne courante, et monte l'écran d'une ligne.....	943
<u>7.68.37 ncurses_delwin()</u> : Efface une fenêtre ncurses.....	943
<u>7.68.38 ncurses_doupdate()</u> : Affiche tout ce qui est prêt, et rafraichit l'écran.....	943

Sommaire

7 Index des fonctions

<u>7.68.39 ncurses_echo()</u> : Active l'écho d'entrée clavier.....	944
<u>7.68.40 ncurses_echochar()</u> : Affiche un caractère et rafraichit l'écran.....	944
<u>7.68.41 ncurses_end()</u> : Cesse l'utilisation de ncurses et efface l'écran.....	944
<u>7.68.42 ncurses_erase()</u> : Efface l'écran du terminal.....	945
<u>7.68.43 ncurses_erasechar()</u> : Lit le caractère qui se fait effacer.....	945
<u>7.68.44 ncurses_filter()</u> : Filtre pour ncurses.....	945
<u>7.68.45 ncurses_flash()</u> : Fait flasher le terminal (visual bell).....	945
<u>7.68.46 ncurses_flushinp()</u> : Vide le buffer d'entrée du clavier.....	946
<u>7.68.47 ncurses_getch()</u> : Lit un caractère sur le clavier.....	946
<u>7.68.48 ncurses_getmaxyx()</u> : Retourne la taille d'une fenêtre.....	946
<u>7.68.49 ncurses_getmouse()</u> : Lit les événements souris.....	946
<u>7.68.50 ncurses_getyx()</u> : Retourne la position courante du curseur pour une fenêtre.....	947
<u>7.68.51 ncurses_halfdelay()</u> : Met le terminal en mode semi-retardé.....	947
<u>7.68.52 ncurses_has_colors()</u> : Vérifie que le que le terminal supporte les couleurs.....	947
<u>7.68.53 ncurses_has_ic()</u> : Vérifie les capacités d'insertion et d'effacement.....	948
<u>7.68.54 ncurses_has_il()</u> : Vérifie les capacités d'insertion et d'effacement.....	948
<u>7.68.55 ncurses_has_key()</u> : Vérifie la présence des touches de fonctions sur le clavier.....	948
<u>7.68.56 ncurses_hide_panel()</u> : Retire un conteneur de la pile, pour le rendre invisible.....	948
<u>7.68.57 ncurses_hline()</u> : Dessine une ligne horizontale à la position courante, en utilisant un caractère et une taille maximale.....	949
<u>7.68.58 ncurses_inch()</u> : Lit le caractère et ses attributs à la position courante.....	949
<u>7.68.59 ncurses_init_color()</u> : Configure une nouvelle valeur RGB pour une couleur.....	949
<u>7.68.60 ncurses_init_pair()</u> : Alloue une paire de couleur.....	949
<u>7.68.61 ncurses_init()</u> : Initialise ncurses.....	950
<u>7.68.62 ncurses_insch()</u> : Insère un caractère et décale le reste de la ligne vers la droite, y compris le caractère courant.....	950
<u>7.68.63 ncurses_insdelln()</u> : Insère des lignes devant la ligne courante en scrollant vers le bas (des nombres négatifs donneront un scroll vers le haut).....	950
<u>7.68.64 ncurses_insertln()</u> : Insère une ligne et décale le reste de l'écran vers le bas.....	950
<u>7.68.65 ncurses_insstr()</u> : Insère une chaîne à la position courante, et décale le reste de la chaîne à droite.....	951
<u>7.68.66 ncurses_instr()</u> : Lit une chaîne depuis le terminal.....	951
<u>7.68.67 ncurses_isendwin()</u> : Ncurses est en mode endwin, des affichages normaux sont possibles.....	951
<u>7.68.68 ncurses_keyok()</u> : Active ou désactive un code de clé (keycode).....	952
<u>7.68.69 ncurses_keypad()</u> : Active ou désactive le keypad.....	952
<u>7.68.70 ncurses_killchar()</u> : Retourne la ligne du caractère actuellement supprimé.....	952
<u>7.68.71 ncurses_longname()</u> : Retourne la description du terminal.....	952
<u>7.68.72 ncurses_meta()</u> : Active/désactive les informations de méta clé 8-bits.....	953
<u>7.68.73 ncurses_mouse_trafo()</u> : Transforme les coordonnées.....	953
<u>7.68.74 ncurses_mouseinterval()</u> : Configure les délais entre les clics de souris.....	953
<u>7.68.75 ncurses_mousemask()</u> : Configure les événements de souris à surveiller.....	953
<u>7.68.76 ncurses_move_panel()</u> : Déplace un conteneur pour que son coin supérieur gauche soit aux coordonnées [startx, starty].....	954
<u>7.68.77 ncurses_move()</u> : Déplace la position d'affichage.....	954
<u>7.68.78 ncurses_mvaddch()</u> : Déplace la position courante et ajoute un caractère.....	954

Sommaire

7 Index des fonctions

<u>7.68.79 ncurses_mvaddchnstr() : Déplace la position et ajoute la chaîne attribuée avec la taille donnée.....</u>	955
<u>7.68.80 ncurses_mvaddchstr() : Déplace la position et ajoute une chaîne attribuée.....</u>	955
<u>7.68.81 ncurses_mvaddnstr() : Déplace la position et ajoute une chaîne de taille donnée.....</u>	955
<u>7.68.82 ncurses_mvaddstr() : Déplace la position et ajoute une chaîne.....</u>	955
<u>7.68.83 ncurses_mvcur() : Déplace immédiatement le curseur.....</u>	956
<u>7.68.84 ncurses_mvdelch() : Déplace la position et efface le caractère, puis décale le reste de la ligne à gauche.....</u>	956
<u>7.68.85 ncurses_mvgetch() : Déplace la position et lit le caractère à la nouvelle position.....</u>	956
<u>7.68.86 ncurses_mvhline() : Choisi une nouvelle position et dessine une ligne horizontale avec le caractère donné, et de taille maximale donnée.....</u>	957
<u>7.68.87 ncurses_mvinch() : Déplace la position et lit le caractère la nouvelle position.....</u>	957
<u>7.68.88 ncurses_mvvline() : Modifie la position et dessine une ligne verticale avec le caractère donné, de taille maximale donnée.....</u>	957
<u>7.68.89 ncurses_mvwaddstr() : Ajoute une chaîne à une nouvelle position dans la fenêtre.....</u>	957
<u>7.68.90 ncurses_napms() : Fait une pause.....</u>	958
<u>7.68.91 ncurses_new_panel() : Crée un nouveau conteneur et l'associe à une fenêtre.....</u>	958
<u>7.68.92 ncurses_newpad() : Crée une nouveau pad (fenêtre).....</u>	958
<u>7.68.93 ncurses_newwin() : Crée une nouvelle fenêtre.....</u>	958
<u>7.68.94 ncurses_nl() : Convertit les nouvelles lignes et retour chariot en saut de ligne.....</u>	958
<u>7.68.95 ncurses_nocbreak() : Passe le terminal en mode cooked.....</u>	959
<u>7.68.96 ncurses_noecho() : Désactive l'écho clavier.....</u>	959
<u>7.68.97 ncurses_nonl() : Ne convertit pas les nouvelles lignes et retour chariot en saut de ligne.....</u>	959
<u>7.68.98 ncurses_nogiflush() : Ne vide pas les buffers sur reception des caractères de signaux.....</u>	960
<u>7.68.99 ncurses_noraw() : Déconfigure le mode brut du terminal.....</u>	960
<u>7.68.100 ncurses_pair_content() : Retourne la valeur RGB d'une couleur.....</u>	960
<u>7.68.101 ncurses_panel_above() : Retourne le conteneur au dessus du conteneur. Si ce dernier est null, retourne le dernier conteneur de la pile.....</u>	960
<u>7.68.102 ncurses_panel_below() : Retour le conteneur sous le conteneur. Si ce dernier est null, le premier conteneur de la pile est retourné.....</u>	960
<u>7.68.103 ncurses_panel_window() : Retourne la fenêtre associée avec un conteneur.....</u>	961
<u>7.68.104 ncurses_pnoutrefresh() : Copie une région depuis un pad dans un écran virtuel.....</u>	961
<u>7.68.105 ncurses_prefresh() : Copie une région depuis un pad dans l'écran virtuel.....</u>	961
<u>7.68.106 ncurses_putp() : ncurses_putp (???).....</u>	961
<u>7.68.107 ncurses_qiflush() : Vide les buffers après détection d'un caractère de signal.....</u>	961
<u>7.68.108 ncurses_raw() : Passe le terminal en mode brut (raw).....</u>	962
<u>7.68.109 ncurses_refresh() : Rafraichit l'écran.....</u>	962
<u>7.68.110 ncurses_replace_panel() : Remplace la fenêtre associée à un conteneur.....</u>	962
<u>7.68.111 ncurses_reset_prog_mode() : Restaure le mode prog sauvé par def_prog_mode.....</u>	962

Sommaire

7 Index des fonctions

<u>7.68.112 ncurses_reset_shell_mode()</u> : Restaure le mode shell mode sauvé ar	
<u>def_shell_mode</u>	963
<u>7.68.113 ncurses_resetty()</u> : Rétablit le terminal sauvé.....	963
<u>7.68.114 ncurses_savetty()</u> : Sauve l'état du terminal.....	963
<u>7.68.115 ncurses_scr_dump()</u> : Sauve un écran dans un fichier.....	963
<u>7.68.116 ncurses_scr_init()</u> : Initialise un écran depuis un fichier de sauvegarde.....	964
<u>7.68.117 ncurses_scr_restore()</u> : Rétablit un écran depuis une fichier de	
<u>sauvegarde</u>	964
<u>7.68.118 ncurses_scr_set()</u> : Configure un écran depuis un fichier de sauvegarde....	964
<u>7.68.119 ncurses_scrll()</u> : Scrolle le contenu de la fenêtre vers le haut ou le bas,	
<u>dans changer la position courante</u>	964
<u>7.68.120 ncurses_show_panel()</u> : Place un conteneur invisible en haut de la pile,	
<u>pour le rendre visible</u>	965
<u>7.68.121 ncurses_slk_attr()</u> : Retourne l'attribut de la clé soft label courante.....	965
<u>7.68.122 ncurses_slk_attroff()</u> : ncurses_slk_attroff.....	965
<u>7.68.123 ncurses_slk_attron()</u> : ncurses_slk_attron.....	965
<u>7.68.124 ncurses_slk_attrset()</u> : ncurses_slk_attrset.....	966
<u>7.68.125 ncurses_slk_clear()</u> : Efface les soft labels de l'écran.....	966
<u>7.68.126 ncurses_slk_color()</u> : Configure la couleur des clés soft label.....	966
<u>7.68.127 ncurses_slk_init()</u> : Initialises les fonctions de clés soft label.....	966
<u>7.68.128 ncurses_slk_noutrefresh()</u> : Copie toutes les clés soft label à l'écran	
<u>virtuel</u>	967
<u>7.68.129 ncurses_slk_refresh()</u> : Copie les clés soft label à l'écran.....	967
<u>7.68.130 ncurses_slk_restore()</u> : Rétablit les clés soft label.....	967
<u>7.68.131 ncurses_slk_set()</u> : Modifie les étiquettes de clé de fonction (function key	
<u>labels)</u>	967
<u>7.68.132 ncurses_slk_touch()</u> : Force l'affichage lorsque ncurses_slk_noutrefresh	
<u>est utilisé</u>	967
<u>7.68.133 ncurses_standend()</u> : Cesse l'utilisation de l'attribut 'standout'.....	968
<u>7.68.134 ncurses_standout()</u> : Commence l'utilisation de l'attribut 'standout'.....	968
<u>7.68.135 ncurses_start_color()</u> : Active les couleurs.....	968
<u>7.68.136 ncurses_termattrs()</u> : Retourne toutes les options supportées par le	
<u>terminal, combinées ensemble par l'opérateur OR</u>	968
<u>7.68.137 ncurses_termname()</u> : Retourne le nom court du terminal.....	969
<u>7.68.138 ncurses_timeout()</u> : Configure le temps d'expiration des des séquences	
<u>clavier spéciales</u>	969
<u>7.68.139 ncurses_top_panel()</u> : Place un conteneur visible sur le haut de la pile.....	969
<u>7.68.140 ncurses_typeahead()</u> : Spécifie un autre descripteur de fichier pour la	
<u>vérification à la volée des données saisies</u>	969
<u>7.68.141 ncurses_ungetch()</u> : Replace un caractère dans la queue d'entrée.....	970
<u>7.68.142 ncurses_ungetmouse()</u> : Ajoute un événement souris dans la queue.....	970
<u>7.68.143 ncurses_update_panels()</u> : Rafraîchit l'écran virtuel pour prendre en	
<u>compte les relations entre les conteneurs de la pile</u>	970
<u>7.68.144 ncurses_use_default_colors()</u> : Assigne la couleur de terminal pour	
<u>l'index -1</u>	971
<u>7.68.145 ncurses_use_env()</u> : Contrôle l'utilisation des informations	
<u>d'environnement concernant la taille du terminal</u>	971
<u>7.68.146 ncurses_use_extended_names()</u> : Controle l'utilisation des noms	
<u>étendus dans les descriptions terminfo</u>	971
<u>7.68.147 ncurses_vidattr()</u> : ncurses_vidattr.....	971
<u>7.68.148 ncurses_vline()</u> : Dessine une ligne verticale à la position courante en	

Sommaire

7 Index des fonctions

<u>utilisant un caractère donnée, pour une taille donnée.....</u>	972
<u>7.68.149 ncurses_waddch() : Ajoute un caractère à la position courante, dans une</u>	
<u>fenêtre, et avance le curseur.....</u>	972
<u>7.68.150 ncurses_waddstr() : Affiche le texte à la position courante dans la</u>	
<u>fenêtre.....</u>	972
<u>7.68.151 ncurses_wattroff() : Désactive les attributs d'une fenêtre.....</u>	972
<u>7.68.152 ncurses_wattron() : Efface les attributs de la fenêtre.....</u>	972
<u>7.68.153 ncurses_wattrset() : Modifie les attributs d'une fenêtre.....</u>	973
<u>7.68.154 ncurses_wborder() : Dessine le bord d'une fenêtre avec un caractère</u>	
<u>qualifié.....</u>	973
<u>7.68.155 ncurses_wclear() : Efface la fenêtre.....</u>	973
<u>7.68.156 ncurses_wcolor_set() : Modifie les paires de couleurs de la fenêtre.....</u>	973
<u>7.68.157 ncurses_werase() : Efface le contenu de la fenêtre.....</u>	973
<u>7.68.158 ncurses_wgetch() : Lit un caractère depuis le clavier (fenêtre).....</u>	973
<u>7.68.159 ncurses_whline() : Dessine une ligne horizontale à partir de la position</u>	
<u>courante, avec un caractère qualifié, et d'une taille maximale.....</u>	974
<u>7.68.160 ncurses_wmouse_trafo() : Transforme les coordonnées d'une fenêtre.....</u>	974
<u>7.68.161 ncurses_wmove() : Déplace le position d'affichage de la fenêtre.....</u>	974
<u>7.68.162 ncurses_wnoutrefresh() : Copie la fenêtre dans l'écran virtuel.....</u>	974
<u>7.68.163 ncurses_wrefresh() : Rafraichit la fenêtre sur l'écran du terminal.....</u>	974
<u>7.68.164 ncurses_wstandend() : Termine le mode standout de la fenêtre.....</u>	975
<u>7.68.165 ncurses_wstandout() : Passe une fenêtre en mode standout.....</u>	975
<u>7.68.166 ncurses_wvline() : Dessine une ligne verticale dans une fenêtre, à la</u>	
<u>position courante, en utilisant un caractère qualifié, avec une taille maximale.....</u>	975
<u>7.69 Lotus Notes.....</u>	976
<u>7.69.1 Introduction.....</u>	976
<u>7.69.3 notes_copy_db() : Crée une note à partir du formulaire form_name.....</u>	976
<u>7.69.4 notes_create_db() : Crée une base Lotus Notes.....</u>	976
<u>7.69.5 notes_create_note() : Crée une note à partir du formulaire form_name.....</u>	977
<u>7.69.6 notes_drop_db() : Supprime une base Lotus Notes.....</u>	977
<u>7.69.7 notes_find_note() : Retourne un identifiant de note, dans le serveur</u>	
<u>database_name. Spécifie le nom de la note.....</u>	977
<u>7.69.8 notes_header_info() : Ouvre le message msg_number dans la boîte aux</u>	
<u>lettres et le serveur spécifié.....</u>	977
<u>7.69.9 notes_list_msgs() : Retourne les notes rangées dans database_name.....</u>	978
<u>7.69.10 notes_mark_read() : Marque la note note_id comme lue par l'utilisateur</u>	
<u>user_name.....</u>	978
<u>7.69.11 notes_mark_unread() : Marque la note note_id comme non lue pour</u>	
<u>l'utilisateur user_name.....</u>	978
<u>7.69.12 notes_nav_create() : Crée un nom de navigateur dans database_name.....</u>	978
<u>7.69.13 notes_search() : Cherche les notes par mots clés dans database_name.....</u>	979
<u>7.69.14 notes_unread() : Retourne l'identifiant de la note non lue pour l'utilisateur</u>	
<u>user_name.....</u>	979
<u>7.69.15 notes_version() : Lit la version de Lotus Notes.....</u>	979
<u>7.70 Netscape Server API.....</u>	980
<u>7.70.1 Introduction.....</u>	980
<u>7.70.2 Installation.....</u>	980
<u>7.70.3 Configuration à l'exécution.....</u>	980
<u>7.70.4 Types de ressources.....</u>	980
<u>7.70.5 Constantes prédefinies.....</u>	980
<u>7.70.6 Voir aussi.....</u>	980

Sommaire

7 Index des fonctions

7.70.8 nsapi_response_headers() : Lit tous les entêtes de réponse HTTP sur Netscape serveur.....	981
7.70.9 nsapi_virtual() : Effectue une sous-requête NSAPI.....	981
7.71 ODBC unifié.....	982
7.71.1 Introduction.....	982
7.71.2 Pré-requis.....	982
7.71.3 Installation.....	982
7.71.4 Configuration à l'exécution.....	983
7.71.5 Types de ressources.....	984
7.71.6 Constantes prédefinies.....	984
7.71.8 odbcc_binmode() : Modifie la gestion des colonnes de données binaires.....	986
7.71.9 odbcc_close_all() : Ferme toutes les connexions ODBC.....	987
7.71.10 odbcc_close() : Ferme une connexion ODBC.....	987
7.71.11 odbcc_columnprivileges() : Liste les colonnes et leurs droits associés.....	987
7.71.12 odbcc_columns() : Liste les colonnes d'une table.....	987
7.71.13 odbcc_commit() : Valide une transaction ODBC.....	988
7.71.14 odbcc_connect() : Connexion à une source.....	988
7.71.15 odbcc_cursor() : Lecture du pointeur de fiche courante (cursorname).....	989
7.71.16 odbcc_data_source() : Retourne des informations sur la connexion courante.....	989
7.71.17 odbcc_do() : Synonyme de odbcc_exec.....	989
7.71.18 odbcc_error() : Lit le dernier code d'erreur.....	989
7.71.19 odbcc_errormsg() : Lit le dernier message d'erreur.....	989
7.71.20 odbcc_exec() : Prépare et exécute une requête SQL.....	990
7.71.21 odbcc_execute() : Exécute une requête SQL préparée.....	990
7.71.22 odbcc_fetch_array() : Lit une ligne de résultat dans un tableau associatif.....	990
7.71.23 odbcc_fetch_into() : Lit une ligne de résultat, et la place dans un tableau.....	991
7.71.24 odbcc_fetch_object() : Lit une ligne de résultat dans un objet.....	992
7.71.25 odbcc_fetch_row() : Lit une ligne de résultat.....	992
7.71.26 odbcc_field_len() : Lit la longueur d'un champ.....	992
7.71.27 odbcc_field_name() : Lit le nom de la colonne.....	992
7.71.28 odbcc_field_num() : Numéro de colonne.....	992
7.71.29 odbcc_field_precision() : Alias de odbcc_field_len.....	993
7.71.30 odbcc_field_scale() : Lit l'échelle d'un champ.....	993
7.71.31 odbcc_field_type() : Type de données d'un champ.....	993
7.71.32 odbcc_foreignkeys() : Liste les clés étrangères.....	993
7.71.33 odbcc_free_result() : Libère les ressources associées à un résultat.....	994
7.71.34 odbcc_gettypeinfo() : Liste les types de données supportés par une source.....	994
7.71.35 odbcc_longreadlen() : Gestion des colonnes de type LONG.....	995
7.71.36 odbcc_next_result() : Vérifie si plusieurs résultats sont disponibles.....	995
7.71.37 odbcc_num_fields() : Nombre de colonnes dans un résultat.....	995
7.71.38 odbcc_num_rows() : Nombre de lignes dans un résultat.....	995
7.71.39 odbcc_pconnect() : Ouvre une connexion persistante à une source de données.....	995
7.71.40 odbcc_prepare() : Prépare une commande pour l'exécution.....	996
7.71.41 odbcc_primarykeys() : Liste les colonnes utilisées dans une clé primaire.....	996
7.71.42 odbcc_procedurecolumns() : Liste les paramètres des procédures.....	996
7.71.43 odbcc_procedures() : Liste les procédures stockées.....	997
7.71.44 odbcc_result_all() : Affiche le résultat sous la forme d'une table HTML.....	997
7.71.45 odbcc_result() : Lit un champ de résultat UODBC.....	997
7.71.46 odbcc_rollback() : Annule une transaction.....	998

Sommaire

7 Index des fonctions

<u>7.71.47</u> <u>odbc_setoption()</u> : Modifie les paramètres ODBC.....	998
<u>7.71.48</u> <u>odbc_specialcolumns()</u> : Retourne l'ensemble optimal de colonnes, qui permet de définir uniquement une ligne dans une table.....	999
<u>7.71.49</u> <u>odbc_statistics()</u> : Calcule des statistiques sur une table.....	999
<u>7.71.50</u> <u>odbc_tableprivileges()</u> : Liste les tables et leurs privilèges.....	1000
<u>7.71.51</u> <u>odbc_tables()</u> : Liste les tables d'une source.....	1000
<u>7.72</u> <u>Aggregation d'objets/Composition de fonctions</u>	1002
<u>7.72.1</u> <u>Introduction</u>	1002
<u>7.72.2</u> <u>Exemples</u>	1003
<u>7.72.4</u> <u>aggregate_methods_by_list()</u> : Aggrège sélectivement les méthodes d'une classe grâce à une liste.....	1006
<u>7.72.5</u> <u>aggregate_methods_by_regexp()</u> : Aggrège sélectivement les méthodes d'une classe grâce à une expression régulière.....	1007
<u>7.72.6</u> <u>aggregate_methods()</u> : Aggrège dynamiquement les méthodes d'une classe à un objet.....	1007
<u>7.72.7</u> <u>aggregate_properties_by_list()</u> : Aggrège sélectivement les propriétés d'une classe grâce à une liste.....	1007
<u>7.72.8</u> <u>aggregate_properties_by_regexp()</u> : Aggrège sélectivement les propriétés d'une classe grâce à une expression régulière.....	1008
<u>7.72.9</u> <u>aggregate_properties()</u> : Aggrège dynamiquement les propriétés d'une classe à un objet.....	1008
<u>7.72.10</u> <u>aggregate()</u> : Aggrège dynamiquement des classes et objets.....	1008
<u>7.72.11</u> <u>aggregation_info()</u> : Retourne un tableau associatif décrivant les classes agrégées.....	1008
<u>7.72.12</u> <u>deaggregate()</u> : Desaggrège un objet.....	1009
<u>7.73</u> <u>Oracle 8</u>	1010
<u>7.73.1</u> <u>Introduction</u>	1010
<u>7.73.2</u> <u>Pré-requis</u>	1010
<u>7.73.3</u> <u>Installation</u>	1011
<u>7.73.4</u> <u>Configuration à l'exécution</u>	1011
<u>7.73.5</u> <u>Types de ressources</u>	1011
<u>7.73.6</u> <u>Constantes prédefinies</u>	1011
<u>7.73.7</u> <u>Exemples</u>	1012
<u>7.73.9</u> <u>ocicancel()</u> : Termine la lecture de ressources Oracle.....	1013
<u>7.73.10</u> <u>ocicloselob()</u> : Ferme un lob Oracle.....	1013
<u>7.73.11</u> <u>ocicollappend()</u> : Ajoute un objet à une collection Oracle.....	1014
<u>7.73.12</u> <u>ocicollassign()</u> : Assigne une collection depuis une autre collection Oracle.....	1014
<u>7.73.13</u> <u>ocicollassignelem()</u> : Assigne l'élément val à une collection Oracle à l'index ndx.....	1014
<u>7.73.14</u> <u>ocicollgetelem()</u> : Lit la valeur d'un élément d'une collection Oracle à l'index ndx.....	1014
<u>7.73.15</u> <u>ocicollmax()</u> : Retourne la plus grande valeur d'une collection Oracle.....	1014
<u>7.73.16</u> <u>ocicollsize()</u> : Retourne la taille d'une collection Oracle.....	1014
<u>7.73.17</u> <u>ocicolltrim()</u> : Supprime num éléments à la fin de la collection Oracle.....	1015
<u>7.73.18</u> <u>ocicolumnisnull()</u> : Teste si la valeur d'une colonne Oracle est NULL.....	1015
<u>7.73.19</u> <u>ocicolumnname()</u> : Retourne le nom d'une colonne Oracle.....	1015
<u>7.73.20</u> <u>ocicolumnprecision()</u> : Lit la précision d'une colonne Oracle.....	1015
<u>7.73.21</u> <u>ocicolumnscale()</u> : Lit la taille d'une colonne Oracle.....	1016
<u>7.73.22</u> <u>ocicolumnsize()</u> : Retourne la taille d'une colonne Oracle.....	1016
<u>7.73.23</u> <u>ocicolumntype()</u> : Retourne le type de données d'une colonne Oracle.....	1016

Sommaire

7 Index des fonctions

<u>7.73.24 ocicolumnntyperaw()</u> : Lit directement le type de colonne Oracle.....	1017
<u>7.73.25 ocicommit()</u> : Valide les transactions Oracle en cours.....	1017
<u>7.73.26 ocidefinebyname()</u> : Utilise une variable PHP pour la phase de définition, dans une commande SELECT Oracle.....	1018
<u>7.73.27 ocierror()</u> : Retourne la dernière erreur de stmt conn global sur Oracle.....	1018
<u>7.73.28 ociexecute()</u> : Exécute une commande SQL Oracle.....	1018
<u>7.73.29 ocifetch()</u> : Lit la prochaine ligne dans le résultat Oracle.....	1019
<u>7.73.30 ocifetchinto()</u> : Retourne la ligne suivante d'un résultat Oracle dans un tableau.....	1019
<u>7.73.31 ocifetchstatement()</u> : Retourne toutes les lignes d'un résultat Oracle.....	1019
<u>7.73.32 ocifreecollection()</u> : Supprime une collection Oracle.....	1020
<u>7.73.33 ocifreecursor()</u> : Libère toutes les ressources réservées par un résultat Oracle.....	1020
<u>7.73.34 ocifreedesc()</u> : Détruit un pointeur de LOB Oracle.....	1020
<u>7.73.35 ocifreestatement()</u> : Libère toutes les ressources occupées par une commande Oracle.....	1021
<u>7.73.36 ociinternaldebug()</u> : Active ou désactive l'affichage des données de débuggage Oracle.....	1021
<u>7.73.37 ociloadlob()</u> : Charge un LOB Oracle.....	1021
<u>7.73.38 ocilogoff()</u> : Déconnecte d'un serveur Oracle.....	1021
<u>7.73.39 ocilogon()</u> : Etablit une connexion à un serveur Oracle.....	1021
<u>7.73.40 ocinewcollection()</u> : Initialise une nouvelle collection.....	1023
<u>7.73.41 ocinewcursor()</u> : Retourne un nouveau pointeur à utiliser pour lier les pointeurs de références.....	1023
<u>7.73.42 ocinewdescriptor()</u> : Initialise un nouveau pointeur vide de LOB/FILE.....	1024
<u>7.73.43 ocinlogon()</u> : Etablit une nouvelle connexion avec un serveur Oracle.....	1025
<u>7.73.44 ocinumcols()</u> : Retourne le nombre de colonnes dans un résultat.....	1026
<u>7.73.45 ociparse()</u> : Analyse une requête SQL avec Oracle.....	1027
<u>7.73.46 ocioplogon()</u> : Ouvre une connexion persistante à un serveur Oracle.....	1027
<u>7.73.47 ocireult()</u> : Retourne la valeur d'une colonne dans un résultat Oracle.....	1027
<u>7.73.48 ocirollback()</u> : Annule les transactions Oracle en cours.....	1027
<u>7.73.49 ocirowcount()</u> : Retourne le nombre de lignes affectées par Oracle.....	1027
<u>7.73.50 ocisavelob()</u> : Sauve un LOB Oracle.....	1028
<u>7.73.51 ocisavelobfile()</u> : Sauve un fichier LOB Oracle.....	1028
<u>7.73.52 ociserverversion()</u> : Retourne une chaîne contenant les informations de version du serveur Oracle.....	1028
<u>7.73.53 ocisetprefetch()</u> : Indique le nombre de lignes qui doivent être pré-lues par Oracle.....	1028
<u>7.73.54 ocistatementtype()</u> : Retourne le type de commande OCI.....	1029
<u>7.73.55 ociwritelobtofile()</u> : Ecrit un LOB Oracle dans un fichier.....	1029
<u>7.73.56 ociwritetemporarylob()</u> : Ecrit un blob Oracle temporaire.....	1029
<u>7.74 OpenSSL</u>	1030
<u>7.74.1 Introduction</u>	1030
<u>7.74.2 Pré-requis</u>	1030
<u>7.74.3 Installation</u>	1030
<u>7.74.4 Configuration à l'exécution</u>	1030
<u>7.74.5 Types de ressources</u>	1031
<u>7.74.6 Constantes prédefinies</u>	1031
<u>7.74.7 Paramètres clés/certificats</u>	1032
<u>7.74.8 Vérification de certificats</u>	1032
<u>7.74.10 openssl_csr_export()</u> : Exporte un CSR vers un fichier ou une variable.....	1033

Sommaire

7 Index des fonctions

<u>7.74.11 openssl_csr_new()</u> : Génère une clé privée et un CSR.....	1033
<u>7.74.12 openssl_csr_sign()</u> : Signe un certificat avec un autre certificat.....	1035
<u>7.74.13 openssl_error_string()</u> : Retourne le message d'erreur OpenSSL.....	1035
<u>7.74.14 openssl_free_key()</u> : Libère les ressources.....	1036
<u>7.74.15 openssl_get_privatekey()</u> : Prépare une clé privée au format PEM.....	1036
<u>7.74.16 openssl_get_publickey()</u> : Extrait une clé publique d'un certificat.....	1036
<u>7.74.17 openssl_open()</u> : Ouvre des données scellées.....	1036
<u>7.74.18 openssl_pkcs7_decrypt()</u> : Déchiffre un message S/MIME.....	1037
<u>7.74.19 openssl_pkcs7_encrypt()</u> : Chiffre un message S/MIME.....	1037
<u>7.74.20 openssl_pkcs7_sign()</u> : Signe un message S/MIME.....	1038
<u>7.74.21 openssl_pkcs7_verify()</u> : Vérifie la signature d'un message S/MIME.....	1039
<u>7.74.22 openssl_pkey_export_to_file()</u> : Sauve une clé au format ASCII dans un fichier.....	1039
<u>7.74.23 openssl_pkey_export()</u> : Lit une représentation exportable de la clé dans une chaîne ou un fichier.....	1040
<u>7.74.24 openssl_pkey_get_private()</u> : Lit une clé privée.....	1040
<u>7.74.25 openssl_pkey_get_public()</u> : Extrait une clé privée d'un certificat, et la prépare.....	1040
<u>7.74.26 openssl_pkey_new()</u> : Génère une nouvelle clé privée.....	1040
<u>7.74.27 openssl_private_decrypt()</u> : Déchiffre des données avec une clé privée.....	1041
<u>7.74.28 openssl_private_encrypt()</u> : Chiffre des données avec une clé privée.....	1041
<u>7.74.29 openssl_public_decrypt()</u> : Déchiffre des données avec une clé publique.....	1041
<u>7.74.30 openssl_public_encrypt()</u> : Chiffre des données avec une clé publique.....	1041
<u>7.74.31 openssl_seal()</u> : Scelle des données.....	1042
<u>7.74.32 openssl_sign()</u> : Signe les données.....	1042
<u>7.74.33 openssl_verify()</u> : Vérifie une signature.....	1043
<u>7.74.34 openssl_x509_check_private_key()</u> : Vérifie si une clé privée correspon à un CERT.....	1043
<u>7.74.35 openssl_x509_checkpurpose()</u> : Vérifie l'usage d'un certificat.....	1044
<u>7.74.36 openssl_x509_export_to_file()</u> : Exporte un CERT vers un fichier ou une variable.....	1044
<u>7.74.37 openssl_x509_export()</u> : Exporte un CERT vers une variable.....	1045
<u>7.74.38 openssl_x509_free()</u> : Libère les ressources prises par un certificat.....	1045
<u>7.74.39 openssl_x509_parse()</u> : Analyse un certificat X509.....	1045
<u>7.74.40 openssl_x509_read()</u> : Analyse un certificat X.509 et retourne une ressource.....	1045
<u>7.75 Oracle</u>	1047
<u>7.75.1 Introduction</u>	1047
<u>7.75.2 Installation</u>	1047
<u>7.75.3 Constantes prédefinies</u>	1047
<u>7.75.5 ora_close()</u> : Ferme un pointeur Oracle.....	1048
<u>7.75.6 ora_columnname()</u> : Retourne le nom de la colonne de résultat Oracle.....	1048
<u>7.75.7 ora_columnsize()</u> : Lit la taille d'une colonne Oracle.....	1048
<u>7.75.8 ora_columntype()</u> : Retourne le type de la colonne de résultat Oracle.....	1048
<u>7.75.9 ora_commit()</u> : Valide une transaction Oracle.....	1048
<u>7.75.10 ora_commitoff()</u> : Désactive la validation automatique Oracle.....	1049
<u>7.75.11 ora_commiton()</u> : Active la validation automatique Oracle.....	1049
<u>7.75.12 ora_do()</u> : Analyse, exécute et lit le résultat d'une requête Oracle.....	1049
<u>7.75.13 ora_error()</u> : Retourne le message d'erreur Oracle.....	1049
<u>7.75.14 ora_errorcode()</u> : Retourne le code d'erreur Oracle.....	1050
<u>7.75.15 ora_exec()</u> : Exécute une commande analysée sur un pointeur Oracle.....	1050

Sommaire

7 Index des fonctions

<u>7.75.16 ora_fetch_into()</u> : Lit une ligne de résultat Oracle dans un tableau.....	1050
<u>7.75.17 ora_fetch()</u> : Retourne une ligne de résultat.....	1050
<u>7.75.18 ora_getcolumn()</u> : Retourne une donnée d'une ligne de résultat Oracle.....	1051
<u>7.75.19 ora_logoff()</u> : Ferme une connexion Oracle.....	1051
<u>7.75.20 ora_logon()</u> : Ouvre une connexion Oracle.....	1051
<u>7.75.21 ora_numcols()</u> : Retourne le nombre de colonnes.....	1052
<u>7.75.22 ora_numrows()</u> : Retourne le nombre de colonnes.....	1052
<u>7.75.23 ora_open()</u> : Ouvre un pointeur Oracle.....	1052
<u>7.75.24 ora_parse()</u> : Analyse une requête SQL avec Oracle.....	1052
<u>7.75.25 ora_plogon()</u> : Ouvre une connexion persistante à Oracle.....	1052
<u>7.75.26 ora_rollback()</u> : Annule une transaction Oracle.....	1052
<u>7.76 Ovrimos SQL</u>	1054
<u>7.76.1 Introduction</u>	1054
<u>7.76.2 Pré-requis</u>	1054
<u>7.76.3 Installation</u>	1054
<u>7.76.4 Configuration à l'exécution</u>	1054
<u>7.76.5 Types de ressources</u>	1054
<u>7.76.6 Exemples</u>	1054
<u>7.76.8 ovrimos_commit()</u> : Valide une transaction Ovrimos.....	1055
<u>7.76.9 ovrimos_connect()</u> : Connexion à un serveur.....	1055
<u>7.76.10 ovrimos_cursor()</u> : Retourne le nom du curseur Ovrimos.....	1056
<u>7.76.11 ovrimos_exec()</u> : Exécute une requête SQL.....	1056
<u>7.76.12 ovrimos_execute()</u> : Exécute une requête Ovrimos préparée.....	1056
<u>7.76.13 ovrimos_fetch_into()</u> : Lit une ligne dans un résultat Ovrimos.....	1056
<u>7.76.14 ovrimos_fetch_row()</u> : Lit une ligne dans un résultat Ovrimos.....	1057
<u>7.76.15 ovrimos_field_len()</u> : Retourne la taille d'une colonne Ovrimos.....	1057
<u>7.76.16 ovrimos_field_name()</u> : Retourne le nom d'une colonne Ovrimos.....	1057
<u>7.76.17 ovrimos_field_num()</u> : Retourne le numéro de colonne Ovrimos.....	1058
<u>7.76.18 ovrimos_field_type()</u> : Retourne le type numérique d'une colonne Ovrimos.....	1058
<u>7.76.19 ovrimos_free_result()</u> : Libère les ressources utilisées par un résultat Ovrimos.....	1058
<u>7.76.20 ovrimos_longreadlen()</u> : Indique la taille des données à lire dans une colonne de grande taille.....	1058
<u>7.76.21 ovrimos_num_fields()</u> : Retourne le nombre de colonnes Ovrimos.....	1058
<u>7.76.22 ovrimos_num_rows()</u> : Retourne le nombre de lignes affectées par une modification Ovrimos.....	1058
<u>7.76.23 ovrimos_prepare()</u> : Prépare une requête SQL Ovrimos.....	1059
<u>7.76.24 ovrimos_result_all()</u> : Affiche un résultat Ovrimos sous forme de table HTML.....	1059
<u>7.76.25 ovrimos_result()</u> : Lit le contenu d'une colonne Ovrimos.....	1060
<u>7.76.26 ovrimos_rollback()</u> : Annule une transaction.....	1060
<u>7.77 Bufferisation de sortie</u>	1061
<u>7.77.1 Introduction</u>	1061
<u>7.77.2 Pré-requis</u>	1061
<u>7.77.3 Installation</u>	1061
<u>7.77.4 Configuration à l'exécution</u>	1061
<u>7.77.5 Types de ressources</u>	1062
<u>7.77.6 Constantes prédefinies</u>	1062
<u>7.77.7 Exemples</u>	1062
<u>7.77.8 Voir aussi</u>	1062

Sommaire

7 Index des fonctions

<u>7.77.10 ob_clean()</u> : Efface le buffer de sortie.....	1063
<u>7.77.11 ob_end_clean()</u> : Détruit les données du buffer de sortie, et éteint la bufferisation de sortie.....	1063
<u>7.77.12 ob_end_flush()</u> : Envoie les données du buffer de sortie, et éteint la bufferisation de sortie.....	1063
<u>7.77.13 ob_flush()</u> : Envoie le buffer de sortie.....	1064
<u>7.77.14 ob_get_clean()</u> : Lit le contenu courant du buffer de sortie puis l'efface.....	1064
<u>7.77.15 ob_get_contents()</u> : Retourne le contenu du buffer de sortie.....	1065
<u>7.77.16 ob_get_length()</u> : Retourne la longueur du contenu du buffer de sortie.....	1065
<u>7.77.17 ob_get_level()</u> : Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie.....	1065
<u>7.77.18 ob_get_status()</u> : Lit le statut du buffer de sortie.....	1065
<u>7.77.19 ob_gzhandler()</u> : Fonction de callback pour la compression automatique des buffers.....	1065
<u>7.77.20 ob_implicit_flush()</u> : Active/désactive l'envoi implicite.....	1066
<u>7.77.21 ob_start()</u> : Enclenche la bufferisation de sortie.....	1066
<u>7.78 Overload</u>	1068
<u>7.78.1 Introduction</u>	1068
<u>7.78.2 Pré-requis</u>	1068
<u>7.78.3 Installation</u>	1068
<u>7.78.4 Configuration à l'exécution</u>	1068
<u>7.78.5 Types de ressources</u>	1068
<u>7.78.6 Constantes prédefinies</u>	1068
<u>7.78.7 Exemples</u>	1069
<u>7.79 PDF</u>	1070
<u>7.79.1 Introduction</u>	1070
<u>7.79.2 Pré-requis</u>	1070
<u>7.79.3 Installation</u>	1070
<u>7.79.4 Configuration à l'exécution</u>	1071
<u>7.79.5 Confusion entre les vieilles versions de PDFLib</u>	1071
<u>7.79.6 Exemples</u>	1071
<u>7.79.7 Voir aussi</u>	1073
<u>7.79.9 pdf_add_bookmark()</u> : Ajoute un signet dans la page courante.....	1076
<u>7.79.10 pdf_add_launchlink()</u> : Ajoute une annotation de lancement dans la page PDF courante.....	1076
<u>7.79.11 pdf_add_locallink()</u> : Ajoute une annotation de lien dans la page PDF courante.....	1076
<u>7.79.12 pdf_add_note()</u> : Ajoute une annotation dans la page PDF courante.....	1077
<u>7.79.13 pdf_add_outline()</u> : Obsolète : Ajoute un signet dans la page courante.....	1077
<u>7.79.14 pdf_add_pdflink()</u> : Ajoute une annotation sur un lien vers un fichier dans la page PDF courante.....	1077
<u>7.79.15 pdf_add_thumbnail()</u> : Ajoute une miniature sur la page PDF courante.....	1077
<u>7.79.16 pdf_add_weblink()</u> : Ajoute un lien web sur la page PDF courante.....	1077
<u>7.79.17 pdf_arc()</u> : Dessine un arc de cercle PDF dans le sens anti-horaire.....	1078
<u>7.79.18 pdf_arcn()</u> : Dessine un arc de cercle dans le sens horaire.....	1078
<u>7.79.19 pdf_attach_file()</u> : Ajoute un fichier attaché à la page PDF.....	1078
<u>7.79.20 pdf_begin_page()</u> : Initialise une nouvelle page de document PDF.....	1078
<u>7.79.21 pdf_begin_pattern()</u> : Initialise un nouveau pattern PDF.....	1079
<u>7.79.22 pdf_begin_template()</u> : Initialise un nouveau template PDF.....	1079
<u>7.79.23 pdf_circle()</u> : Dessine un cercle dans un document PDF.....	1079
<u>7.79.24 pdf_clip()</u> : Modifie le chemin de clipping PDF.....	1079

Sommaire

7 Index des fonctions

<u>7.79.25 pdf_close_image()</u> : Ferme une image dans un document PDF.....	1079
<u>7.79.26 pdf_close_pdi_page()</u> : Ferme la page PDF.....	1080
<u>7.79.27 pdf_close_pdi()</u> : Ferme le fichier PDF d'entrée.....	1080
<u>7.79.28 pdf_close()</u> : Ferme le fichier PDF.....	1080
<u>7.79.29 pdf_closepath_fill_stroke()</u> : Termine le chemin, dessine les bords et remplit la forme.....	1080
<u>7.79.30 pdf_closepath_stroke()</u> : Termine le chemin et dessine les bords.....	1080
<u>7.79.31 pdf_closepath()</u> : Termine le chemin PDF courant.....	1080
<u>7.79.32 pdf_concat()</u> : Concatène une matrice avec le CTM.....	1081
<u>7.79.33 pdf_continue_text()</u> : Affiche du texte à la prochaine ligne PDF.....	1081
<u>7.79.34 pdf_curveto()</u> : Dessine une courbe de Bezier.....	1081
<u>7.79.35 pdf_delete()</u> : Efface un objet PDF.....	1081
<u>7.79.36 pdf_end_page()</u> : Termine la page PDF courante.....	1081
<u>7.79.37 pdf_end_pattern()</u> : Termine le pattern PDF.....	1082
<u>7.79.38 pdf_end_template()</u> : Termine le template PDF.....	1082
<u>7.79.39 pdf_endpath()</u> : Obsolète : Termine le chemin courant.....	1082
<u>7.79.40 pdf_fill_stroke()</u> : Remplit et passe le pinceau sur le chemin PDF courant.....	1082
<u>7.79.41 pdf_fill()</u> : Remplit le chemin PDF courant avec la couleur courante.....	1082
<u>7.79.42 pdf_findfont()</u> : Prépare une police pour utilisation ultérieure avec pdf_setfont.....	1082
<u>7.79.43 pdf_get_buffer()</u> : Lit le tampon contenant le fichier PDF généré.....	1083
<u>7.79.44 pdf_get_font()</u> : Obsolète: charge une police.....	1083
<u>7.79.45 pdf_get_fontname()</u> : Obsolète : lit le nom de la police.....	1083
<u>7.79.46 pdf_get_fontsize()</u> : Obsolète : gère les polices.....	1083
<u>7.79.47 pdf_get_image_height()</u> : Retourne la hauteur d'une image.....	1083
<u>7.79.48 pdf_get_image_width()</u> : Retourne la largeur d'une image.....	1084
<u>7.79.49 pdf_get_majorversion()</u> : Retourne le numéro de version majeur de PDFlib.....	1084
<u>7.79.50 pdf_get_minorversion()</u> : Retourne le numéro de version mineure de PDFlib.....	1084
<u>7.79.51 pdf_get_parameter()</u> : Lit certains paramètres.....	1084
<u>7.79.52 pdf_get_pdi_parameter()</u> : Lit des paramètres textuels dans le document PDI.....	1084
<u>7.79.53 pdf_get_pdi_value()</u> : Lit des paramètres numériques dans le document PDF d'entrée.....	1084
<u>7.79.54 pdf_get_value()</u> : Lit des paramètres numériques.....	1085
<u>7.79.55 pdf_initgraphics()</u> : Remet à zéro l'environnement graphique PDF.....	1085
<u>7.79.56 pdf_lineto()</u> : Dessine une ligne PDF.....	1085
<u>7.79.57 pdf_makespotcolor()</u> : Place un point de couleur PDF.....	1085
<u>7.79.58 pdf_moveto()</u> : Place le point courant PDF.....	1085
<u>7.79.59 pdf_new()</u> : Crée un nouvel objet PDF.....	1085
<u>7.79.60 pdf_open_ccitt()</u> : Ouvre une image contenant des données brutes CCITT.....	1086
<u>7.79.61 pdf_open_file()</u> : Ouvre un nouveau fichier PDF.....	1086
<u>7.79.62 pdf_open_gif()</u> : Obsolète : Ouvre une image GIF.....	1086
<u>7.79.63 pdf_open_image_file()</u> : Lit une image depuis un fichier.....	1086
<u>7.79.64 pdf_open_image()</u> : Ouvre une image.....	1087
<u>7.79.65 pdf_open_jpeg()</u> : Obsolète : Ouvre une image JPEG.....	1087
<u>7.79.66 pdf_open_memory_image()</u> : Ouvre une image créée en mémoire par PHP.....	1087
<u>7.79.67 pdf_open_pdi_page()</u> : Prépare une page.....	1087

Sommaire

7 Index des fonctions

<u>7.79.68 pdf_open_pdi() : Ouvre un fichier PDF.....</u>	1087
<u>7.79.69 pdf_open_png() : Obsolète : ouvre une image PNG.....</u>	1088
<u>7.79.70 pdf_open_tiff() : Obsolète : Ouvre une image TIFF.....</u>	1088
<u>7.79.71 pdf_open() : Obsolète : Ouvre un nouvel objet PDF.....</u>	1088
<u>7.79.72 pdf_place_image() : Place une image dans la page.....</u>	1088
<u>7.79.73 pdf_place_pdi_page() : Place une page dans le document.....</u>	1088
<u>7.79.74 pdf_rect() : Dessine un rectangle.....</u>	1088
<u>7.79.75 pdf_restore() : Rétablit l'ancien environnement graphique PDF.....</u>	1088
<u>7.79.76 pdf_rotate() : Configure la rotation.....</u>	1089
<u>7.79.77 pdf_save() : Sauve l'environnement graphique courant.....</u>	1089
<u>7.79.78 pdf_scale() : Configure l'échelle du document.....</u>	1089
<u>7.79.79 pdf_set_border_color() : Configure la couleur des bords autour des liens et annotations.....</u>	1089
<u>7.79.80 pdf_set_border_dash() : Configure le style des lignes autour des liens at annotations.....</u>	1089
<u>7.79.81 pdf_set_border_style() : Choisit le style de bord autour des liens et annotations.....</u>	1089
<u>7.79.82 pdf_set_char_spacing() : Obsolète : Configure l'espacement de caractères character spacing.....</u>	1090
<u>7.79.83 pdf_set_duration() : Obsolète : Configure la durée entre deux pages.....</u>	1090
<u>7.79.84 pdf_set_font() : Obsolète : Sélectionne la taille et le nom d'une police.....</u>	1090
<u>7.79.85 pdf_set_horiz_scaling() : Obsolète : Configure l'échelle horizontale du texte.....</u>	1090
<u>7.79.86 pdf_set_info_author() : Obsolète : remplit le champ d'auteur du document.....</u>	1090
<u>7.79.87 pdf_set_info_creator() : Obsolète : remplit le champ de créateur du document.....</u>	1090
<u>7.79.88 pdf_set_info_keywords() : Obsolète : remplit le champ de mots clés du document.....</u>	1090
<u>7.79.89 pdf_set_info_subject() : Obsolète : remplit le champ de sujet du document.....</u>	1091
<u>7.79.90 pdf_set_info_title() : Obsolète : remplit le champ de titre du document.....</u>	1091
<u>7.79.91 pdf_set_info() : Remplit un champ de l'en-tête de document PDF.....</u>	1091
<u>7.79.92 pdf_set_leading() : Obsolète : Configure la distance entre deux lignes de texte.....</u>	1091
<u>7.79.93 pdf_set_parameter() : Modifie certains paramètres.....</u>	1091
<u>7.79.94 pdf_set_text_matrix() : Obsolète : configure la matrice de texte.....</u>	1091
<u>7.79.95 pdf_set_text_pos() : Modifie la position du texte.....</u>	1091
<u>7.79.96 pdf_set_text_rendering() : Obsolète : Détermine le rendu du texte.....</u>	1092
<u>7.79.97 pdf_set_text_rise() : Obsolète : configure l'élévation de texte.....</u>	1092
<u>7.79.98 pdf_set_value() : Modifie les paramètres numériques.....</u>	1092
<u>7.79.99 pdf_set_word_spacing() : Obsolète : configure l'espace entre deux mots.....</u>	1092
<u>7.79.100 pdf_setcolor() : Configure la couleur de dessin et de remplissage.....</u>	1092
<u>7.79.101 pdf_setdash() : Configure le mode de pointillé.....</u>	1093
<u>7.79.102 pdf_setflat() : Configure la position à plat (flatness).....</u>	1093
<u>7.79.103 pdf_setfont() : Configure la police courante.....</u>	1093
<u>7.79.104 pdf_setgray_fill() : Configure la couleur de remplissage à un niveau de gris.....</u>	1093
<u>7.79.105 pdf_setgray_stroke() : Configure la couleur de dessin à un niveau de gris.....</u>	1094
<u>7.79.106 pdf_setgray() : Configure la couleur de dessin et de remplissage à un</u>	

Sommaire

7 Index des fonctions

<u>niveau de gris</u>	1094
<u>7.79.107 pdf_setlinecap()</u> : Configure le paramètre de linecap.....	1094
<u>7.79.108 pdf_setlinejoin()</u> : Configure le paramètre de linejoin.....	1094
<u>7.79.109 pdf_setlinewidth()</u> : Configure la largeur de ligne.....	1094
<u>7.79.110 pdf_setmatrix()</u> : Configure la matrice de transformation.....	1095
<u>7.79.111 pdf_setmiterlimit()</u> : Configure la "miter limit".....	1095
<u>7.79.112 pdf_setpolydash()</u> : Configure des pointillés complexes.....	1095
<u>7.79.113 pdf_setrgbcolor_fill()</u> : Choisit la couleur utilisée pour le remplissage.....	1095
<u>7.79.114 pdf_setrgbcolor_stroke()</u> : Choisit la couleur utilisée pour le dessin.....	1095
<u>7.79.115 pdf_setrgbcolor()</u> : Choisit la couleur rgb de remplissage et de dessin.....	1095
<u>7.79.116 pdf_show_boxed()</u> : Affiche le texte dans un cadre.....	1096
<u>7.79.117 pdf_show_xy()</u> : Affiche un text à une position donnée.....	1096
<u>7.79.118 pdf_show()</u> : Affiche le texte à la position courante.....	1096
<u>7.79.119 pdf_skew()</u> : Incline le système de coordonnées.....	1096
<u>7.79.120 pdf_stringwidth()</u> : Retourne la largeur d'un texte avec la police courante.....	1096
<u>7.79.121 pdf_stroke()</u> : Dessine la ligne le long du chemin.....	1097
<u>7.79.122 pdf_translate()</u> : Translate l'origine du système de coordonnées.....	1097
<u>7.80 Paiement par Verisign</u>	1098
<u>7.80.1 Introduction</u>	1098
<u>7.80.2 Pré-requis</u>	1098
<u>7.80.3 Installation</u>	1098
<u>7.80.4 Configuration à l'exécution</u>	1098
<u>7.80.5 Types de ressources</u>	1099
<u>7.80.6 Constantes prédefinies</u>	1099
<u>7.80.8 pfpro_init()</u> : Initialise la librairie Payflow Pro.....	1099
<u>7.80.9 pfpro_process_raw()</u> : Envoie une transaction brute à Payflow Pro.....	1099
<u>7.80.10 pfpro_process()</u> : Effectue une transaction avec Payflow Pro.....	1100
<u>7.80.11 pfpro_version()</u> : Lit le numéro de version de Payflow Pro.....	1101
<u>7.81 Options PHP et informations</u>	1102
<u>7.81.1 Introduction</u>	1102
<u>7.81.2 Pré-requis</u>	1102
<u>7.81.3 Installation</u>	1102
<u>7.81.4 Configuration à l'exécution</u>	1102
<u>7.81.5 Types de ressources</u>	1103
<u>7.81.6 Constantes prédefinies</u>	1103
<u>7.81.8 assert()</u> : Vérifie si une assertion est fausse.....	1105
<u>7.81.9 dl()</u> : Charge une extension PHP à la volée.....	1106
<u>7.81.10 extension_loaded()</u> : Détermine si une extension est chargée ou non.....	1107
<u>7.81.11 get_cfg_var()</u> : Retourne la valeur d'une option de PHP.....	1107
<u>7.81.12 get_current_user()</u> : Retourne le nom du possesseur du script courant.....	1108
<u>7.81.13 get_defined_constants()</u> : Retourne la liste des constantes et leur valeur.....	1108
<u>7.81.14 get_extension_funcs()</u> : Liste les fonctions d'une extension.....	1108
<u>7.81.15 get_include_path()</u> : Lit la valeur de la directive de configuration include_path.....	1109
<u>7.81.16 get_included_files()</u> : Retourne un tableau avec les noms des fichiers qui sont inclus dans un script.....	1109
<u>7.81.17 get_loaded_extensions()</u> : Retourne la liste de tous les modules compilés et chargés.....	1110
<u>7.81.18 get_magic_quotes_gpc()</u> : Retourne la configuration actuelle de l'option magic_quotes_gpc.....	1110

Sommaire

7 Index des fonctions

<u>7.81.19 get_magic_quotes_runtime() : Retourne la configuration actuelle de l'option magic_quotes_runtime.....</u>	1111
<u>7.81.20 get_required_files() : Alias de get_included_files.....</u>	1111
<u>7.81.21 getenv() : Retourne la valeur d'une variable d'environnement.....</u>	1111
<u>7.81.22 getlastmod() : Retourne la date de dernière modification de la page.....</u>	1111
<u>7.81.23 getmygid() : Retourne le GID du propriétaire du script.....</u>	1112
<u>7.81.24 getmyinode() : Retourne l'inode du script.....</u>	1112
<u>7.81.25 getmypid() : Retourne le numéro de processus courant.....</u>	1112
<u>7.81.26 getmyuid() : Retourne l'UID du propriétaire du script actuel.....</u>	1112
<u>7.81.27 getopt() : Lit des options passés dans la ligne de commande.....</u>	1113
<u>7.81.28 getrusage() : Retourne le niveau d'utilisation des ressources.....</u>	1113
<u>7.81.29 ini_alter() : Alias de ini_set.....</u>	1113
<u>7.81.30 ini_get_all() : Lit toutes les valeurs de configuration.....</u>	1114
<u>7.81.31 ini_get() : Lit la valeur d'une option de configuration.....</u>	1114
<u>7.81.32 ini_restore() : Restaure la valeur de l'option de configuration.....</u>	1115
<u>7.81.33 ini_set() : Modifie la valeur d'une option de configuration.....</u>	1115
<u>7.81.34 main() : Fausse documentation pour main.....</u>	1119
<u>7.81.35 memory_get_usage() : Indique la quantité de mémoire utilisée par PHP.....</u>	1119
<u>7.81.36 php_ini_scanned_files() : Retourne la liste des fichiers .ini analysés dans les dossiers de configuration supplémentaires.....</u>	1120
<u>7.81.37 php_logo_guid() : Retourne l'identifiant du logo PHP.....</u>	1120
<u>7.81.38 php_sapi_name() : Retourne le type d'interface utilisé entre le serveur web et PHP.....</u>	1121
<u>7.81.39 php_undef() : Retourne les informations sur le système d'exploitation.....</u>	1121
<u>7.81.40 phpcredits() : Imprime les crédits de PHP.....</u>	1122
<u>7.81.41 phpinfo() : Affiche de nombreuses informations sur le PHP.....</u>	1123
<u>7.81.42 phpversion() : Retourne le numéro de la version courante de PHP.....</u>	1123
<u>7.81.43 putenv() : Fixe la valeur d'une variable d'environnement.....</u>	1124
<u>7.81.44 restore_include_path() : Restaure la valeur de la directive de configuration include_path.....</u>	1124
<u>7.81.45 set_include_path() : Modifie la valeur de la directive de configuration include_path.....</u>	1125
<u>7.81.46 set_magic_quotes_runtime() : Active/désactive l'option magic_quotes_runtime.....</u>	1125
<u>7.81.47 set_time_limit() : Fixe le temps maximum d'exécution d'un script.....</u>	1125
<u>7.81.48 version_compare() : Compare deux versions de PHP.....</u>	1126
<u>7.81.49 zend_logo_guid() : Retourne le logo de Zend.....</u>	1126
<u>7.81.50 zend_version() : Lit la version courante du moteur Zend.....</u>	1126
<u>7.82 POSIX.....</u>	1127
<u>7.82.1 Introduction.....</u>	1127
<u>7.82.2 Installation.....</u>	1127
<u>7.82.3 Voir aussi.....</u>	1127
<u>7.82.5 posix_get_last_error() : Lit le dernier numéro d'erreur généré par la dernière fonction POSIX qui a échoué.....</u>	1128
<u>7.82.6 posix_getcwd() : Chemin du dossier de travail courant.....</u>	1128
<u>7.82.7 posix_getegid() : Retourne l'ID effectif du groupe du processus courant.....</u>	1128
<u>7.82.8 posix_geteuid() : Retourne l'UID effectif de l'utilisateur du processus courant.....</u>	1128
<u>7.82.9 posix_getgid() : Retourne l'UID du groupe du processus courant.....</u>	1129
<u>7.82.10 posix_getgrgid() : Retourne des informations sur un groupe.....</u>	1129
<u>7.82.11 posix_getgrnam() : Retourne des informations sur un groupe.....</u>	1129

Sommaire

7 Index des fonctions

7.82.12 <u>posix_getgroups()</u> : Retourne les identifiants du groupe du processus courant.....	1129
7.82.13 <u>posix_getlogin()</u> : Retourne le nom de login.....	1130
7.82.14 <u>posix_getpgid()</u> : Retourne l'id du groupe de processus.....	1130
7.82.15 <u>posix_getpgrp()</u> : Retourne l'identifiant du groupe de processus.....	1130
7.82.16 <u>posix_getpid()</u> : Retourne l'identifiant du processus courant.....	1130
7.82.17 <u>posix_getppid()</u> : Retourne l'identifiant du processus parent.....	1130
7.82.18 <u>posix_getpwnam()</u> : Retourne des informations sur un utilisateur.....	1130
7.82.19 <u>posix_getpwuid()</u> : Retourne des informations sur un utilisateur.....	1131
7.82.20 <u>posix_getrlimit()</u> : Retourne les limites système.....	1131
7.82.21 <u>posix_getsid()</u> : Retourne le sid du processus.....	1131
7.82.22 <u>posix_getuid()</u> : Retourne l'ID de l'utilisateur du processus courant.....	1131
7.82.23 <u>posix_isatty()</u> : Détermine si un pointeur de fichier est un terminal interactif.....	1132
7.82.24 <u>posix_kill()</u> : Envoie un signal à un processus.....	1132
7.82.25 <u>posix_mkfifo()</u> : Crée un fichier FIFO (first in, first out) (un pipe nommé).....	1132
7.82.26 <u>posix_setegid()</u> : Modifie le GID réel du processus courant.....	1132
7.82.27 <u>posix seteuid()</u> : Set the effective UID of the current process.....	1133
7.82.28 <u>posix_setgid()</u> : Fixe le GID effective du processus courant.....	1133
7.82.29 <u>posix_setpgid()</u> : Fixe l'identifiant de group de processus.....	1133
7.82.30 <u>posix_setsid()</u> : Fait du processus courant un chef de session.....	1133
7.82.31 <u>posix_setuid()</u> : Fixe l'UID effective du processus courant.....	1133
7.82.32 <u>posix_strerror()</u> : Lit le message d'erreur associé à un numéro d'erreur POSIX.....	1133
7.82.33 <u>posix_times()</u> : Utilisation des ressources.....	1134
7.82.34 <u>posix_ttyname()</u> : Retourne le nom de device du terminal.....	1134
7.82.35 <u>posix_uname()</u> : Retourne le nom du système.....	1134
7.83 <u>PostgreSQL</u>	1135
7.83.1 <u>Introduction</u>	1135
7.83.2 <u>Pré-requis</u>	1135
7.83.3 <u>Installation</u>	1135
7.83.4 <u>Configuration à l'exécution</u>	1135
7.83.5 <u>Trucs et astuces d'utilisation</u>	1136
7.83.6 <u>Constantes prédefinies</u>	1137
7.83.7 <u>Exemples</u>	1138
7.83.9 <u>pg_cancel_query()</u> : Annule une requête asynchrone.....	1139
7.83.10 <u>pg_client_encoding()</u> : Lit l'encodage du client.....	1140
7.83.11 <u>pg_close()</u> : Termine une connexion PostgreSQL.....	1140
7.83.12 <u>pg_connect()</u> : Ouvre une connexion PostgreSQL.....	1140
7.83.13 <u>pg_connection_busy()</u> : Vérifie si la connexion PostgreSQL est occupée.....	1141
7.83.14 <u>pg_connection_reset()</u> : Relance la connexion au serveur PostgreSQL.....	1141
7.83.15 <u>pg_connection_status()</u> : Lit le statut de la connexion PostgreSQL.....	1142
7.83.16 <u>pg_convert()</u> : Convertit un tableau associatif en une commande PostgreSQL.....	1142
7.83.17 <u>pg_copy_from()</u> : Insère des lignes dans une table à partir d'un tableau.....	1142
7.83.18 <u>pg_copy_to()</u> : Copie une table dans un tableau.....	1143
7.83.19 <u>pg_dbname()</u> : Nom de la base de données.....	1143
7.83.20 <u>pg_delete()</u> : Efface des lignes PostgreSQL.....	1143
7.83.21 <u>pg_end_copy()</u> : Synchronise avec le serveur PostgreSQL.....	1144
7.83.22 <u>pg_escape_bytea()</u> : Protège le caractères d'une chaîne binaire en mode bytea.....	1144

Sommaire

7 Index des fonctions

<u>7.83.23 pg_escape_string()</u> : Protège une chaîne de caractères.....	1144
<u>7.83.24 pg_fetch_all()</u> : Lit toutes les lignes d'un résultat.....	1144
<u>7.83.25 pg_fetch_array()</u> : Lit une ligne de résultat PGSQL dans un tableau.....	1145
<u>7.83.26 pg_fetch_assoc()</u> : Lit une ligne de résultat PGSQL sous forme de tableau numérique.....	1146
<u>7.83.27 pg_fetch_object()</u> : Lit une ligne de résultat PGSQL dans un objet.....	1146
<u>7.83.28 pg_fetch_result()</u> : Retourne les valeurs d'un résultat.....	1147
<u>7.83.29 pg_fetch_row()</u> : Lit une ligne dans un tableau.....	1148
<u>7.83.30 pg_field_is_null()</u> : Teste si un champ PostGreSQL est à NULL.....	1148
<u>7.83.31 pg_field_name()</u> : Retourne le nom d'un champ PostGreSQL.....	1149
<u>7.83.32 pg_field_num()</u> : Retourne le numéro d'une colonne.....	1149
<u>7.83.33 pg_field_prflen()</u> : Retourne la taille imprimée.....	1150
<u>7.83.34 pg_field_size()</u> : Retourne la taille interne de stockage d'un champ donné.....	1150
<u>7.83.35 pg_field_type()</u> : Retourne le type d'un champ PostGreSQL donné par index.....	1150
<u>7.83.36 pg_free_result()</u> : Libère la mémoire.....	1150
<u>7.83.37 pg_get_notify()</u> : Lit le message de NOTIFY.....	1151
<u>7.83.38 pg_get_pid()</u> : Lit l'identifiant de processus du serveur.....	1151
<u>7.83.39 pg_get_result()</u> : Lit un résultat asynchrone.....	1152
<u>7.83.40 pg_host()</u> : Retourne le nom d'hôte.....	1152
<u>7.83.41 pg_insert()</u> : Insère un tableau dans une table.....	1152
<u>7.83.42 pg_last_error()</u> : Lit le dernier message d'erreur sur la connexion.....	1152
<u>7.83.43 pg_last_notice()</u> : Retourne la dernière note du serveur PostgreSQL.....	1153
<u>7.83.44 pg_last_oid()</u> : Retourne l'identifiant du dernier objet.....	1153
<u>7.83.45 pg_lo_close()</u> : Ferme un objet de grande taille PostgreSQL.....	1154
<u>7.83.46 pg_lo_create()</u> : Crée un objet de grande taille PostgreSQL.....	1154
<u>7.83.47 pg_lo_export()</u> : Exporte un objet de grande taille vers un fichier.....	1154
<u>7.83.48 pg_lo_import()</u> : Importe un objet de grande taille depuis un fichier.....	1154
<u>7.83.49 pg_lo_open()</u> : Ouvre un objet de grande taille PostgreSQL.....	1155
<u>7.83.50 pg_lo_read_all()</u> : Lit un objet de grande taille en totalité.....	1155
<u>7.83.51 pg_lo_read()</u> : Lit un objet de grande taille.....	1156
<u>7.83.52 pg_lo_seek()</u> : Modifie la position dans un objet de grande taille.....	1156
<u>7.83.53 pg_lo_tell()</u> : Retourne la position courante dans un objet de grande taille.....	1156
<u>7.83.54 pg_lo_unlink()</u> : Efface un objet de grande taille.....	1156
<u>7.83.55 pg_lo_write()</u> : Écrit un objet de grande taille PGSQL.....	1157
<u>7.83.56 pg_meta_data()</u> : Lit les données méta de la table PostGreSQL.....	1157
<u>7.83.57 pg_num_fields()</u> : Retourne le nombre de champ.....	1158
<u>7.83.58 pg_num_rows()</u> : Retourne le nombre de lignes PostgreSQL.....	1158
<u>7.83.59 pg_options()</u> : Retourne les options PostgreSQL.....	1158
<u>7.83.60 pg_pconnect()</u> : Etablit une connexion PostgreSQL persistante.....	1159
<u>7.83.61 pg_ping()</u> : Pingue la connexion à la base.....	1159
<u>7.83.62 pg_port()</u> : Retourne le numéro de port.....	1159
<u>7.83.63 pg_put_line()</u> : Envoie une chaîne au serveur PostgreSQL.....	1159
<u>7.83.64 pg_query()</u> : Exécute une requête.....	1160
<u>7.83.65 pg_result_error()</u> : Lit le message d'erreur associé à un résultat.....	1160
<u>7.83.66 pg_result_seek()</u> : Modifie la ligne courant dans un résultat.....	1161
<u>7.83.67 pg_result_status()</u> : Lit le statut du résultat.....	1161
<u>7.83.68 pg_select()</u> : Effectue une selection.....	1161
<u>7.83.69 pg_send_query()</u> : Exécute une requête PostGreSQL asynchrone.....	1161
<u>7.83.70 pg_set_client_encoding()</u> : Choisit l'encodage du client.....	1162

Sommaire

7 Index des fonctions

<u>7.83.71 pg_trace()</u> : Active le suivi d'une connexion PostgreSQL.....	1162
<u>7.83.72 pg_tty()</u> : Retourne le nom de tty.....	1163
<u>7.83.73 pg_unescape_bytea()</u> : Supprime la protection d'une chaîne de type bytea.....	1163
<u>7.83.74 pg_untrace()</u> : Termine le suivi d'une connexion PostgreSQL.....	1163
<u>7.83.75 pg_update()</u> : Modifie les lignes d'une table.....	1164
<u>7.84 Contrôle des processus</u>	1165
<u>7.84.1 Introduction</u>	1165
<u>7.84.2 Pré-requis</u>	1165
<u>7.84.3 Installation</u>	1165
<u>7.84.4 Configuration à l'exécution</u>	1165
<u>7.84.5 Types de ressources</u>	1165
<u>7.84.6 Constantes prédefinies</u>	1165
<u>7.84.7 Exemples</u>	1166
<u>7.84.8 Voir aussi</u>	1167
<u>7.84.10 pcntl_fork()</u> : Forks the currently running process.....	1167
<u>7.84.11 pcntl_signal()</u> : Installe un gestionnaire de signaux.....	1168
<u>7.84.12 pcntl_waitpid()</u> : Attend la fin de l'exécution d'un processus fils.....	1169
<u>7.84.13 pcntl_wexitstatus()</u> : Retourne le code d'un processus fils terminé.....	1169
<u>7.84.14 pcntl_wifexited()</u> : Retourne TRUE si le code de retour représente une fin normale.....	1170
<u>7.84.15 pcntl_wifsignaled()</u> : Returns TRUE if status code represents a termination due to a signal.....	1170
<u>7.84.16 pcntl_wifstopped()</u> : Returns TRUE if child process is currently stopped.....	1170
<u>7.84.17 pcntl_wstopsig()</u> : Returns the signal which caused the child to stop.....	1170
<u>7.84.18 pcntl_wtermsig()</u> : Returns the signal which caused the child to terminate.....	1171
<u>7.85 Exécution de programmes externes</u>	1172
<u>7.85.1 Introduction</u>	1172
<u>7.85.2 Pré-requis</u>	1172
<u>7.85.3 Installation</u>	1172
<u>7.85.4 Configuration à l'exécution</u>	1172
<u>7.85.5 Types de ressources</u>	1172
<u>7.85.6 Constantes prédefinies</u>	1172
<u>7.85.7 Voir aussi</u>	1172
<u>7.85.9 escapeshellcmd()</u> : Protège les caractères spéciaux du Shell.....	1173
<u>7.85.10 exec()</u> : Exécute un programme externe.....	1173
<u>7.85.11 passthru()</u> : Exécute un programme externe et affiche le résultat brut.....	1174
<u>7.85.12 proc_close()</u> : Ferme un processus ouvert par proc_open() et retourne le code de sortie du processus.....	1174
<u>7.85.13 proc_get_status()</u> : Lit les informations concernant un processus.....	1174
<u>7.85.14 proc_nice()</u> : Change la priorité d'exécution du processus courant.....	1174
<u>7.85.15 proc_open()</u> : Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties.....	1175
<u>7.85.16 proc_terminate()</u> : Termine un processus ouvert par proc_open().....	1176
<u>7.85.17 shell_exec()</u> : Exécute une commande via le shell et retourne le résultat sous forme de chaîne.....	1176
<u>7.85.18 system()</u> : Exécute un programme externe et affiche le résultat.....	1176
<u>7.86 Impression</u>	1178
<u>7.86.1 Introduction</u>	1178
<u>7.86.2 Installation</u>	1178
<u>7.86.3 Configuration à l'exécution</u>	1178

Sommaire

7 Index des fonctions

<u>7.86.5 printer_close()</u> : Ferme une connexion à une imprimante.....	1179
<u>7.86.6 printer_create_brush()</u> : Crée une nouvelle brosse.....	1179
<u>7.86.7 printer_create_dc()</u> : Crée un contexte d'imprimante.....	1179
<u>7.86.8 printer_create_font()</u> : Crée une nouvelle police.....	1180
<u>7.86.9 printer_create_pen()</u> : Crée un nouveau stylo.....	1180
<u>7.86.10 printer_delete_brush()</u> : Efface une brosse.....	1181
<u>7.86.11 printer_delete_dc()</u> : Efface un contexte d'imprimante.....	1181
<u>7.86.12 printer_delete_font()</u> : Efface une police.....	1181
<u>7.86.13 printer_delete_pen()</u> : Efface un stylo.....	1181
<u>7.86.14 printer_draw_bmp()</u> : Dessine un fichier BMP.....	1181
<u>7.86.15 printer_draw_chord()</u> : Dessine un arc de cercle.....	1182
<u>7.86.16 printer_draw_ellipse()</u> : Dessine une ellipse.....	1182
<u>7.86.17 printer_draw_line()</u> : Dessine une ligne.....	1183
<u>7.86.18 printer_draw_pie()</u> : Dessine un secteur angulaire.....	1183
<u>7.86.19 printer_draw_rectangle()</u> : Dessine un rectangle.....	1184
<u>7.86.20 printer_draw_roundrect()</u> : Dessine un rectangle, avec des coins arrondis.....	1185
<u>7.86.21 printer_draw_text()</u> : Dessine un texte.....	1185
<u>7.86.22 printer_end_doc()</u> : Ferme le document.....	1186
<u>7.86.23 printer_end_page()</u> : Ferme la page active.....	1186
<u>7.86.24 printer_get_option()</u> : Lit les informations de configuration de l'imprimante.....	1186
<u>7.86.25 printer_list()</u> : Retourne la liste des imprimantes attachées à un serveur.....	1186
<u>7.86.26 printer_logical_fontheight()</u> : Lit la taille logique de la police.....	1187
<u>7.86.27 printer_open()</u> : Ouvre une connexion à une imprimante.....	1187
<u>7.86.28 printer_select_brush()</u> : Sélectionne une brosse.....	1188
<u>7.86.29 printer_select_font()</u> : Sélectionne une police de caractères.....	1188
<u>7.86.30 printer_select_pen()</u> : Sélectionne un stylo.....	1188
<u>7.86.31 printer_set_option()</u> : Configure la connexion à l'imprimante.....	1189
<u>7.86.32 printer_start_doc()</u> : Commence un nouveau document.....	1190
<u>7.86.33 printer_start_page()</u> : Commence une nouvelle page.....	1190
<u>7.86.34 printer_write()</u> : Ecrit des données sur l'imprimante.....	1190
<u>7.87 Pspell</u>	1192
<u>7.87.1 Introduction</u>	1192
<u>7.87.2 Pré-requis</u>	1192
<u>7.87.3 Installation</u>	1192
<u>7.87.4 Configuration à l'exécution</u>	1192
<u>7.87.5 Types de ressources</u>	1192
<u>7.87.6 Constantes prédefinies</u>	1192
<u>7.87.8 pspell_add_to_session()</u> : Ajoute le mot au dictionnaire personnel de la session courante.....	1193
<u>7.87.9 pspell_check()</u> : Vérifie un mot.....	1193
<u>7.87.10 pspell_clear_session()</u> : Remet à zéro la session courante.....	1193
<u>7.87.11 pspell_config_create()</u> : Crée une configuration utilisée pour ouvrir un dictionnaire.....	1194
<u>7.87.12 pspell_config_ignore()</u> : Ignore les mots des moins de N caractères.....	1194
<u>7.87.13 pspell_config_mode()</u> : Change le mode de suggestion.....	1195
<u>7.87.14 pspell_config_personal()</u> : Choisit le fichier qui contient le dictionnaire personnel.....	1195
<u>7.87.15 pspell_config_repl()</u> : Choisit le fichier qui contient les paires de remplacement.....	1195
<u>7.87.16 pspell_config_runtogether()</u> : Considère deux mots accolés comme un composé.....	1196

Sommaire

7 Index des fonctions

7.87.17 pspell_config_save_repl() : Active la sauvegarde des paires de remplacement.....	1196
7.87.18 pspell_new_config() : Charge un nouveau dictionnaire.....	1196
7.87.19 pspell_new_personal() : Charge un nouveau dictionnaire avec un dictionnaire personnel.....	1197
7.87.20 pspell_new() : Charge un nouveau dictionnaire.....	1198
7.87.21 pspell_save_wordlist() : Sauve le dictionnaire personnel dans un fichier.....	1198
7.87.22 pspell_store_replacement() : Enregistre une paire de remplacement pour un mot.....	1199
7.87.23 pspell_suggest() : Suggère une orthographe.....	1199
7.88 Readline (GNU).....	1200
7.88.1 Introduction.....	1200
7.88.2 Pré-requis.....	1200
7.88.3 Installation.....	1200
7.88.4 Configuration à l'exécution.....	1200
7.88.5 Types de ressources.....	1200
7.88.6 Constantes prédefinies.....	1200
7.88.8 readline_clear_history() : Efface l'historique.....	1201
7.88.9 readline_completion_function() : Enregistre une fonction de complétion.....	1201
7.88.10 readline_info() : Lit/modifie diverses variables internes.....	1201
7.88.11 readline_list_history() : Liste l'historique.....	1201
7.88.12 readline_read_history() : Lit l'historique.....	1201
7.88.13 readline_write_history() : Ecrit dans l'historique.....	1202
7.88.14 readline() : Lit une ligne.....	1202
7.89 Recode (GNU).....	1203
7.89.1 Introduction.....	1203
7.89.2 Pré-requis.....	1203
7.89.3 Installation.....	1203
7.89.4 Configuration à l'exécution.....	1203
7.89.5 Types de ressources.....	1203
7.89.6 Constantes prédefinies.....	1203
7.89.8 recode_string() : Recode une chaîne en fonction de la requête.....	1204
7.89.9 recode() : Recode une fonction grâce à une requête.....	1204
7.90 Expressions régulières compatibles Perl.....	1205
7.90.1 Introduction.....	1205
7.90.2 Pré-requis.....	1205
7.90.3 Installation.....	1205
7.90.4 Configuration à l'exécution.....	1205
7.90.5 Types de ressources.....	1205
7.90.6 Constantes prédefinies.....	1205
7.90.7 Exemples.....	1206
7.90.9 syntaxe des masques() : Fonctionnement des expressions régulières.....	1206
7.90.10 preg_grep() : Retourne un tableau avec les résultat de la recherche.....	1206
7.90.11 preg_match_all() : Expression régulière globale.....	1207
7.90.12 preg_match() : Expression régulière standard.....	1208
7.90.13 preg_quote() : Echappement des caractères spéciaux des expressions régulières.....	1209
7.90.14 preg_replace_callback() : Rechercher/remplacer avec une expression régulière et fonction de callback.....	1210
7.90.15 preg_replace() : Rechercher et remplacer par expression régulière standard.....	1211

Sommaire

7 Index des fonctions

<u>7.90.16 preg_split() : Eclate une chaîne par expression régulière</u>	1213
<u>7.91 qtdom</u>	1215
<u>7.91.1 Pré-requis</u>	1215
<u>7.91.2 Installation</u>	1215
<u>7.91.4 qdom_tree() : Crée une arbre à partir d'un code XML</u>	1215
<u>7.92 Expressions régulières</u>	1216
<u>7.92.1 Introduction</u>	1216
<u>7.92.2 Pré-requis</u>	1216
<u>7.92.3 Installation</u>	1216
<u>7.92.4 Configuration à l'exécution</u>	1216
<u>7.92.5 Types de ressources</u>	1217
<u>7.92.6 Constantes prédefinies</u>	1217
<u>7.92.7 Exemples</u>	1217
<u>7.92.8 Voir aussi</u>	1217
<u>7.92.10 ereg() : Expression régulière standard</u>	1218
<u>7.92.11 eregi_replace() : Remplacement par expression régulière insensible à la casse</u>	1218
<u>7.92.12 eregi() : Recherche par expression régulière insensible à la casse</u>	1219
<u>7.92.13 split() : Scinde une chaîne en un tableau, grâce à une expression régulière</u>	1219
<u>7.92.14 spliti() : Scinde une chaîne en un tableau, grâce à une expression régulière</u>	1220
<u>7.92.15 sql_regcase() : Prépare une expression régulière pour effectuer une recherche insensible à la casse</u>	1220
<u>7.93 Sémaphores et gestion de la mémoire partagée</u>	1221
<u>7.93.1 Introduction</u>	1221
<u>7.93.2 Pré-requis</u>	1221
<u>7.93.3 Installation</u>	1221
<u>7.93.4 Configuration à l'exécution</u>	1221
<u>7.93.5 Types de ressources</u>	1221
<u>7.93.6 Constantes prédefinies</u>	1222
<u>7.93.8 msg_get_queue() : Crée ou s'attache à une file de message</u>	1222
<u>7.93.9 msg_receive() : Reçoit un message depuis une file de messages</u>	1222
<u>7.93.10 msg_remove_queue() : Détruit une file de messages</u>	1223
<u>7.93.11 msg_send() : Envoie un message dans une file</u>	1223
<u>7.93.12 msg_set_queue() : Modifie des informations dans la file de messages</u>	1224
<u>7.93.13 msg_stat_queue() : Retourne des informations sur la file de messages</u>	1224
<u>7.93.14 sem_acquire() : Réserve un sémaphore</u>	1225
<u>7.93.15 sem_get() : Retourne un identifiant de sémaphore</u>	1225
<u>7.93.16 sem_release() : Libère un sémaphore</u>	1225
<u>7.93.17 sem_remove() : Détruit un sémaphore</u>	1226
<u>7.93.18 shm_attach() : Crée ou ouvre un segment de mémoire partagée</u>	1226
<u>7.93.19 shm_detach() : Libère un segment de mémoire partagée</u>	1226
<u>7.93.20 shm_get_var() : Lit une variable dans la mémoire partagée</u>	1227
<u>7.93.21 shm_put_var() : Insère ou modifie une variable de la mémoire partagée</u>	1227
<u>7.93.22 shm_remove_var() : Efface une variable de la mémoire partagée</u>	1227
<u>7.93.23 shm_remove() : Supprime un segment de mémoire partagée sous Unix</u>	1227
<u>7.94 SESAM</u>	1228
<u>7.94.1 Introduction</u>	1228
<u>7.94.2 Configuration à l'exécution</u>	1228
<u>7.94.3 Notes sur la configuration</u>	1229

Sommaire

7 Index des fonctions

<u>7.94.4 Considérations à l'exécution</u>	1230
<u>7.94.5 Types de curseurs</u>	1230
<u>7.94.6 Portage</u>	1230
<u>7.94.7 Sécurité</u>	1231
<u>7.94.8 Migration d'une autre base SQL</u>	1231
<u>7.94.9 Notes sur l'utilisation de types SQL divers</u>	1232
<u>7.94.10 Support des "champs multiples" de SESAM</u>	1232
<u>7.94.11 Voir aussi</u>	1232
<u>7.94.13 <code>sesam_commit()</code> : Valide la transaction SESAM en cours</u>	1233
<u>7.94.14 <code>sesam_connect()</code> : Ouvre une connexion SESAM</u>	1233
<u>7.94.15 <code>sesam_diagnostic()</code> : Retourne l'état de la dernière requête SESAM</u>	1234
<u>7.94.16 <code>sesam_disconnect()</code> : Déconnexion d'une base SESAM</u>	1235
<u>7.94.17 <code>sesam_errormsg()</code> : Retourne le message d'erreur SESAM</u>	1236
<u>7.94.18 <code>sesam_execimm()</code> : Exécute immédiatement une requête SQL</u>	1236
<u>7.94.19 <code>sesam_fetch_array()</code> : Lit une ligne dans un tableau associatif</u>	1236
<u>7.94.20 <code>sesam_fetch_result()</code> : Retourne tout ou partie d'un résultat SESAM</u>	1238
<u>7.94.21 <code>sesam_fetch_row()</code> : Lit une ligne dans un tableau</u>	1238
<u>7.94.22 <code>sesam_field_array()</code> : Retourne des informations sur une colonne</u>	1239
<u>7.94.23 <code>sesam_field_name()</code> : Retourne le nom d'une colonne</u>	1240
<u>7.94.24 <code>sesam_free_result()</code> : Libère les ressources SESAM</u>	1240
<u>7.94.25 <code>sesam_num_fields()</code> : Retourne le nombre de colonne dans un résultat</u>	1240
<u>7.94.26 <code>sesam_query()</code> : Exécute une requête SESAM</u>	1241
<u>7.94.27 <code>sesam_rollback()</code> : Annule une transaction SESAM</u>	1242
<u>7.94.28 <code>sesam_seek_row()</code> : Déplace un curseur à défilement</u>	1242
<u>7.94.29 <code>sesam_settransaction()</code> : Modifie les paramètres de transaction SESAM</u>	1242
<u>7.95 Sessions</u>	1244
<u>7.95.1 Introduction</u>	1244
<u>7.95.2 Sessions et sécurité</u>	1244
<u>7.95.3 Pré-requis</u>	1245
<u>7.95.4 Installation</u>	1245
<u>7.95.5 Configuration à l'exécution</u>	1245
<u>7.95.6 Types de ressources</u>	1247
<u>7.95.7 Constantes prédefinies</u>	1248
<u>7.95.8 Exemples</u>	1248
<u>7.95.9 Passer l'identifiant de session (session ID)</u>	1249
<u>7.95.10 Gestion personnalisée des sessions</u>	1250
<u>7.95.12 <code>session_cache_limiter()</code> : Lit et/ou modifie le limiteur de cache de session</u>	1251
<u>7.95.13 <code>session_decode()</code> : Décode les données de session</u>	1251
<u>7.95.14 <code>session_destroy()</code> : Détruit une session</u>	1252
<u>7.95.15 <code>session_encode()</code> : Encode les données de session</u>	1252
<u>7.95.16 <code>session_get_cookie_params()</code> : Lit la configuration du cookie de session</u>	1252
<u>7.95.17 <code>session_id()</code> : Let et/ou modifie l'identifiant courant de session</u>	1253
<u>7.95.18 <code>session_is_registered()</code> : Vérifie si une variable est enregistrée dans la session</u>	1253
<u>7.95.19 <code>session_module_name()</code> : Lit et/ou modifie le module de session courant</u>	1253
<u>7.95.20 <code>session_name()</code> : Lit et/ou modifie le nom de la session</u>	1254
<u>7.95.21 <code>session_regenerate_id()</code> : Remplace l'identifiant de session courant par un nouveau</u>	1254
<u>7.95.22 <code>session_register()</code> : Enregistre une variable dans une session</u>	1255
<u>7.95.23 <code>session_save_path()</code> : Lit et/ou modifie le chemin de sauvegarde des sessions</u>	1256

Sommaire

7 Index des fonctions

7.95.24 <u>session_set_cookie_params()</u> : Modifie les paramètres du cookie de session.....	1256
7.95.25 <u>session_set_save_handler()</u> : Configure les fonctions de stockage de sessions.....	1256
7.95.26 <u>session_start()</u> : Initialise une session.....	1258
7.95.27 <u>session_unregister()</u> : Supprime une variable de la session.....	1259
7.95.28 <u>session_unset()</u> : Détruit toutes les variables d'une session.....	1259
7.95.29 <u>session_write_close()</u> : Ecrit les données de session et ferme la session....	1259
7.96 <u>Mémoire partagée</u>	1260
7.96.1 <u>Introduction</u>	1260
7.96.2 <u>Pré-requis</u>	1260
7.96.3 <u>Installation</u>	1260
7.96.4 <u>Configuration à l'exécution</u>	1260
7.96.5 <u>Types de ressources</u>	1260
7.96.6 <u>Constantes prédéfinies</u>	1260
7.96.7 <u>Exemples</u>	1260
7.96.9 <u>shmop_delete()</u> : Détruit un bloc de mémoire partagée.....	1261
7.96.10 <u>shmop_open()</u> : Crée ou ouvre un bloc de mémoire partagée.....	1261
7.96.11 <u>shmop_read()</u> : Lit un bloc.....	1262
7.96.12 <u>shmop_size()</u> : Lire la taille du bloc de mémoire partagée.....	1262
7.96.13 <u>shmop_write()</u> : Ecrire dans un bloc de mémoire partagée.....	1263
7.97 <u>SQLite</u>	1264
7.97.1 <u>Introduction</u>	1264
7.97.2 <u>Installation</u>	1264
7.97.3 <u>Contacts</u>	1264
7.97.4 <u>Pré-requis</u>	1264
7.97.5 <u>Types de ressources</u>	1264
7.97.6 <u>Constantes prédéfinies</u>	1264
7.97.7 <u>Configuration à l'exécution</u>	1265
7.97.9 <u>sqlite_busy_timeout()</u> : Configure le délai d'attente d'une base SQLite occupée.....	1266
7.97.10 <u>sqlite_changes()</u> : Retourne le nombre de lignes qui ont été modifiées par la dernière requête SQLite.....	1266
7.97.11 <u>sqlite_close()</u> : Ferme la connexion à SQLite.....	1266
7.97.12 <u>sqlite_column()</u> : Lit la valeur d'une colonne dans un résultat SQLite.....	1267
7.97.13 <u>sqlite_create_aggregate()</u> : Enregistre une UDF agregeante pour les requêtes SQLite.....	1267
7.97.14 <u>sqlite_create_function()</u> : Enregistre une fonction utilisateur "classique" UDF pour SQLite.....	1268
7.97.15 <u>sqlite_current()</u> : Lit une ligne de résultat SQLite dans un tableau.....	1269
7.97.16 <u>sqlite_error_string()</u> : Retourne le message d'erreur SQLite.....	1269
7.97.17 <u>sqlite_escape_string()</u> : Protège une chaîne de caractères pour utilisation avec SQLite.....	1269
7.97.18 <u>sqlite_fetch_array()</u> : Lit la prochaine ligne de résultat SQLite dans un tableau.....	1270
7.97.19 <u>sqlite_fetch_single()</u> : Lit la première ligne d'un résultat SQLite sous forme de chaîne.....	1270
7.97.20 <u>sqlite_fetch_string()</u> : Alias de <u>sqlite_fetch_single</u>	1271
7.97.21 <u>sqlite_field_name()</u> : Retourne le nom du champ SQLite.....	1271
7.97.22 <u>sqlite_has_more()</u> : Indique si il reste des lignes SQLite à lire.....	1271
7.97.23 <u>sqlite_last_error()</u> : Retourne le dernier code d'erreur SQLite.....	1271

Sommaire

7 Index des fonctions

<u>7.97.24 sqlite_last_insert_rowid()</u> : Retourne le numéro de ligne de la dernière ligne insérée.....	1272
<u>7.97.25 sqlite_libencoding()</u> : Retourne l'encodage utilisé par la librairie SQLite.....	1272
<u>7.97.26 sqlite_libversion()</u> : Retourne la version de la librairie SQLite.....	1272
<u>7.97.27 sqlite_next()</u> : Déplace le pointeur SQLite vers la prochaine ligne.....	1272
<u>7.97.28 sqlite_num_fields()</u> : Retourne le nombre de champs dans un résultat SQLite.....	1273
<u>7.97.29 sqlite_num_rows()</u> : Retourne le nombre de lignes d'un résultat SQLite.....	1273
<u>7.97.30 sqlite_open()</u> : Ouvre une base SQLite.....	1273
<u>7.97.31 sqlite_popen()</u> : Ouvre une connexion SQLite persistante.....	1274
<u>7.97.32 sqlite_query()</u> : Exécute une requête SQLite et lit le résultat.....	1274
<u>7.97.33 sqlite_rewind()</u> : Place le pointeur de résultat SQLite au début.....	1275
<u>7.97.34 sqlite_seek()</u> : Déplace le pointeur de résultat SQLite vers une ligne.....	1275
<u>7.97.35 sqlite_udf_decode_binary()</u> : Décode des données binaires, passées à une UDF SQLite.....	1276
<u>7.97.36 sqlite_udf_encode_binary()</u> : Encode les données binaires avant de les retourner d'une UDF SQLite.....	1276
<u>7.97.37 sqlite_unbuffered_query()</u> : Exécute une requête SQLite non bufferisée.....	1277
<u>7.98 Shockwave Flash</u>	1278
<u>7.98.1 Introduction</u>	1278
<u>7.98.2 Pré-requis</u>	1278
<u>7.98.3 Installation</u>	1278
<u>7.98.4 Configuration à l'exécution</u>	1278
<u>7.98.5 Types de ressources</u>	1278
<u>7.98.6 Constantes prédéfinies</u>	1278
<u>7.98.7 Exemples</u>	1279
<u>7.98.9 swf_actiongotoframe()</u> : Joue un frame puis stoppe.....	1281
<u>7.98.10 swf_actiongotolabel()</u> : Affiche le frame nommé.....	1281
<u>7.98.11 swf_actionnextframe()</u> : Avance d'un frame.....	1281
<u>7.98.12 swf_actionplay()</u> : Joue l'animation flash à partir du frame courant.....	1281
<u>7.98.13 swf_actionprevframe()</u> : Recule d'un frame.....	1281
<u>7.98.14 swf_actionsettarget()</u> : Fixe le contexte des actions.....	1282
<u>7.98.15 swf_actionstop()</u> : Arrête l'animation flash.....	1282
<u>7.98.16 swf_actiontogglequality()</u> : Choisit le niveau de qualité haut ou bas.....	1282
<u>7.98.17 swf_actionwaitforframe()</u> : Ignore les actions si le frame n'est pas chargé.....	1282
<u>7.98.18 swf_addbuttonrecord()</u> : Contrôle la situation, l'apparence et la zone active du bouton courant.....	1282
<u>7.98.19 swf_addcolor()</u> : Fixe la couleur globale d'addition (? : the global add color).....	1283
<u>7.98.20 swf_closefile()</u> : Ferme le fichier courant Shockwave Flash.....	1283
<u>7.98.21 swf_definebitmap()</u> : Définit une image bitmap.....	1284
<u>7.98.22 swf_definefont()</u> : Définit une police.....	1284
<u>7.98.23 swf_defineline()</u> : Définit une ligne.....	1284
<u>7.98.24 swf_definepoly()</u> : Définit un polygone.....	1284
<u>7.98.25 swf_definerect()</u> : Définit un rectangle.....	1284
<u>7.98.26 swf_definetext()</u> : Définit une chaîne de texte.....	1284
<u>7.98.27 swf_endbutton()</u> : Termine la définition du bouton courant.....	1285
<u>7.98.28 swf_enddoaction()</u> : Termine l'action courante.....	1285
<u>7.98.29 swf_endshape()</u> : Complète la définition de la forme courante.....	1285
<u>7.98.30 swf_endsymbol()</u> : Termine la définition de symbole.....	1285

Sommaire

7 Index des fonctions

<u>7.98.31 swf_fontsize()</u> : Change la taille de la police.....	1285
<u>7.98.32 swf_fontslant()</u> : Change l'inclinaison de la police courante.....	1285
<u>7.98.33 swf_fonttracking()</u> : Change l'espacement des caractères.....	1285
<u>7.98.34 swf_getbitmapinfo()</u> : Lit les informations sur une image.....	1286
<u>7.98.35 swf_getfontinfo()</u> : Retourne la hauteur du A majuscule, et du x minuscule.....	1286
<u>7.98.36 swf_getframe()</u> : Retourne le numéro de frame courant.....	1286
<u>7.98.37 swf_labelframe()</u> : Nomme le frame courant.....	1286
<u>7.98.38 swf_lookat()</u> : Définit une transformation de vue.....	1286
<u>7.98.39 swf_modifyobject()</u> : Modifie un objet.....	1286
<u>7.98.40 swf_mulcolor()</u> : Fixe la couleur globale de multiplication (? : the global multiply color).....	1287
<u>7.98.41 swf_nextid()</u> : Retourne le prochain identifiant d'objet libre.....	1287
<u>7.98.42 swf_oncondition()</u> : Décrit une transition utilisée pour déclencher une liste d'actions.....	1287
<u>7.98.43 swf_openfile()</u> : Ouvre un nouveau fichier Shockwave Flash.....	1288
<u>7.98.44 swf_ortho()</u> : Définit une projection orthogonale entre les coordonnées utilisateur et le port courant.....	1288
<u>7.98.45 swf_ortho2()</u> : Définit une projection orthogonale à 2 dimensions entre les coordonnées utilisateur et le port courant.....	1288
<u>7.98.46 swf_perspective()</u> : Définit une projection orthogonale à 3 dimensions entre les coordonnées utilisateur et le port courant.....	1288
<u>7.98.47 swf_placeobject()</u> : Place un objet sur la scène.....	1288
<u>7.98.48 swf_polarview()</u> : Définit le point de vue de l'utilisateur en coordonnées polaire.....	1289
<u>7.98.49 swf_popmatrix()</u> : Dépile la matrice de transformation.....	1289
<u>7.98.50 swf_posround()</u> : Active l'approximation des translation d'objets.....	1289
<u>7.98.51 swf_pushmatrix()</u> : Empile la matrice de transformation courante dans la pile.....	1289
<u>7.98.52 swf_removeobject()</u> : Enlève un objet.....	1289
<u>7.98.53 swf_rotate()</u> : Rotation de la transformation courante.....	1290
<u>7.98.54 swf_scale()</u> : Homothétie.....	1290
<u>7.98.55 swf_setfont()</u> : Change la police courante.....	1290
<u>7.98.56 swf_setframe()</u> : Fixe le frame courant.....	1290
<u>7.98.57 swf_shapearc()</u> : Dessine une arc de cercle.....	1290
<u>7.98.58 swf_shapecurveto()</u> : Dessine une courbe de Bézier quadratique entre deux points.....	1290
<u>7.98.59 swf_shapecurveto3()</u> : Dessine une courbe Bézier cubique.....	1290
<u>7.98.60 swf_shapefillbitmapclip()</u> : Choisit le mode de remplissage par texture.....	1291
<u>7.98.61 swf_shapefillbitmaptile()</u> : Choisit le mode de remplissage par texture répétée.....	1291
<u>7.98.62 swf_shapefilloff()</u> : Désactive le remplissage.....	1291
<u>7.98.63 swf_shapefillsolid()</u> : Fixe la couleur pour le style courant de remplissage.....	1291
<u>7.98.64 swf_shapelinesolid()</u> : Fixe le style courant de ligne.....	1291
<u>7.98.65 swf_shapelineto()</u> : Dessine une ligne.....	1291
<u>7.98.66 swf_shapemoveto()</u> : Change la position courante.....	1292
<u>7.98.67 swf_showframe()</u> : Affiche le frame courant.....	1292
<u>7.98.68 swf_startbutton()</u> : Commence la définition d'un bouton.....	1292
<u>7.98.69 swf_startdoaction()</u> : Commence la description d'une liste d'action pour la frame courante.....	1292

Sommaire

7 Index des fonctions

<u>7.98.70 swf_startshape()</u> : Commence une forme complexe.....	1292
<u>7.98.71 swf_startsymbol()</u> : Définit un symbole.....	1292
<u>7.98.72 swf_textwidth()</u> : Retourne la longueur d'une chaîne.....	1292
<u>7.98.73 swf_translate()</u> : Translate la transformation courante.....	1293
<u>7.98.74 swf_viewport()</u> : Sélectionne une nouvelle zone pour un dessin ultérieur.....	1293
<u>7.99 SNMP</u>	1294
<u>7.99.1 Introduction</u>	1294
<u>7.99.2 Pré-requis</u>	1294
<u>7.99.3 Installation</u>	1294
<u>7.99.4 Configuration à l'exécution</u>	1294
<u>7.99.5 Types de ressources</u>	1294
<u>7.99.6 Constantes prédefinies</u>	1294
<u>7.99.8 snmp_set_quick_print()</u> : Ecrit la valeur courante de l'option quick_print de la librairie UCD.....	1295
<u>7.99.9 snmpget()</u> : Reçoit un objet SNMP.....	1295
<u>7.99.10 snmprealwalk()</u> : Retourne tous les objets, y compris leur ID d'objet.....	1296
<u>7.99.11 snmpset()</u> : Envoie un objet SNMP.....	1296
<u>7.99.12 snmpwalk()</u> : Reçoit tous les objets SNMP d'un agent.....	1296
<u>7.99.13 snmpwalkoid()</u> : Demande d'informations d'arbre sur une entité du réseau.....	1297
<u>7.100 Socket</u>	1298
<u>7.100.1 Introduction</u>	1298
<u>7.100.2 Pré-requis</u>	1298
<u>7.100.3 Installation</u>	1298
<u>7.100.4 Configuration à l'exécution</u>	1298
<u>7.100.5 Types de ressources</u>	1298
<u>7.100.6 Constantes prédefinies</u>	1299
<u>7.100.7 Socket Errors</u>	1299
<u>7.100.8 Exemples</u>	1300
<u>7.100.10 socket_bind()</u> : Lie un nom à une socket.....	1302
<u>7.100.11 socket_clear_error()</u> : Efface toutes les erreurs précédemment générées par une socket.....	1302
<u>7.100.12 socket_close()</u> : Ferme une socket.....	1303
<u>7.100.13 socket_connect()</u> : Crée une connexion sur une socket.....	1303
<u>7.100.14 socket_create_listen()</u> : Ouvre une socket sur un port pour accepter les connexions.....	1304
<u>7.100.15 socket_create_pair()</u> : Crée une paire de socket identiques et les stocke dans fds.....	1304
<u>7.100.16 socket_create()</u> : Crée une socket (point de communication).....	1304
<u>7.100.17 socket_get_option()</u> : Lit les options de socket.....	1305
<u>7.100.18 socket_getpeername()</u> : Interroge l'autre extrémité de la communication.....	1305
<u>7.100.19 socket_getsockname()</u> : Interroge la socket locale.....	1306
<u>7.100.20 socket_iovec_add()</u> : Ajoute un nouveau vecteur dans le tableau.....	1306
<u>7.100.21 socket_iovec_alloc()</u> : Construit un 'struct iovec'.....	1306
<u>7.100.22 socket_iovec_delete()</u> : Efface un vecteur.....	1307
<u>7.100.23 socket_iovec_fetch()</u> : Lit les données du vecteur spécifié par iovec_id[iovec_position].....	1307
<u>7.100.24 socket_iovec_free()</u> : Détruit le vecteur iovec spécifié par iovec_id.....	1307
<u>7.100.25 socket_iovec_set()</u> : Modifie les données du vecteur iovec_id[iovec_position].....	1307

Sommaire

7 Index des fonctions

<u>7.100.26 socket_last_error()</u> : Lit la dernière erreur générée par une socket.....	1308
<u>7.100.27 socket_listen()</u> : Attend une connexion (listen) sur une socket.....	1308
<u>7.100.28 socket_read()</u> : Lit des données d'une socket.....	1309
<u>7.100.29 socket_readv()</u> : Lit des données dans un fd, en utilisant le tableau scatter-gather défini par iovec_id.....	1309
<u>7.100.30 socket_recv()</u> : Reçoit des données d'une socket connectée.....	1310
<u>7.100.31 socket_recvfrom()</u> : Reçoit des données d'une socket, connectée ou pas.....	1310
<u>7.100.32 socket_recvmsg()</u> : Reçoit des messages sur uen socket, qu'elle soit orientée connexion ou pas.....	1310
<u>7.100.33 socket_select()</u> : Exécute l'appel système select() un tableau de sockets avec une durée d'expiration.....	1310
<u>7.100.34 socket_send()</u> : Envoie des données à une socket connectée.....	1312
<u>7.100.35 socket_sendmsg()</u> : Envoie un message à une socket, qu'elle soit orientée connexion ou pas.....	1312
<u>7.100.36 socket_sendto()</u> : Envoie une message à une socket, qu'elle soit connectée ou pas.....	1313
<u>7.100.37 socket_set_block()</u> : Met la socket en mode bloquant.....	1313
<u>7.100.38 socket_set_nonblock()</u> : Sélectionne le mode non bloquant d'un descripteur de fichier.....	1314
<u>7.100.39 socket_set_option()</u> : Modifie les options de socket.....	1314
<u>7.100.40 socket_shutdown()</u> : Eteind une socket en lecture et/ou écriture.....	1314
<u>7.100.41 socket_strerror()</u> : Retourne une chaîne décrivant un message d'erreur....	1314
<u>7.100.42 socket_write()</u> : Ecrit dans une socket.....	1315
<u>7.100.43 socket_writev()</u> : Ecrit des données dans un descripteur de fichier, fd, en utilisant le tableau scatter-gather définie par iovec_id.....	1316
<u>7.101 Flôts</u>	1317
<u>7.101.1 Introduction</u>	1317
<u>7.101.2 Filtres de flôts</u>	1317
<u>7.101.3 Contextes de flôts</u>	1317
<u>7.101.4 Installation</u>	1318
<u>7.101.5 Classes Stream</u>	1318
<u>7.101.6 Constantes prédefinies</u>	1318
<u>7.101.7 Erreurs de flôts</u>	1318
<u>7.101.8 Exemples</u>	1319
<u>7.101.10 stream_context_get_options()</u> : Lit la valeur des options pour un flôt/wrapper/contexte.....	1320
<u>7.101.11 stream_context_set_option()</u> : Configure une option pour un flôt/wrapper/contexte.....	1320
<u>7.101.12 stream_context_set_params()</u> : Configure les paramètres pour un flôt/wrapper/contexte.....	1320
<u>7.101.13 stream_copy_to_stream()</u> : Copie des données depuis un flôt vers un autre.....	1321
<u>7.101.14 stream_filter_append()</u> : Attache un filtre à un flôt en fin de liste.....	1321
<u>7.101.15 stream_filter_prepend()</u> : Attache un filtre à un flôt en début de liste.....	1322
<u>7.101.16 stream_filter_register()</u> : Enregistre un filtre de flôt.....	1322
<u>7.101.17 stream_get_filters()</u> : Liste les filtres enregistrés.....	1324
<u>7.101.18 stream_get_line()</u> : Lit une ligne dans un flôt.....	1324
<u>7.101.19 stream_get_meta_data()</u> : Lit les en-têtes et données méta des flôts.....	1325
<u>7.101.20 stream_get_transports()</u> : Liste les gestionnaires de transports de sockets disponibles.....	1325

Sommaire

7 Index des fonctions

<u>7.101.21 stream_get_wrappers()</u> : Liste les gestionnaires de flôts.....	1326
<u>7.101.22 stream_register_wrapper()</u> : Alias de stream_wrapper_register.....	1326
<u>7.101.23 stream_select()</u> : Runs the equivalent of the select() system call on the given arrays of streams with a timeout specified by tv_sec and tv_usec.....	1326
<u>7.101.24 stream_set_blocking()</u> : Configure le mode bloquant d'un flôt.....	1327
<u>7.101.25 stream_set_timeout()</u> : Configure la durée d'expiration d'un flôt.....	1328
<u>7.101.26 stream_set_write_buffer()</u> : Configure la bufferisation de fichier pour un flôt.....	1328
<u>7.101.27 stream_socket_accept()</u> : Accepte une connexion sur une socket créée par stream_socket_server.....	1329
<u>7.101.28 stream_socket_client()</u> : Ouvre une connexion socket Internet ou Unix.....	1329
<u>7.101.29 stream_socket_get_name()</u> : Lit le nom des sockets locale ou distante.....	1330
<u>7.101.30 stream_socket_server()</u> : Crée une socket serveur Unix ou Internet.....	1331
<u>7.101.31 stream_wrapper_register()</u> : Enregistre une enveloppe URL, implémentée comme classe PHP.....	1332
<u>7.102 Chaînes de caractères (Strings)</u>	1336
<u>7.102.1 Introduction</u>	1336
<u>7.102.2 Pré-requis</u>	1336
<u>7.102.3 Installation</u>	1336
<u>7.102.4 Constantes prédefinies</u>	1336
<u>7.102.5 Voir aussi</u>	1337
<u>7.102.7 addslashes()</u> : Ajoute des anti-slashes dans une chaîne.....	1338
<u>7.102.8 bin2hex()</u> : Convertit des données binaires en représentation hexadécimale.....	1339
<u>7.102.9 chop()</u> : Alias de rtrim.....	1339
<u>7.102.10 chr()</u> : Retourne un caractère spécifique.....	1339
<u>7.102.11 chunk_split()</u> : Scinde une chaîne.....	1339
<u>7.102.12 convert_cyr_string()</u> : Convertit une chaîne d'un jeu de caractères cyrillique à l'autre.....	1340
<u>7.102.13 count_chars()</u> : Retourne des statistiques sur les caractères utilisés dans une chaîne.....	1340
<u>7.102.14 crc32()</u> : Calcule la somme de contrôle CRC32.....	1340
<u>7.102.15 crypt()</u> : Chiffage indéchiffrable (hashing).....	1341
<u>7.102.16 echo()</u> : Affiche une chaîne de caractères.....	1342
<u>7.102.17 explode()</u> : Coupe une chaîne en segments.....	1343
<u>7.102.18 fprintf()</u> : Ecrit une chaîne formatée dans un flôt.....	1343
<u>7.102.19 get_html_translation_table()</u> : Retourne la table de traduction des entités utilisée par htmlspecialchars et htmlentities.....	1344
<u>7.102.20 hebreve()</u> : Convertit un texte logique hébreux en texte visuel.....	1345
<u>7.102.21 hebrevc()</u> : Convertit un texte logique hébreux en texte visuel, avec retours à la ligne.....	1345
<u>7.102.22 html_entity_decode()</u> : Convertit toutes les entités HTML en caractères normaux.....	1345
<u>7.102.23 htmlentities()</u> : Convertit tous les caractères éligibles en entités HTML.....	1346
<u>7.102.24 htmlspecialchars()</u> : Convertit les caractères spéciaux en entités HTML.....	1347
<u>7.102.25 implode()</u> : Rassemble les éléments d'un tableau en une chaîne.....	1348
<u>7.102.26 join()</u> : Rassemble les éléments d'un tableau en une chaîne.....	1349
<u>7.102.27 levenshtein()</u> : Calcule la distance Levenshtein entre deux chaînes.....	1349
<u>7.102.28 localeconv()</u> : Lit la configuration locale.....	1350
<u>7.102.29 ltrim()</u> : Supprime les caractères invisibles de début de chaîne.....	1351
<u>7.102.30 md5_file()</u> : Calcule le md5 d'un fichier.....	1352

Sommaire

7 Index des fonctions

<u>7.102.31 md5()</u> : Calcule le md5 d'une chaîne.....	1352
<u>7.102.32 metaphone()</u> : Calcule la clé métaphone.....	1352
<u>7.102.33 money_format()</u> : Met un nombre au format monétaire.....	1353
<u>7.102.34 nl_langinfo()</u> : Rassemble des informations sur la langue et la configuration locale.....	1355
<u>7.102.35 nl2br()</u> : Insère un retour à la ligne HTML à chaque nouvelle ligne.....	1355
<u>7.102.36 number_format()</u> : Formate un nombre pour l'affichage.....	1355
<u>7.102.37 ord()</u> : Retourne le code ASCII d'un caractère.....	1356
<u>7.102.38 parse_str()</u> : Analyse une requête HTTP.....	1356
<u>7.102.39 print()</u> : Affiche une chaîne de caractères.....	1357
<u>7.102.40 printf()</u> : Affiche une chaîne de caractères formatée.....	1358
<u>7.102.41 quoted_printable_decode()</u> : Convertit une chaîne quoted-printable en chaîne 8 bits.....	1358
<u>7.102.42 quotemeta()</u> : Echappe les méta-caractères.....	1358
<u>7.102.43 rtrim()</u> : Supprime les espaces de fin de chaîne.....	1358
<u>7.102.44 setlocale()</u> : Modifie les informations de localisation.....	1359
<u>7.102.45 sha1_file()</u> : Calcule le sha1 d'un fichier.....	1360
<u>7.102.46 sha1()</u> : Calcule le sha1 d'une chaîne de caractères.....	1360
<u>7.102.47 similar_text()</u> : Calcule la similarité de deux chaînes.....	1361
<u>7.102.48 soundex()</u> : Calcule la clé soundex.....	1361
<u>7.102.49 sprintf()</u> : Retourne une chaîne formatée.....	1361
<u>7.102.50 sscanf()</u> : Analyse une chaîne à l'aide d'un format.....	1363
<u>7.102.51 str_ireplace()</u> : Version insensible à la casse de str_replace.....	1364
<u>7.102.52 str_pad()</u> : Complète une chaîne jusqu'à une taille donnée.....	1364
<u>7.102.53 str_repeat()</u> : Répète une chaîne.....	1365
<u>7.102.54 str_replace()</u> : Remplace toutes les occurrences dans une chaîne.....	1365
<u>7.102.55 str_rot13()</u> : Effectue une transformation rot13.....	1366
<u>7.102.56 str_shuffle()</u> : Mélange les caractères d'une chaîne de caractères.....	1366
<u>7.102.57 str_split()</u> : Convertit une chaîne de caractères en tableau.....	1366
<u>7.102.58 str_word_count()</u> : Compte le nombre de mots utilisés dans une chaîne.....	1367
<u>7.102.59 strcasecmp()</u> : Comparaison de chaînes binaires.....	1368
<u>7.102.60 strchr()</u> : Trouve la première occurrence d'un caractère dans une chaîne.....	1368
<u>7.102.61 strcmp()</u> : Comparaison binaire de chaînes.....	1369
<u>7.102.62 strcoll()</u> : Comparaison de chaînes localisée.....	1369
<u>7.102.63 strcspn()</u> : Trouve un segment de chaîne ne contenant pas certains caractères.....	1369
<u>7.102.64 strip_tags()</u> : Supprime les balises HTML et PHP d'une chaîne.....	1369
<u>7.102.65 stripslashes()</u> : Supprime les anti-slash d'une chaîne C.....	1370
<u>7.102.66 strpos()</u> : Recherche la position d'une occurrence dans une chaîne, sans tenir compte de la casse.....	1370
<u>7.102.67 stripslashes()</u> : Supprimer les anti-slash d'une chaîne.....	1371
<u>7.102.68 stristr()</u> : Trouve la première occurrence dans une chaîne (insensible à la casse).....	1371
<u>7.102.69 strlen()</u> : Calcule la taille d'une chaîne.....	1371
<u>7.102.70 strnatcasecmp()</u> : Comparaison de chaînes avec l'algorithme d'ordre naturel" (insensible à la casse).....	1372
<u>7.102.71 strnatcmp()</u> : Comparaison de chaînes avec l'algorithme d'ordre naturel".....	1372
<u>7.102.72 strncasecmp()</u> : Compare en binaire des chaînes de caractères.....	1373
<u>7.102.73 strncmp()</u> : Comparaison binaire des n premiers caractères.....	1373
<u>7.102.74 strpos()</u> : Trouve la position d'un caractère dans une chaîne.....	1373

Sommaire

7 Index des fonctions

<u>7.102.75 strrchr()</u> : Retourne la fin de la chaîne.....	1374
<u>7.102.76 strrev()</u> : Inverse une chaîne.....	1374
<u>7.102.77 stripos()</u> : Trouve la position de la dernière occurrence d'une chaîne dans une autre de façon insensible à la casse.....	1375
<u>7.102.78 strrpos()</u> : Trouve la position de la dernière occurrence d'un caractère dans une chaîne.....	1375
<u>7.102.79 strspn()</u> : Trouve le premier segment de chaîne.....	1376
<u>7.102.80 strstr()</u> : Trouve la première occurrence dans une chaîne.....	1376
<u>7.102.81 strtok()</u> : Coupe une chaîne en segments.....	1377
<u>7.102.82 strtolower()</u> : Renvoie une chaîne en minuscules.....	1378
<u>7.102.83 strtoupper()</u> : Renvoie une chaîne en majuscules.....	1378
<u>7.102.84 strtr()</u> : Remplace des chaînes dans une chaîne.....	1378
<u>7.102.85 substr_count()</u> : Compte de le nombre d'occurrences de segments dans une chaîne.....	1379
<u>7.102.86 substr_replace()</u> : Remplace un segment dans une chaîne.....	1379
<u>7.102.87 substr()</u> : Retourne un segment de chaîne.....	1380
<u>7.102.88 trim()</u> : Supprime les espaces en début et fin de chaîne.....	1381
<u>7.102.89 ucfirst()</u> : Met le premier caractère en majuscule.....	1381
<u>7.102.90 ucwords()</u> : Met en majuscule la première lettre de tous les mots.....	1382
<u>7.102.91 vprintf()</u> : Affiche une chaîne formatée.....	1382
<u>7.102.92 vsprintf()</u> : Retourne une chaîne formatée.....	1382
<u>7.102.93 wordwrap()</u> : Effectue la césure d'une chaîne.....	1383
<u>7.103 Sybase</u>	1384
<u>7.103.1 Introduction</u>	1384
<u>7.103.2 Pré-requis</u>	1384
<u>7.103.3 Installation</u>	1384
<u>7.103.4 Configuration à l'exécution</u>	1384
<u>7.103.5 Types de ressources</u>	1385
<u>7.103.6 Constantes prédefinies</u>	1385
<u>7.103.8 sybase_close()</u> : Ferme une connexion Sybase.....	1386
<u>7.103.9 sybase_connect()</u> : Ouvre une connexion à un serveur Sybase.....	1386
<u>7.103.10 sybase_data_seek()</u> : Déplace le pointeur interne de lignes Sybase.....	1387
<u>7.103.11 sybase_deadlock_retry_count()</u> : Configure le nombre de tentatives lors de blocages (deadlock).....	1387
<u>7.103.12 sybase_fetch_array()</u> : Retourne une ligne Sybase sous la forme d'un tableau.....	1387
<u>7.103.13 sybase_fetch_assoc()</u> : Lit une ligne de résultat Sybase sous forme de tableau associatif.....	1388
<u>7.103.14 sybase_fetch_field()</u> : Lit les informations d'un champ Sybase.....	1388
<u>7.103.15 sybase_fetch_object()</u> : Retourne une ligne Sybase sous la forme d'un objet.....	1389
<u>7.103.16 sybase_fetch_row()</u> : Retourne une ligne Sybase sous la forme d'un tableau numérique.....	1390
<u>7.103.17 sybase_field_seek()</u> : Modifie l'index d'un champ Sybase.....	1390
<u>7.103.18 sybase_free_result()</u> : Libère un résultat Sybase de la mémoire.....	1390
<u>7.103.19 sybase_get_last_message()</u> : Retourne le dernier message du serveur.....	1390
<u>7.103.20 sybase_min_client_severity()</u> : Fixe la sévérité minimale du client Sybase.....	1391
<u>7.103.21 sybase_min_error_severity()</u> : Fixe la sévérité minimale du client pour les erreurs.....	1391
<u>7.103.22 sybase_min_message_severity()</u> : Fixe la sévérité minimale du client	

Sommaire

7 Index des fonctions

<u>pour les messages</u>	1391
<u>7.103.23 sybase_min_server_severity() : Fixe la sévérité minimale du client pour le serveur Sybase</u>	1391
<u>7.103.24 sybase_num_fields() : Retourne le nombre de champs dans un résultat Sybase</u>	1391
<u>7.103.25 sybase_num_rows() : Retourne le nombre de lignes dans un résultat Sybase</u>	1392
<u>7.103.26 sybase_pconnect() : Ouvre une connexion persistante à un serveur Sybase</u>	1392
<u>7.103.27 sybase_query() : Envoie une requête à une base Sybase</u>	1392
<u>7.103.28 sybase_result() : Lit une valeur dans un résultat</u>	1392
<u>7.103.29 sybase_select_db() : Sélectionne une base de données Sybase</u>	1393
<u>7.103.30 sybase_set_message_handler() : Configure le gestionnaire de message Sybase</u>	1393
<u>7.103.31 sybase_unbuffered_query() : Envoie une requête à Sybase et ne bloque pas</u>	1394
<u>7.104 Analyseur de code PHP</u>	1395
<u>7.104.1 Introduction</u>	1395
<u>7.104.2 Pré-requis</u>	1395
<u>7.104.3 Installation</u>	1395
<u>7.104.4 Constantes prédefinies</u>	1395
<u>7.104.5 Exemples</u>	1397
<u>7.104.7 token_name() : Lit le nom d'un élément de code source</u>	1398
<u>7.105 URL</u>	1399
<u>7.105.1 Introduction</u>	1399
<u>7.105.2 Pré-requis</u>	1399
<u>7.105.3 Installation</u>	1399
<u>7.105.4 Configuration à l'exécution</u>	1399
<u>7.105.5 Types de ressources</u>	1399
<u>7.105.6 Constantes prédefinies</u>	1399
<u>7.105.8 base64_encode() : Encode une chaîne en MIME base64</u>	1400
<u>7.105.9 get_meta_tags() : Extrait tous les méta tags d'un fichier</u>	1400
<u>7.105.10 http_build_query() : Génère une chaîne en encodage URL</u>	1401
<u>7.105.11 parse_url() : Analyse une URL et retourne ses composants</u>	1402
<u>7.105.12 rawurldecode() : Décode une chaîne URL</u>	1403
<u>7.105.13 rawurlencode() : Encode une chaîne en URL, selon la RFC1738</u>	1403
<u>7.105.14 urldecode() : Décode une chaîne encodée URL</u>	1403
<u>7.105.15 urlencode() : Encode une chaîne en URL</u>	1404
<u>7.106 Variables</u>	1405
<u>7.106.1 Introduction</u>	1405
<u>7.106.2 Pré-requis</u>	1405
<u>7.106.3 Installation</u>	1405
<u>7.106.4 Configuration à l'exécution</u>	1405
<u>7.106.5 Types de ressources</u>	1405
<u>7.106.6 Constantes prédefinies</u>	1405
<u>7.106.8 empty() : Détermine si une variable contient une valeur non nulle</u>	1406
<u>7.106.9 floatval() : Convertit une chaîne en nombre à virgule flottante</u>	1407
<u>7.106.10 get_defined_vars() : Liste toutes les variables définies</u>	1407
<u>7.106.11 get_resource_type() : Retourne le type de ressource</u>	1408
<u>7.106.12 gettype() : Retourne le type de la variable</u>	1408
<u>7.106.13 import_request_variables() : Importe les variables de</u>	

Sommaire

7 Index des fonctions

<u>GET/POST/Cookie dans l'environnement global</u>	1409
<u>7.106.14 intval() : Retourne la valeur numérique entière équivalent d'une variable</u>	1409
<u>7.106.15 is_array() : Détermine si une variable est un tableau</u>	1410
<u>7.106.16 is_bool() : Détermine si une variable est un booléen</u>	1410
<u>7.106.17 is_callable() : Détermine si l'argument peut être appelé comme fonction</u> ..	1410
<u>7.106.18 is_double() : Alias de is_float</u>	1411
<u>7.106.19 is_float() : Détermine si une variable est de type float</u>	1411
<u>7.106.20 is_int() : Détermine si une variable est de type integer</u>	1411
<u>7.106.21 is_integer() : Détermine si une variable est de type int</u>	1411
<u>7.106.22 is_long() : Alias de is_int</u>	1412
<u>7.106.23 is_null() : Indique si une variable vaut NULL</u>	1412
<u>7.106.24 is_numeric() : Détermine si une variable est un type numérique</u>	1412
<u>7.106.25 is_object() : Détermine si une variable est de type object</u>	1412
<u>7.106.26 is_real() : Détermine si une variable est de type real</u>	1412
<u>7.106.27 is_resource() : Détermine si une variable est une ressource</u>	1412
<u>7.106.28 is_scalar() : Indique si une variable est un scalaire</u>	1413
<u>7.106.29 is_string() : Détermine si une variable est de type chaîne</u>	1413
<u>7.106.30 isset() : Détermine si une variable est affectée</u>	1413
<u>7.106.31 print_r() : Affiche des informations lisibles pour une variable</u>	1414
<u>7.106.32 serialize() : Linéarise une variable</u>	1415
<u>7.106.33 settype() : Affecte un type à une variable</u>	1416
<u>7.106.34 strval() : Retourne la valeur de la variable, au format chaîne</u>	1416
<u>7.106.35 unserialize() : Crée une variable PHP à partir d'une valeur linéarisée</u>	1417
<u>7.106.36 unset() : Détruit une variable</u>	1418
<u>7.106.37 var_dump() : Affiche les informations d'une variable</u>	1419
<u>7.106.38 var_export() : Retourne le code PHP utilisé pour générer une variable</u>	1420
<u>7.107 vpopmail</u>	1421
<u>7.107.1 Introduction</u>	1421
<u>7.107.2 Installation</u>	1421
<u>7.107.4 vpopmail_add_alias_domain() : Ajout un alias pour un domaine virtuel</u>	1421
<u>7.107.5 vpopmail_add_domain_ex() : Ajoute un nouveau domaine virtuel</u>	1422
<u>7.107.6 vpopmail_add_domain() : Ajoute un nouveau domaine virtuel</u>	1422
<u>7.107.7 vpopmail_add_user() : Ajoute un nouvel utilisateur à un domaine virtuel</u>	1422
<u>7.107.8 vpopmail_alias_add() : Ajoute un alias virtuel</u>	1422
<u>7.107.9 vpopmail_alias_del_domain() : Efface tous les alias virtuels d'un domaine</u>	1423
<u>7.107.10 vpopmail_alias_del() : Efface tous les alias d'un utilisateur</u>	1423
<u>7.107.11 vpopmail_alias_get_all() : Lit toutes les lignes d'alias d'un domaine</u>	1423
<u>7.107.12 vpopmail_alias_get() : Lit toutes les lignes d'un alias de domaine</u>	1423
<u>7.107.13 vpopmail_auth_user() : Valide le mot de passe d'un utilisateur pour un domaine. Retourne TRUE / FALSE</u>	1424
<u>7.107.14 vpopmail_del_domain_ex() : Efface un domaine virtuel</u>	1424
<u>7.107.15 vpopmail_del_domain() : Efface un domaine virtuel</u>	1424
<u>7.107.16 vpopmail_del_user() : Efface un utilisateur d'un domaine virtuel</u>	1424
<u>7.107.17 vpopmail_error() : Lit le dernier message d'erreur vpopmail</u>	1425
<u>7.107.18 vpopmail_passwd() : Remplace le mot de passe d'un utilisateur virtuel</u>	1425
<u>7.107.19 vpopmail_set_user_quota() : Modifie le quota d'un utilisateur virtuel</u>	1425
<u>7.108 API windows</u>	1426
<u>7.108.1 Introduction</u>	1426
<u>7.108.2 Pré-requis</u>	1426

Sommaire

7 Index des fonctions

<u>7.108.3 Installation</u>	1426
<u>7.108.4 Configuration à l'exécution</u>	1426
<u>7.108.5 Types de ressources</u>	1426
<u>7.108.6 Constantes prédefinies</u>	1426
<u>7.108.7 Exemples</u>	1427
<u>7.108.9 w32api_init_dtype() : Crée une instance de type de données et la remplit</u> ...	1427
<u>7.108.10 w32api_invoke_function() : Appelle une fonction windows 32</u>	1428
<u>7.108.11 w32api_register_function() : Enregistre une fonction win32 dans PHP</u>	1428
<u>7.108.12 w32api_set_call_method() : Modifie le type d'appel de la fonction</u>	1428
<u>7.109 WDDX</u>	1430
<u>7.109.1 Introduction</u>	1430
<u>7.109.2 Pré-requis</u>	1430
<u>7.109.3 Installation</u>	1430
<u>7.109.4 Configuration à l'exécution</u>	1430
<u>7.109.5 Types de ressources</u>	1430
<u>7.109.6 Constantes prédefinies</u>	1430
<u>7.109.7 Exemples</u>	1430
<u>7.109.9 wddx_deserialize() : Lire un paquet WDDX</u>	1431
<u>7.109.10 wddx_packet_end() : Clos un paquet WDDX</u>	1431
<u>7.109.11 wddx_packet_start() : Commencer un nouveau paquet WDDX avec une structure</u>	1431
<u>7.109.12 wddx_serialize_value() : Enregistrer une valeur dans un paquet WDDX</u>	1432
<u>7.109.13 wddx_serialize_vars() : Enregistrer plusieurs valeurs dans un paquet WDDX</u>	1432
<u>7.110 Analyseur syntaxique XML</u>	1433
<u>7.110.1 Introduction</u>	1433
<u>7.110.2 Pré-requis</u>	1433
<u>7.110.3 Installation</u>	1433
<u>7.110.4 Configuration à l'exécution</u>	1433
<u>7.110.5 Types de ressources</u>	1433
<u>7.110.6 Constantes prédefinies</u>	1434
<u>7.110.7 gestionnaires d'évènements</u>	1434
<u>7.110.8 Problèmes de casse</u>	1435
<u>7.110.9 Codes d'erreurs</u>	1435
<u>7.110.10 Codage des caractères</u>	1435
<u>7.110.11 Exemples</u>	1436
<u>7.110.13 utf8_encode() : Convertit une chaîne ISO-8859-1 en UTF-8</u>	1440
<u>7.110.14 xml_error_string() : Lit le message d'erreur de l'analyseur XML</u>	1440
<u>7.110.15 xml_get_current_byte_index() : Retourne l'index de l'octet courant d'un analyseur XML</u>	1440
<u>7.110.16 xml_get_current_column_number() : Retourne le nombre courant de colonne d'un analyseur XML</u>	1440
<u>7.110.17 xml_get_current_line_number() : Retourne le numéro de ligne courant d'un analyseur XML</u>	1441
<u>7.110.18 xml_get_error_code() : Retourne le nombre courant de colonne d'un analyseur XML</u>	1441
<u>7.110.19 xml_parse_into_struct() : Analyse une structure XML</u>	1441
<u>7.110.20 xml_parse() : Commence l'analyse d'un fichier XML</u>	1443
<u>7.110.21 xml_parser_create_ns() : Crée un analyseur XML</u>	1444
<u>7.110.22 xml_parser_create() : Création d'un analyseur XML</u>	1444
<u>7.110.23 xml_parser_free() : Détruit un analyseur XML</u>	1444

Sommaire

7 Index des fonctions

<u>7.110.24 xml_parser_get_option()</u> : Lit les options d'un analyseur XML.....	1444
<u>7.110.25 xml_parser_set_option()</u> : Affecte les options d'un analyseur XML.....	1445
<u>7.110.26 xml_set_character_data_handler()</u> : Affecte les gestionnaires de texte littéral.....	1445
<u>7.110.27 xml_set_default_handler()</u> : Affecte le gestionnaire XML par défaut.....	1446
<u>7.110.28 xml_set_element_handler()</u> : Affecte les gestionnaires de début et de fin de balise XML.....	1446
<u>7.110.29 xml_set_end_namespace_decl_handler()</u> : Configure le gestionnaire XML de données.....	1447
<u>7.110.30 xml_set_external_entity_ref_handler()</u> : Configure le gestionnaire XML de références externes.....	1447
<u>7.110.31 xml_set_notation_decl_handler()</u> : Configure le gestionnaire XML de notations.....	1448
<u>7.110.32 xml_set_object()</u> : Utilise un analyseur XML à l'intérieur d'un objet.....	1449
<u>7.110.33 xml_set_processing_instruction_handler()</u> : Affecte les gestionnaires d'instructions exécutables.....	1449
<u>7.110.34 xml_set_start_namespace_decl_handler()</u> : Configure le gestionnaire de caractères.....	1450
<u>7.110.35 xml_set_unparsed_entity_decl_handler()</u> : Affecte les gestionnaires d'entités non déclarés.....	1450
<u>7.111 XML-RPC</u>	1452
<u>7.111.1 Introduction</u>	1452
<u>7.111.2 Pré-requis</u>	1452
<u>7.111.3 Installation</u>	1452
<u>7.111.4 Configuration à l'exécution</u>	1452
<u>7.111.5 Types de ressources</u>	1452
<u>7.111.6 Constantes prédefinies</u>	1452
<u>7.111.8 xmlrpc_decode()</u> : Décode le code XML en types PHP natifs.....	1453
<u>7.111.9 xmlrpc_encode_request()</u> : Génère le code XML d'une requête de méthode.....	1453
<u>7.111.10 xmlrpc_encode()</u> : Génère le code XML pour une valeur PHP.....	1453
<u>7.111.11 xmlrpc_get_type()</u> : Retourne le type XMLRPC d'une valeur PHP. Particulièrement pratique pour les types chaînes de type base64 et datetime....	1454
<u>7.111.12 xmlrpc_parse_method_descriptions()</u> : Décode le code XML en une liste de descriptions de méthodes.....	1454
<u>7.111.13 xmlrpc_server_add_introspection_data()</u> : Ajoute des données d'introspection.....	1454
<u>7.111.14 xmlrpc_server_call_method()</u> : Analyse une requête XML est appelle les méthodes associées.....	1454
<u>7.111.15 xmlrpc_server_create()</u> : Crée un serveur XMLRPC.....	1455
<u>7.111.16 xmlrpc_server_destroy()</u> : Détruit un serveur XMLRPC.....	1455
<u>7.111.17 xmlrpc_server_register_introspection_callback()</u> : Enregistre une fonction PHP pour générer la documentation.....	1455
<u>7.111.18 xmlrpc_server_register_method()</u> : Enregistre une fonction PHP avec une méthode.....	1455
<u>7.111.19 xmlrpc_set_type()</u> : Modifie le type XMLRPC, base64 ou datetime, pour une chaînes de caractères PHP.....	1456
<u>7.112 XSLT</u>	1457
<u>7.112.1 Introduction</u>	1457
<u>7.112.2 Pré-requis</u>	1457
<u>7.112.3 Installation</u>	1457

Sommaire

7 Index des fonctions

<u>7.112.4 Configuration à l'exécution</u>	1457
<u>7.112.5 Types de ressources</u>	1458
<u>7.112.6 Constantes prédefinies</u>	1458
<u>7.112.8 xslt_erro() : Retourne le numéro d'erreur XSLT courant</u>	1458
<u>7.112.9 xslt_error() : Retourne le message d'erreur courant</u>	1459
<u>7.112.10 xslt_free() : Détruit un analyseur XSLT</u>	1459
<u>7.112.11 xslt_process() : Transforme des données XML avec XSLT</u>	1459
<u>7.112.12 xslt_set_base() : Configure l'URI de base de toutes les transformations XSLT</u>	1461
<u>7.112.13 xslt_set_encoding() : Configure le jeu de caractères pour l'analyse des documents XML</u>	1461
<u>7.112.14 xslt_set_error_handler() : Configure le gestionnaire d'erreurs du processeur XSLT</u>	1461
<u>7.112.15 xslt_set_log() : Configure le fichier d'historique pour les messages XSLT</u>	1462
<u>7.112.16 xslt_set_sax_handler() : Modifie les gestionnaires SAX de l'analyseur XSLT</u>	1462
<u>7.112.17 xslt_set_sax_handlers() : Configure les gestionnaires SAX qui seront appelés pour gérer les documents XML</u>	1463
<u>7.112.18 xslt_set_scheme_handler() : Configure les gestionnaires de Scheme du processeur XSLT</u>	1463
<u>7.112.19 xslt_set_scheme_handlers() : Configure un gestionnaire de Scheme pour un processeur XSLT</u>	1463
<u>7.113 YAZ</u>	1464
<u>7.113.1 Introduction</u>	1464
<u>7.113.2 Installation</u>	1464
<u>7.113.3 Configuration à l'exécution</u>	1464
<u>7.113.4 Types de ressources</u>	1465
<u>7.113.5 Constantes prédefinies</u>	1465
<u>7.113.6 Exemples</u>	1465
<u>7.113.8 yaz_ccl_conf() : Configure l'analyseur CCL</u>	1466
<u>7.113.9 yaz_ccl_parse() : Appelle l'analyseur CCL</u>	1467
<u>7.113.10 yaz_close() : Ferme une connexion YAZ</u>	1467
<u>7.113.11 yaz_connect() : Prépare une connexion à un hôte YAZ</u>	1467
<u>7.113.12 yaz_database() : Spécifie la base d'une session YAZ</u>	1468
<u>7.113.13 yaz_element() : Spécifie le type d'éléments YAZ à lire</u>	1468
<u>7.113.14 yaz_erro() : Retourne le numéro d'erreur</u>	1468
<u>7.113.15 yaz_error() : Retourne une description de l'erreur</u>	1469
<u>7.113.16 yaz_get_option() : Retourne la valeur d'une option de connexion YAZ</u>	1469
<u>7.113.17 yaz_hits() : Retourne le nombre de résultats de la dernière recherche</u>	1469
<u>7.113.18 yaz_itemorder() : Prépare une requête Z39.50 Item Order avec le package ILL-Request</u>	1469
<u>7.113.19 yaz_present() : Prépare à la lecture (Z39.50 present)</u>	1471
<u>7.113.20 yaz_range() : Spécifie le nombre maximal de résultat à lire</u>	1471
<u>7.113.21 yaz_record() : Retourne un résultat</u>	1471
<u>7.113.22 yaz_scan_result() : Retourne le résultat d'un scan</u>	1471
<u>7.113.23 yaz_scan() : Prépare un scan</u>	1472
<u>7.113.24 yaz_schema() : Spécifie le schema de lecture</u>	1472
<u>7.113.25 yaz_search() : Prépare une recherche</u>	1473
<u>7.113.26 yaz_set_option() : Modifie une ou plusieurs options de connexion YAZ</u>	1473
<u>7.113.27 yaz_sort() : Configure les critères de tri</u>	1473

Sommaire

7 Index des fonctions

<u>7.113.28 yaz_syntax()</u> : Spécifie la syntaxe de lecture des lignes.....	1474
<u>7.113.29 yaz_wait()</u> : Attend l'exécution d'une requête.....	1474
<u>7.114 NIS</u>	1475
<u>7.114.1 Introduction</u>	1475
<u>7.114.2 Pré-requis</u>	1475
<u>7.114.3 Installation</u>	1475
<u>7.114.4 Configuration à l'exécution</u>	1475
<u>7.114.5 Constantes prédefinies</u>	1475
<u>7.114.7 yp_cat()</u> : Retourne un tableau contenant toute la carte.....	1476
<u>7.114.8 yp_err_string()</u> : Retourne la chaîne correspondante à l'erreur donnée.....	1476
<u>7.114.9 yp_errno()</u> : Retourne le code d'erreur de la dernière opération NIS.....	1476
<u>7.114.10 yp_first()</u> : Retourne le premier couple (clé ; valeur) d'une carte donnée.....	1477
<u>7.114.11 yp_get_default_domain()</u> : Retourne le domaine NIS par défaut.....	1477
<u>7.114.12 yp_master()</u> : Retourne le nom de la machine maître pour une carte.....	1478
<u>7.114.13 yp_match()</u> : Retourne la ligne associée.....	1478
<u>7.114.14 yp_next()</u> : Retourne le couple (clé ; valeur) suivant d'une carte donnée.....	1478
<u>7.114.15 yp_order()</u> : Retourne le numéro d'ordre d'une carte.....	1479
<u>7.115 Accès aux fichiers Zip (Lecture seule)</u>	1480
<u>7.115.1 Introduction</u>	1480
<u>7.115.2 Pré-requis</u>	1480
<u>7.115.3 Installation</u>	1480
<u>7.115.4 Configuration à l'exécution</u>	1480
<u>7.115.5 Types de ressources</u>	1480
<u>7.115.6 Constantes prédefinies</u>	1480
<u>7.115.7 Exemples</u>	1480
<u>7.115.9 zip_entry_close()</u> : Ferme un dossier d'archive.....	1481
<u>7.115.10 zip_entry_compressedsize()</u> : Lit la taille compressée d'un dossier d'archive.....	1481
<u>7.115.11 zip_entry_compressionmethod()</u> : Lit la méthode de compression utilisée sur un dossier d'archive.....	1482
<u>7.115.12 zip_entry_filesize()</u> : Lit la taille décompressée d'un dossier d'archive.....	1482
<u>7.115.13 zip_entry_name()</u> : Lit le nom du dossier d'archive.....	1482
<u>7.115.14 zip_entry_open()</u> : Ouvre un fichier d'archive en lecture.....	1482
<u>7.115.15 zip_entry_read()</u> : Lit le contenu d'un fichier dans un dossier.....	1483
<u>7.115.16 zip_open()</u> : Ouvre une archive ZIP.....	1483
<u>7.115.17 zip_read()</u> : Lit la prochaine entrée dans une archive ZIP.....	1483
<u>7.116 Compression Zlib</u>	1484
<u>7.116.1 Introduction</u>	1484
<u>7.116.2 Pré-requis</u>	1484
<u>7.116.3 Installation</u>	1484
<u>7.116.4 Configuration à l'exécution</u>	1484
<u>7.116.5 Types de ressources</u>	1485
<u>7.116.6 Constantes prédefinies</u>	1485
<u>7.116.7 Exemples</u>	1485
<u>7.116.9 gzcompress()</u> : Comprime une chaîne.....	1486
<u>7.116.10 gzdeflate()</u> : Comprime une chaîne avec la méthode DEFLATE.....	1487
<u>7.116.11 gzencode()</u> : Crée une chaîne compressée.....	1487
<u>7.116.12 gzeof()</u> : Indique si la fin d'un fichier compressé est atteinte.....	1488
<u>7.116.13 gzfile()</u> : Lit la totalité d'un fichier compressé.....	1488

Sommaire

7 Index des fonctions

<u>7.116.14 gzgetc() : Lit un caractère dans un fichier compressé.....</u>	1488
<u>7.116.15 gzgets() : Lit une ligne dans un fichier compressé.....</u>	1488
<u>7.116.16 gzgetss() : Lit une ligne dans un fichier compressé, et supprime les balises.....</u>	1489
<u>7.116.17 gzinflate() : Décompresse une chaîne.....</u>	1489
<u>7.116.18 gzopen() : Ouvre un fichier compressé avec gzip.....</u>	1489
<u>7.116.19 gzpassthru() : Affiche le contenu d'un fichier en une seule fois.....</u>	1490
<u>7.116.20 gzputs() : Ecrit une chaîne dans un fichier compressé.....</u>	1490
<u>7.116.21 gzread() : Lecture de fichier compressé binaire.....</u>	1490
<u>7.116.22 gzrewind() : Recommence la lecture du fichier.....</u>	1490
<u>7.116.23 gzseek() : Déplace le pointeur de lecture.....</u>	1491
<u>7.116.24 gztell() : Lit la position courante du pointeur de lecture.....</u>	1491
<u>7.116.25 gzuncompress() : Décompresse une chaîne compressée.....</u>	1491
<u>7.116.26 gzwrite() : Ecrit dans un fichier compressé, en gzip.....</u>	1491
<u>7.116.27 readgzfile() : Lit tout le fichier compressé.....</u>	1492
<u>7.116.28 zlib_get_coding_type() : Retourne la méthode de compression utilisée avec Gzip.....</u>	1492

8 Développer PHP 4.0.....1494

<u>8.1 Introduction.....</u>	1494
<u>8.1.1 Qu'est ce que Zend? et qu'est ce que PHP?.....</u>	1494
<u>8.2 Possibilités d'extension.....</u>	1495
<u>8.2.1 Modules externes.....</u>	1495
<u>8.2.2 Modules internes.....</u>	1496
<u>8.2.3 Zend Engine.....</u>	1496
<u>8.3 Source Layout.....</u>	1496
<u>8.3.1 Conventions sur les extensions.....</u>	1497
<u>8.3.2 Macros.....</u>	1497
<u>8.3.3 Gestion de la mémoire.....</u>	1497
<u>8.3.4 Fonctions de fichiers et dossiers.....</u>	1498
<u>8.3.5 Gestion des chaînes de caractères.....</u>	1498
<u>8.3.6 Types complexes.....</u>	1498
<u>8.4 Le système de squelette automatique de PHP.....</u>	1498
<u>8.5 Créer une extension.....</u>	1500
<u>8.5.2 Compiler le module.....</u>	1501
<u>8.6 Utiliser des extensions.....</u>	1502
<u>8.7 Résolution de problèmes.....</u>	1503
<u>8.8 Présentation des sources.....</u>	1504
<u>8.8.1 Structure de module.....</u>	1504
<u>8.8.2 Inclusions des fichiers d'entête.....</u>	1504
<u>8.8.3 Déclaration des fonctions exportées.....</u>	1504
<u>8.8.4 Déclaration du bloc de fonctions Zend.....</u>	1505
<u>8.8.5 Déclaration du bloc module de Zend.....</u>	1506
<u>8.8.6 Création de la fonction get_module.....</u>	1508
<u>8.8.7 Implémentation de toutes les fonctions exportées.....</u>	1508
<u>8.8.8 Sommaire.....</u>	1509
<u>8.9 Gestion des arguments.....</u>	1509
<u>8.9.1 Déterminer le nombre d'arguments.....</u>	1509
<u>8.9.2 Lire les arguments.....</u>	1510
<u>8.9.3 Ancien méthode de lecture des arguments (obsolète).....</u>	1512
<u>8.9.4 Traitement des nombres d'arguments variables et des paramètres.....</u>	

Sommaire

8 Développer PHP 4.0	
optionnels.....	1513
8.9.5 Accéder aux arguments.....	1514
8.9.6 Traiter les arguments passés par référence.....	1517
8.9.7 Assurer les autres paramètres contre l'écriture.....	1519
8.10 Créer des variables.....	1520
8.10.1 Présentation.....	1520
8.10.2 Entiers (Longs).....	1523
8.10.3 Les nombres à virgules flottantes (Doubles, Floats).....	1523
8.10.4 Chaînes de caractères (strings).....	1523
8.10.5 Booléens.....	1524
8.10.6 Tableaux.....	1524
8.10.7 Objets.....	1526
8.10.8 Ressources.....	1527
8.10.9 Macros de création automatique de variables globales.....	1530
8.10.10 Créations de constantes.....	1530
8.11 Duplique le contenu d'une variable : le bâtisseur de copie.....	1531
8.12 Retourner des valeurs.....	1532
8.13 Afficher des informations.....	1533
8.13.1 zend_printf.....	1533
8.13.2 zend_error.....	1533
8.13.3 Inclure du contenu dans phpinfo.....	1534
8.13.4 Informations d'exécution.....	1535
8.14 Fonctions de démarrage et d'extinction.....	1536
8.15 Appeler des fonctions utilisateurs.....	1536
8.16 Support du fichier d'initialisation File Support.....	1538
8.17 Par où continuer?.....	1540
8.18 Référence : quelques macros de configuration.....	1540
8.18.1 config.m4.....	1540
8.19 API Macros.....	1540
9 PHP API: Interface pour les extensions.....	1542
9.1 API de flôts les auteurs d'extensions.....	1542
9.1.1 Présentation.....	1542
9.1.2 Eléments de bases.....	1542
9.1.3 Les flôts qui sont des ressources.....	1543
9.1.4 Référence des API communes de flôts.....	1544
9.1.5 Streams Dir API Reference.....	1552
9.1.6 API des flôts pour les fichiers.....	1553
9.1.7 Streams Socket API Reference.....	1553
9.1.8 Streams Structures.....	1555
9.1.9 Constantes de flôts.....	1556
10 FAQ : Foire Aux Questions.....	1558
10.1 General Information.....	1558
10.1.1 What is PHP?.....	1558
10.1.2 What does PHP stand for?.....	1558
10.1.3 What is the relation between the versions?.....	1558
10.1.4 Can I run several versions of PHP at the same time?.....	1558
10.1.5 What are the differences between PHP 3 and PHP 4?.....	1558
10.1.6 I think I found a bug! Who should I tell?.....	1559
10.2 Mailing lists.....	1559

Sommaire

10 FAQ : Foire Aux Questions

<u>10.2.1 Are there any PHP mailing lists?</u>	1559
<u>10.2.2 Are there any other communities?</u>	1559
<u>10.2.3 Help! I can't seem to subscribe/unsubscribe to/from one of the mailing lists!</u>	1559
<u>10.2.4 Is there an archive of the mailing lists anywhere?</u>	1560
<u>10.2.5 What can I ask the mailing list?</u>	1560
<u>10.2.6 What information should I include when posting to the mailing list?</u>	1560
<u>10.3 Obtenir PHP</u>	1560
<u>10.3.1 Où puis-je obtenir PHP ?</u>	1561
<u>10.3.2 Est-ce que les versions binaires pré-compilées sont disponibles ?</u>	1561
<u>10.3.3 Où puis-je obtenir les librairies dont j'ai besoin pour compiler les extensions optionelles de PHP ?</u>	1561
<u>10.3.4 Comment faire fonctionner ces librairies ?</u>	1561
<u>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 sur mon Windows ?</u>	1562
<u>10.3.6 Où puis-je trouver le fichier Browser Capabilities ?</u>	1562
<u>10.4 Database issues</u>	1562
<u>10.4.1 I heard it's possible to access Microsoft SQL Server from PHP. How?</u>	1562
<u>10.4.2 Can I access Microsoft Access databases?</u>	1562
<u>10.4.3 I upgraded to PHP 4, and now mysql keeps telling me "Warning: MySQL: Unable to save result set in ...". What's up?</u>	1563
<u>10.4.4 PHP 5 no longer bundles MySQL client libraries, what does this mean to me? Can I still use MySQL with PHP? I try to use MySQL and get "function undefined" errors, what gives?</u>	1564
<u>10.4.5 After installing shared MySQL support, Apache dumps core as soon as libphp4.so is loaded. Can this be fixed?</u>	1564
<u>10.4.6 Why do I get an error that looks something like this: "Warning: 0 is not a MySQL result index in <file> on line <x>" or "Warning: Supplied argument is not a valid MySQL result resource in <file> on line <x>?"</u>	1564
<u>10.5 Installation</u>	1565
<u>10.5.1 Unix/Windows: Where should my</u>	1565
<u>10.5.2 Unix: I installed PHP, but every time I load a document, I get the message 'Document Contains No Data'! What's going on here?</u>	1565
<u>10.5.3 Unix: I installed PHP using RPMS, but Apache isn't processing the PHP pages! What's going on here?</u>	1566
<u>10.5.4 Unix: I installed PHP 3 using RPMS, but it doesn't compile with the database support I need! What's going on here?</u>	1566
<u>10.5.5 Unix: I patched Apache with the FrontPage extensions patch, and suddenly PHP stopped working. Is PHP incompatible with the Apache FrontPage extensions?</u>	1567
<u>10.5.6 Unix/Windows: I have installed PHP, but when I try to access a PHP script file via my browser, I get a blank screen.</u>	1567
<u>10.5.7 Unix/Windows: I have installed PHP, but when try to access a PHP script file via my browser, I get a server 500 error.</u>	1567
<u>10.5.8 Some operating systems: I have installed PHP without errors, but when I try to start apache I get undefined symbol errors:</u>	1568
<u>10.5.9 Windows: I have installed PHP, but when I to access a PHP script file via my browser, I get the error:</u>	1568
<u>10.5.10 Windows: I've followed all the instructions, but still can't get PHP and IIS to work together!</u>	1568
<u>10.5.11 When running PHP as CGI with IIS, PWS, OmniHTTPD or Xitami, I get</u>	

Sommaire

10 FAQ : Foire Aux Questions

<u>the following error:</u>	1568
<u>10.5.12 How do I know if my</u>	1569
<u>10.6 Build Problems</u>	1569
<u>10.6.1 I got the latest version of PHP using the anonymous CVS service, but there's no configure script!</u>	1569
<u>10.6.2 I'm having problems configuring PHP to work with Apache. It says it can't find</u>	1569
<u>10.6.3 While configuring PHP (</u>	1569
<u>10.6.4 When I try to start Apache, I get the the following message:</u>	1569
<u>10.6.5 When I run configure, it says that it can't find the include files or library for GD, gdbm, or some other package!</u>	1570
<u>10.6.6 When it is compiling the file</u>	1570
<u>10.6.7 When I run</u>	1570
<u>10.6.8 When linking PHP, it complains about a number of undefined references</u>	1570
<u>10.6.9 I can't figure out how to build PHP with Apache 1.3</u>	1570
<u>10.6.10 I have followed all the steps to install the Apache module version on UNIX, and my PHP scripts show up in my browser or I am being asked to save the file</u>	1571
<u>10.6.11 It says to use:</u>	1571
<u>10.6.12 When I try to build Apache with PHP as a static module using</u>	1571
<u>10.6.13 When I try to build PHP using</u>	1571
<u>10.6.14 During</u>	1572
<u>10.6.15 When compiling PHP with MySQL, configure runs fine but during</u>	1572
<u>10.6.16 I want to upgrade my PHP. Where can I find the</u>	1573
<u>10.6.17 When building PHP with the GD library it either gives strange compile errors or segfaults on execution</u>	1573
<u>10.6.18 When compiling PHP I seemingly get random errors, like it hangs. I'm using Solaris if that matters</u>	1573
<u>10.7 Utiliser PHP</u>	1573
<u>10.7.1 J'aimerais écrire un script</u>	1573
<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>	1574
<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>	1574
<u>10.7.4 Quand je fais ce qui suit, l'affichage se fait dans le mauvais ordre :</u>	1574
<u>10.7.5 Hey, où sont mes nouvelles lignes ?</u>	1575
<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>	1575
<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>	1575
<u>10.7.8 Quand j'essaye d'utiliser l'authentification avec IIS j'obtiens 'No Input file specified'</u>	1575
<u>10.7.9 Mon script</u>	1576
<u>10.7.10 Comment mélanger XML et</u>	1576
<u>10.7.11 Comment utiliser</u>	1576
<u>10.7.12 Où puis-je trouver une liste complète des variables prédéfinies que je peux utiliser dans mes scripts</u>	1576
<u>10.7.13 Comment puis-je générer des fichiers PDF sans utiliser les librairies payantes</u>	1576
<u>10.7.14 J'essaye d'accéder à une des variables standard CGI (comme</u>	

Sommaire

10 FAQ : Foire Aux Questions

<u>\$DOCUMENT_ROOT ou \$HTTP_REFERER) dans une fonction écrite par moi-même, et il semblerait qu'elle ne soit pas définie. Que se passe-t-il ?</u>	1577
<u>10.8 PHP and HTML</u>	1577
<u>10.8.1 What encoding/decoding do I need when I pass a value through a form/URL?</u>	1577
<u>10.8.2 I'm trying to use an <input type="image"> tag, but the</u>	1578
<u>10.8.3 How do I create arrays in a HTML <form>?</u>	1578
<u>10.8.4 How do I get all the results from a select multiple HTML tag?</u>	1579
<u>10.8.5 How can I pass a variable from Javascript to PHP?</u>	1580
<u>10.9 PHP and COM</u>	1580
<u>10.9.1 I have built a DLL to calculate something. Is there any way to run this DLL under PHP ?</u>	1580
<u>10.9.2 What does 'Unsupported variant type: xxxx (0xxxxx)' mean ?</u>	1580
<u>10.9.3 Is it possible manipulate visual objects in PHP ?</u>	1581
<u>10.9.4 Can I store a COM object in a session ?</u>	1581
<u>10.9.5 How can I trap COM errors ?</u>	1581
<u>10.9.6 Can I generate DLL files from PHP scripts like i can in Perl ?</u>	1581
<u>10.9.7 What does 'Unable to obtain IDispatch interface for CLSID {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}' mean ?</u>	1581
<u>10.9.8 How can I run COM object from remote server ?</u>	1581
<u>10.9.9 I get 'DCOM is disabled in C:\path...\scriptname.php on line 6', what can I do ?</u>	1581
<u>10.9.10 Is it possible to load/manipulate an ActiveX object in a page with PHP ?</u>	1582
<u>10.9.11 Is it possible to get a running instance of a component ?</u>	1582
<u>10.9.12 Is there a way to handle an event sent from COM object ?</u>	1582
<u>10.9.13 I'm having problems when trying to invoke a method of a COM object which exposes more than one interface. What can I do ?</u>	1582
<u>10.9.14 So PHP works with COM, how about COM+ ?</u>	1583
<u>10.9.15 If PHP can manipulate COM objects, can we imagine to use MTS to manage components resources, in conjunction with PHP ?</u>	1583
<u>10.10 PHP and other languages</u>	1583
<u>10.10.1 PHP vs. ASP?</u>	1583
<u>10.10.2 Is there an ASP to PHP converter?</u>	1583
<u>10.10.3 PHP vs. Cold Fusion?</u>	1583
<u>10.10.4 PHP vs. Perl?</u>	1584
<u>10.11 Migrer de PHP 2 à PHP 3</u>	1584
<u>10.11.1 Migrer de</u>	1584
<u>10.12 Migrer de PHP 3 à PHP 4</u>	1584
<u>10.12.1 Migrer de</u>	1584
<u>10.12.2 Les sessions fonctionnent-elles avec</u>	1584
<u>10.12.3 Fonctions incompatibles ?</u>	1584
<u>10.13 Questions diverses</u>	1585
<u>10.13.1 Où sont passés les pop-ups du site web ? Puis-je en voir le code ?</u>	1585
<u>10.13.2 Comment puis-je manipuler le manuel compressé en bzip2 sous Windows ?</u>	1585

11 Appendices.....1586

<u>11.1 Histoire de PHP</u>	1586
<u>11.1.1 Histoire de PHP</u>	1586
<u>11.1.2 Quelques projets liés à PHP</u>	1587
<u>11.1.3 Livres traitant de PHP</u>	1589

Sommaire

11 Appendices

<u>11.1.4 Publications sur PHP.....</u>	1589
<u>11.2 Migration de PHP 3.0 à PHP 4.0.....</u>	1589
<u>11.2.1 Ce qui a changé en PHP 4.0.....</u>	1589
<u>11.2.2 Utiliser PHP 3 et PHP 4 simultanément.....</u>	1589
<u>11.2.3 Migration des fichiers de configuration.....</u>	1590
<u>11.2.4 Comportement de l'analyseur.....</u>	1591
<u>11.2.5 Rapport d'erreur.....</u>	1591
<u>11.2.6 Initialiseur.....</u>	1593
<u>11.2.7 empty("0").....</u>	1593
<u>11.2.8 Fonctions manquantes.....</u>	1593
<u>11.2.9 Extensions PHP 3.0.....</u>	1594
<u>11.2.10 Substitution de variables dans les chaînes.....</u>	1594
<u>11.2.11 Cookies.....</u>	1594
<u>11.2.12 Gestion des variables globales.....</u>	1594
<u>11.3 Migration de PHP/FI 2.0 à PHP 3.0.....</u>	1595
<u>11.3.1 A propos des incompatibilités en 3.0.....</u>	1595
<u>11.3.2 old_function.....</u>	1595
<u>11.3.3 Balises PHP.....</u>	1595
<u>11.3.4 Syntaxe if..endif.....</u>	1596
<u>11.3.5 Syntaxe while.....</u>	1597
<u>11.3.6 Types d'expression.....</u>	1597
<u>11.3.7 Les messages d'erreur ont changé.....</u>	1598
<u>11.3.8 Evaluation rapide des booléens.....</u>	1598
<u>11.3.9 La valeur TRUE / FALSE comme retour de fonctions.....</u>	1598
<u>11.3.10 Diverses incompatibilités.....</u>	1598
<u>11.4 Débugueur PHP.....</u>	1599
<u>11.4.1 A propos du débbugueur.....</u>	1599
<u>11.4.2 Utiliser le débbugueur PHP.....</u>	1599
<u>11.4.3 Protocole du débbugueur.....</u>	1600
<u>11.5 Développement PHP.....</u>	1601
<u>11.5.1 Créer une fonction PHP 3.....</u>	1601
<u>11.5.2 Appeler des fonctions utilisateurs.....</u>	1608
<u>11.5.3 Rapport d'erreurs.....</u>	1609
<u>11.6 Liste d'alias.....</u>	1611
<u>11.7 Mots réservés en PHP.....</u>	1615
<u>11.7.1 Liste de mots-clé.....</u>	1616
<u>11.7.2 Variables prédéfinies.....</u>	1616
<u>11.7.3 Classes prédéfinies.....</u>	1621
<u>11.7.4 Constantes prédéfinies en PHP.....</u>	1622
<u>11.8 Types des ressources PHP.....</u>	1627
<u>11.9 Liste des protocoles supportés.....</u>	1632
<u>11.9.1 HTTP et HTTPS.....</u>	1632
<u>11.9.2 FTP and FTPS.....</u>	1633
<u>11.9.3 Flôts d'entrée/sortie.....</u>	1633
<u>11.9.4 Flôts compressés.....</u>	1634
<u>11.10 Liste des modes de transports de sockets disponibles.....</u>	1634
<u>11.10.1 Domaién Internet : TCP, UDP, SSL et TLS.....</u>	1634
<u>11.10.2 Domaine Unix : UNIX et UDG.....</u>	1635
<u>11.11 Comparaison de types en PHP.....</u>	1636
<u>11.12 Liste des token de l'analyseur.....</u>	1637
<u>11.13 A propos du manuel.....</u>	1638

Sommaire

11 Appendices

<u>11.13.1 Formats</u>	1638
<u>11.13.2 A propos des notes utilisateurs</u>	1639
<u>11.13.3 Comment lire la définition d'une fonction (prototype)</u>	1639
<u>11.13.4 Versions de PHP documentées dans ce manuel</u>	1640
<u>11.13.5 Où trouver plus d'informations sur PHP ?</u>	1641
<u>11.13.6 Comment aider à l'amélioration de la documentation</u>	1641
<u>11.13.7 Comment sont générées les documentations</u>	1641
<u>11.13.8 Traductions</u>	1642
<u>11.13.9 Traduction française</u>	1642

12 Index.....**1643**

<u>12.1 Index des fonctions</u>	1643
<u>12.2 Index des concepts</u>	1701
<u>12.3 Index des exemples</u>	1760

Manuel PHP

1.1 Les auteurs

- Stig Sæther Bakken
- Alexander Aulbach
- Egon Schmid
- Jim Winstead
- Lars Torben Wilson
- Rasmus Lerdorf
- Andrei Zmievski
- Jouni Ahto

Date de génération : 16/10/2003 08:10:44

1.2 L'éditeur

- Damien Seguy
- Mehdi Achour

©

1.3 Note légale

Copyright

Ce manuel est © Copyright 1997 – 2003 par PHP Documentation Group. Les membres de ce groupe sont listés sur la première page de ce manuel.

Ce manuel peut être redistribué sous licence GNU General Public License, comme stipulé par la Free Software Foundation; soit la version 2 de la Licence, soit (à votre choix), une version ultérieure.

Les membres du groupe de documentation sont listés au début de ce manuel. Si vous souhaitez des contacts avec les membres du groupe, écrivez à phpdoc@lists.php.net.

Les membres de la traduction française sont listés au début de ce manuel. Si vous souhaitez des contacts avec les membres du groupe, écrivez à doc-fr@lists.php.net.

La section 'Développer PHP 4.0' de ce manuel est copyrighté © 2000 par Zend Technologies, Ltd. Ce document ne peut pas être redistribué sans satisfaire les termes et conditions de la licence Open Publication License, v1.0 ou plus récente (la dernière version est actuellement disponible à <http://www.opencontent.org/openpub/>).

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 dans divers formats, sur le site officiel de <http://www.php.net/docs.php> , et sur le site de <http://www.nexen.net/> . Les fichiers en téléchargement sont mis à jour au fur et à mesure des modifications. Plus d'informations sur ce manuel sont disponibles dans la section A propos du manuel .

Voir aussi PHP History .

3 Comment 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 le 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 suivante : Que peut faire PHP? .

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. 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 graphiques. ^{PHP} n'est probablement pas le meilleur langage pour écrire des applications 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, de nombreuses librairies et applications d'envergure (y compris la bibliothèque PEAR) sont écrites en langage objet.

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 librairies 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
- 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 par exemple 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 régulières POSIX étendues ou Perl aux traitements des fichiers XML, avec les standards SAX et DOM. Vous pouvez utiliser les transformations XSLT.

En commerce électronique, vous trouverez des outils de paiement intégrés comme Cybercash, CyberMut, VeriSign Payflow Pro et CCVS, 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.

Supposons que vous souhaitiez économiser du temps en ligne et travailler localement. Dans ce cas, vous devez installer un serveur web comme Apache, et bien sur PHP. Vous souhaiterez aussi installer une base de données comme par exemple MySQL. Vous pouvez installer ces logiciels individuellement, ou bien avec des installateurs 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 Linux et Windows. Sous Linux, vous pouvez aussi trouver des commandes comme rpmfind, très pratique pour rechercher les paquets pré compilés.

3.2.2 Votre première page PHP

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

Notre premier script PHP : `bonjour.php`

```
<html>
<head>
  <title>Test PHP</title>
</head>
<body>
  <?php echo "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. Bien que cela soit hors du contexte de ce tutoriel, étudiez `DocumentRoot` et `ServerName` dans le fichier de configuration de votre serveur web (sous Apache, c'est `httpd.conf`). 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>
  Bonjour le monde<p>
</body>
</html>
```

Notez que cela n'est pas un script CGI. Ce fichier n'a nul besoin d'exécutable ou de configuration particulière. C'est vraiment comme un fichier HTML classique, qui comporte des balises spéciales, capables de réaliser des tâches particulières.

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.

Si vous avez essayé cet exemple, et qu'il n'a rien affiché de spécial, ou même qu'un 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 le PHP. Demandez à votre administrateur de l'activer pour vous, en utilisant le chapitre [Installation](#) . Si vous voulez développer des scripts PHP localement, voyez la section de [téléchargements](#) . Vous pouvez développer en PHP, localement, sur tous les systèmes d'exploitation. N'oubliez pas d'installer aussi un serveur web approprié. Si le problème persiste, n'hésitez pas à utiliser l'une des options de [support](#) .

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.

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, rendez donc une visite à cette page, et demandez au webmestre d'ajouter votre éditeur.

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 sauvs 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 guillemets (i.e. "hello.php").

Ou bien, vous pouvez 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 guillemets.

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

Attention pour les auto-globales de PHP

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

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 5.01; Windows NT 5.0)

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. Vous pouvez aussi en obtenir une liste complète en créant un fichier comme ceci :

Afficher toutes les variables prédéfinies phpinfo

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

Lorsque vous chargez cette page dans votre navigateur, vous allez découvrir une page pleine d'informations sur PHP, ainsi qu'une liste des variables qui sont disponibles.

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 (strstr($_SERVER["HTTP_USER_AGENT"], "MSIE")) {
    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 strstr. strstr 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 `TRUE` et sinon, `FALSE`. Si elle retourne `TRUE`, la commande if reçoit `TRUE` et le code entre accolades `{ }` est 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.

Nous pouvons aussi 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 (strstr($_SERVER["HTTP_USER_AGENT"], "MSIE")) {
?>
<h3>strstr a retourné TRUE</h3>
<center><b>Vous utilisez Internet Explorer</b></center>
<?php
} else {
?>
<h3>strstr a retourné FALSE</h3>
<center><b>Vous n'utilisez pas Internet Explorer</b></center>
<?php
}
?>
```

Un exemple de résultat obtenu dans ce script est :

```
<h3>strstr a retourné TRUE</h3>
<center><b>Vous utilisez Internet Explorer</b></center>
```

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 que le navigateur utilise ou pas Internet Explorer.

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 :

```
U*n simple formulaire HTML

<form action="action.php" method="POST">
Votre nom : <input type="text" name="nom" />
Votre age : <input type="text" name="age" />
<input type="submit" value="OK">
</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 d'envoi, 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 tutorial, 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 avions 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` .

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 variable 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 `$id` , si l'URL `http://www.exemple.com/foo.php?id=42` a été appelée. Quelle que soit la valeur de la directive, `$ _GET['id']` est toujours disponible.

Pour plus de détails sur ces modifications, reportez-vous à variables prédéfinies .

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'exemple 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/> ou <http://talks.php.net/> .

3.3 Installation

3.3.1 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? section:

- 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/> . Nous recommandons l'utilisation du [miroir](#) le plus proche pour accélérer les chargements.

3.3.2 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.

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 (pour compiler)
- bison (pour compiler)
- Un serveur web
- Tous les composants nécessaires aux extensions (bibliothèque ^{GD} , ^{PDF} , etc...)

Il y a plusieurs façons d'installer PHP sur une plate-forme UNIX : soit un processus de compilation–configuration, ou bien avec des packages déjà tout prêts. Cette documentation se concentre sur la première solution.

La première partie du processus est faite en ligne de commande, grâce aux options du script `configure`. Cette section présente l'utilisation des options les plus courantes, mais il y en a beaucoup d'autres à essayer. Reportez-vous à la [liste complète des options de configuration](#) pour une liste exhaustive. Voici les différentes méthodes d'installation de PHP :

- Comme module Apache
- Comme module fhttpd
- Pour l'utiliser avec AOLServer, NSAPI, phttpd, Pi3Web, Roxen, thttpd, ou Zeus.
- Comme exécutable CGI

3.3.2.1 Installer PHP comme module Apache

PHP peut être compilé de nombreuses manières différentes, mais la plus populaire est le module Apache. La liste suivante est un récapitulatif de l'installation.

Installer PHP comme module Apache

```
1. gunzip apache_1.3.x.tar.gz
2. tar xvf apache_1.3.x.tar
3. gunzip php-x.x.x.tar.gz
4. tar xvf php-x.x.x.tar
5. cd apache_1.3.x
6. ./configure --prefix=/www
7. cd ../php-x.x.x
8. ./configure --with-mysql --with-apache=../apache_1.3.x --enable-track-vars
9. make
10. make install
11. cd ../apache_1.3.x
12. ./configure --activate-module=src/modules/php4/libphp4.a
13. make
14. make install
15. cd ../php-x.x.x
16. cp php.ini-dist /usr/local/lib/php.ini
17. "Editez votre fichier httpd.conf ou srm.conf et ajoutez : "
    AddType application/x-httpd-php .php
18. "Utilisez votre procédure habituelle pour redémarrer le serveur Apache.
    (vous devez arrêter puis redémarrer le serveur, et pas seulement forcer
    le serveur à relire la configuration initiale).
```

3.3.2.2 Compilation

Lorsque PHP est configuré, vous êtes prêts à compiler l'exécutable CGI. La commande `make` doit prendre tout en charge. Si ce n'est pas le cas et que vous restez bloqués, reportez-vous aux [problèmes courants](#).

3.3.3 Installation sous Linux/Unix

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

3.3.3.1 Utilisation des packages

De nombreuses distributions Linux disposent d'un système d'installation par package, comme le fameux RPM. Ils vous permettent de faire des installations standard, mais si vous avez des configurations spécifiques (comme par exemple un serveur sécurisé, ou un pilote de bases de données exotique), vous aurez probablement à compiler vous-même votre PHP et votre serveur

web. Si vous n'êtes pas familier avec la compilation de vos propres logiciels, il vaut mieux rechercher le package qui pourra répondre à vos besoins.

3.3.4 Installation sous HP-UX

Cette section contient les notes et conseils d'installation de PHP sur les distributions HP-UX. (Fournis 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 rusées, mais le plus simple et le plus efficace est d'ajouter `"lt_target=hpux10.20"` à la ligne 47286 du script de configuration.
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 de shared object 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.

3.3.5 Installation sous Solaris

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

3.3.5.1 Logiciels nécessaires

L'installation Solaris oublie généralement les compilateurs C, et leurs utilitaires. Voici la liste des outils nécessaires :

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

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

3.3.5.2 Utilisation des packages

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

3.3.6 Installations Unix/OpenBSD

Cette section contient les notes spécifiques à l'installation de PHP sous OpenBSD 3.2 .

3.3.6.1 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 package principal que vous devez installer est `php4-core-4.2.3.tgz` , qui contient le moteur de base, plus gettext et iconv). Puis, jetez un oeil aux packages de module, comme `php4-mysql-4.2.3.tgz` ou `php4-imap-4.2.3.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-4.2.3.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-4.2.3.tgz
# /usr/local/sbin/phpxs -a mysql
# (add in imap)
# pkg_add php4-imap-4.2.3.tgz
```

```
# /usr/local/sbin/phpxs -a imap
# (remove mysql as a test)
# pkg_delete php4-mysql-4.2.3
# /usr/local/sbin/phpxs -r mysql
# (install the PEAR libraries)
# pkg_add php4-pear-4.2.3.tgz
```

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

3.3.6.2 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 trois sous-dossiers : core, extensions et pear. 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}.

3.3.6.3 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](#).

3.3.7 Installation sous Unix et Mac OS X

Cette section contient les notes et conseils pour installer ^{PHP} sur un serveur Mac OS X Server.

3.3.7.1 Utilisation des paquets

Il existe quelques versions pré-packagées et pré-compilées de ^{PHP} pour MacOSX. 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 familiers avec la compilation et la mise en place d'applications, il est bon de vérifier si personne d'autre n'a réalisé un package.

3.3.7.2 Compilation pour OS X server

Il existe deux versions légèrement différentes de Mac OS X, le client et le serveur. Les lignes ci-dessous sont pour OS X Server.

Installation sur Mac OS X server

```
1. Get the latest distributions of Apache and PHP
2. Untar them, and run the configure program on Apache like so.
./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

4. You may also want to add this line:
setenv OPTIM=-O2
If you want the compiler to do some optimization.

5. Next, go to the PHP 4 source directory and configure it.
./configure --prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--with-xml \
--with-apache=/src/apache_1.3.12

If you have any other additions (MySQL, GD, etc.), be sure to add
them here. For the --with-apache string, put in the path to your
apache source directory, for example "/src/apache_1.3.12".
6. make
7. make install
This will add a directory to your Apache source directory under
src/modules/php4.

8. Now, reconfigure Apache to build in 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

You may get a message telling you that libmodphp4.a is out of date.
If so, go to the src/modules/php4 directory inside your apache
source directory and run this command:

ranlib libmodphp4.a

Then go back to the root of the apache source directory and run the
above configure command again. That'll bring the link table up to
date.

9. make

10. make install

11. copy and rename the php.ini-dist file to your "bin" directory from your
PHP 4 source directory:
cp php.ini-dist /usr/local/bin/php.ini

or (if your don't have a local directory)

cp php.ini-dist /usr/bin/php.ini
```

3.3.7.3 Compilation pour MacOS X client

Ces conseils sont fournis gracieusement par [Marc Liyanage](#) .

Le module ^{PHP} d'Apache est inclus dans l'installation de Mac OS X. Cette version inclut le support des bases de données MySQL et PostgreSQL.

NOTE: Attention lors des manipulations ci-dessous : vous pourriez détruire la configuration de votre serveur.

Faites ceci :

- 1. Ouvrez une fenêtre de terminal
- 2. Tapez "wget http://www.diax.ch/users/liyanage/software/macosx/libphp4.so.gz", et attendez le téléchargement.
- 3. Tapez "gunzip libphp4.so.gz"
- 4. Tapez "sudo apxs -i -a -n php4 libphp4.so"

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.

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 ci-contre : " <?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}.

3.3.8 Liste des options de configuration diverses

Vous trouverez ci-dessus une liste partielle des options de configuration utilisées par le script de configuration ^{PHP configure}, lors de la compilation sur les environnements de type Unix. La plupart des options de configuration sont décrites dans leur chapitre et ne sont pas présentées ici. Pour une liste complète des options de configuration, utilisez la commande en ligne ^{configure} --help dans votre dossier ^{PHP} après avoir exécuté la commande ^{autoconf} (voyez aussi le [chapitre d'installation](#)). Vous serez aussi intéressés par la lecture de la documentation de [GNU configure](#) pour plus de détails sur les options du script ^{configure}, telles que ^{--prefix=PREFIX}.

Note

Ces options sont utilisées uniquement au moment de la compilation. Si vous voulez modifier le comportement de PHP lors de l'exécution, reportez vous au chapitre [configuration](#).

Ci-dessous, vous trouverez une liste complète des options supportées par le script de configuration de PHP 4, ^{configure} (depuis PHP 4.1.0), utilisé lors de la compilation dans les environnements Unix. Certaines options sont disponibles en PHP 3, d'autres en PHP 4, d'autres encore dans les deux.

Il y a des options de configuration générales pour le script ^{configure}, consultez les pages appropriées du manuel de GNU ^{autoconf} ou bien utilisez la commande ^{configure} --help pour avoir une liste complète et à jour des options.

- [Bases de données](#)
- [Graphisme](#)
- [Divers](#)
- [PHP](#)
- [Serveur](#)

3.3.8.1 Options de configuration en PHP 4

Note

Ces options ne sont utilisées qu'en PHP 4, depuis la version PHP 4.1.0. Certaines sont disponibles dans les anciennes versions PHP 4, certaines même en PHP 3, certaines uniquement en PHP 4.1.0. Si vous essayez de compiler une version plus ancienne, certaines options peuvent simplement ne pas exister.

3.3.8.1.1 Options graphiques

`--with-imagick`

L'extension imagick a été placée dans PECL de PEAR et peut être trouvée [ici](#) . Les instructions d'installation pour ^{PHP} 4 sont disponibles sur le site de PEAR.

En bref, `--with-imagick` est uniquement supporté en ^{PHP} 3, à moins que vous ne suiviez les instructions disponibles sur le site de PEAR.

3.3.8.1.2 Options diverses

`--enable-debug`

Compile PHP avec les symboles de débogage.

`--with-layout=type`

Indique comment les fichiers installés doivent être rangés. Les valeurs possibles sont Type (valeur par défaut) et GNU.

`--with-pear=dir`

Installe PEAR dans le dossier DIR. Par défaut, c'est PREFIX/lib/php.

`--without-pear`

Exclut l'installation de PEAR.

`--enable-sigchild`

Active le gestionnaire de SIGCHLD propre à PHP.

`--disable-rpath`

Désactive le passage de chemins supplémentaires pour chercher les librairies additionnelles.

`--enable-libgcc`

Active explicitement le linkage avec libgcc.

`--enable-php-streams`

Inclut le support des flux de PHP (expérimental). Ne pas utiliser à moins que vous ne testiez le code.

`--with-zlib-dir=<dir>`

Définit le chemin d'installation de zlib.

`--enable-trans-sid`

Active la propagation transparente des sessions.

`--with-tsrm-pthreads`

Utilise les threads POSIX (configuré par défaut).

`--enable-shared[=pkgs]`

Compile les librairies partagées. Par défaut, c'est yes.

`--enable-static[=pkgs]`

Compile les librairies statiques. Par défaut, c'est yes.

`--enable-fast-install[=pkgs]`

Optimise pour l'installation rapide. Par défaut, c'est yes.

`--with-gnu-ld`

Suppose que le compilateur utilise GNU ld. Par défaut, c'est no.

`--disable-libtool-lock`

Evite les verrouillages (mais cela risque de corrompre des compilations parallèles).

`--with-pic`

Essaie d'utiliser uniquement des objets PIC/non-PIC. Par défaut, c'est ^{both} (les deux).

`--enable-memory-limit`

Compile PHP avec le support de la limitation de mémoire.

`--disable-url-fopen-wrapper`

Désactive les gestionnaires HTTP et FTP avec les fonctions d'accès aux fichiers.

`--enable-versioning`

Exporte uniquement les symboles nécessaires. Voir le fichier INSTALL pour plus de détails.

`--with-impl[=dir]`

Inclut le support de IMSP. DIR est le dossier d'IMSP qui contient les fichiers d'en-têtes et le fichier libimsp.a. Valable uniquement pour PHP 3!

`--with-mck[=dir]`

Inclut le support de Cybercash MCK. DIR est le dossier d'installation de cybercash mck. Par défaut, c'est /usr/src/mck-3.2.0.3-linux. Pour de l'aide, regardez dans extra/cyberlib. Valable uniquement pour PHP 3!

`--with-mod-dav=dir`

Inclut le support de DAV, grâce à l'interface Apache mod_dav. DIR est le dossier d'installation de mod_dav. Fonctionne uniquement pour la version module d'Apache.

Valable uniquement pour PHP 3!

--enable-debugger

Compile PHP avec les fonctionnalités de débogage à distance. Valable uniquement pour PHP 3!

--enable-versioning

Profite du système de versionnage et de gestion d'environnement de Solaris 2.x et Linux. Valable uniquement pour PHP 3!

3.3.8.1.3 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 par défaut.

--with-exec-dir[=dir]

Autorise uniquement les exécutables situés dans le dossier DIR lorsque le safe mode est activé. Par défaut, c'est DIR vaut `/usr/local/php/bin`.

--enable-magic-quotes

Active les guillemets magiques.

--disable-short-tags

Désactive les balises courtes `<? .`

3.3.8.1.4 Options Serveur

--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, c'est `/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

Empêche la compilation de la version CLI de ^{PHP} (cela force aussi l'option `--without-pear`). Disponible depuis PHP 4.3.0.

--enable-embed[=type]

Active la compilation de la librairie 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, c'est `/usr/local/src/fhttpd`.

--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 Netscape Server.

--with-phhttpd=dir

Aucune information fournie.

--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, c'est /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 librairie 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.

--with-fastcgi

Compile ^{PHP} comme application FastCGI. Cette option n'est plus disponible depuis ^{PHP} 4.3.0.

--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.

3.3.9 Installation sous Windows

Cette section s'applique aux systèmes Windows 98/Me et Windows NT/2000/XP. Ne vous attendez pas à ce que ^{PHP} fonctionne sur les plateformes 16 bits, comme Windows 3.1. Parfois, on décrit les plateformes 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 InstallShield.

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.

3.3.9.1 InstallShield sous 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.

Notez bien que bien que InstallShield soit une méthode simple pour installer ^{PHP}, il est limité dans son fonctionnement, puisque l'installation automatique des extensions n'est pas prise en compte. Le jeu complet des extensions ^{PHP} est disponible en téléchargeant et décompressant manuellement l'archive.

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} et le serveur web qui utilisera ^{PHP}. Pour IIS, mais aussi PWS sous NT Workstation, une liste de l'arborescence web est affichée, et vous pouvez sélectionner les dossiers qui utiliseront ^{PHP}.

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.

3.3.9.2 Installation manuelle sous Windows

Ce guide d'installation vous aide à installer manuellement et configurer ^{PHP} sur votre serveur web Windows. La version initiale de ce guide a été compilée par Bob Silva, et est disponible en anglais, à <http://www.umesd.k12.or.us/php/win32install.html>. Vous devrez télécharger la distribution binaire Zip sur la page <http://www.php.net/downloads.php>.

^{PHP} 4 pour Windows est distribué en trois versions : exécutable ^{CGI}, un exécutable CLI (sapi/php.exe) et d'autres modules SAPI :

- php4apache.dll – module Apache 1.3.x.
- php4apache2.dll – module Apache 2.0.x.
- php4isapi.dll – module ISAPI compatible avec les serveurs comme IIS 4.0/PWS 4.0 ou plus récents.
- php4nsapi.dll – module Netscape/iPlanet.

Cette dernière forme est nouvelle en ^{PHP} 4, et fournit des performances améliorées et de nouvelles fonctionnalités. La version CLI est conçue pour utiliser ^{PHP} en ligne de commande. Plus d'informations sur la forme CLI est disponible dans le chapitre "utiliser ^{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 autres systèmes.

Note

Pré-requis DCOM et MDAC

Si vous choisissez d'installer le module SAPI sur **Windows 95**, assurez vous de télécharger et installer la mise à jour DCOM, disponible à [Microsoft DCOM pages](#). 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://www.microsoft.com/data/>.

Voici les différentes étapes d'installation avant les étapes spécifiques au serveur.

- Extrayez la distribution dans un dossier de votre choix, comme par exemple ^{c:\}. L'archive zip devient un dossier appelé ^{php-4.3.1-win32} qui est supposé être renommé en ^{php}. Par souci de clarté, nous supposons alors que vous avez placé votre distribution dans le dossier

c:\php . Vous pouvez choisir un autre dossier, mais vous serez prudents d'éviter les espaces dans le nom du chemin au dossier (comme par exemple c:\program files\php n'est pas une bonne idée). car certains serveurs web vont crasher sinon. La structure du dossier que vous avez extrait ressemble à ceci :

```
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
| |
| |--php4isapi.dll
| |
| |--..
|
|--install.txt
|
|--..
|
|--php.exe           -- executable CGI
|
|--..
|
|--php.ini-dist
|
|--php.ini-recommended
|
|--php4ts.dll        -- DLL principale --> dossier systeme Windows
|
|--...
```

Le binaire CGI (c:\php\php.exe), le binaire CLI (c:\php\cli\php.exe), et les modules SAPI (c:\php\sapi*.dll) dépendent de la DLL principale c:\php\php4ts.dll . Vous devez vous assurer que cette DLL est disponible pour votre installation ^{PHP} . L'ordre des dossiers lors de la recherche est le suivant :

- Le dossier d'où ^{php.exe} est appelé. Lorsque vous utilisez le module SAPI, c'est le dossier d'où le serveur web charge la DLL (i.e. ^{php4apache.dll}).
- Tout dossier dans la variable d'environnement Windows ^{PATH} .
- La meilleure chose à faire est de rendre ^{php4ts.dll} accessible, indépendamment de l'interface utilisée (CGI ou module SAPI). Pour cela, vous devez copier la DLL dans un dossier de votre PATH Windows. La meilleure place est alors le dossier système Windows :
 - ◆ c:\windows\system pour Windows 9x/ME
 - ◆ c:\winnt\system32 pour Windows NT/2000 ou c:\winnt40\system32 pour NT/2000 server
 - ◆ c:\windows\system32 pour Windows XP

Si vous voulez utiliser le module SAPI depuis `c:\php\sapi` et que vous ne voulez pas copier vos DLL dans le dossier système Windows, vous avez la possibilité de copier simplement `php4ts.dll` dans le dossier SAPI de votre archive décompressée `c:\php\sapi`.

- L'étape suivante est de faire un fichier de configuration valide pour PHP, c'est à dire le fichier `php.ini`. Il y a deux versions de ce fichier disponibles dans l'archive : `php.ini-dist` et `php.ini-recommended`. Nous vous recommandons d'utiliser `php.ini-recommended`, car nous avons optimisé les configurations par défaut dans ce fichier pour les performances et la sécurité. Lisez bien ce fichier documenté, et étudiez toutes les configurations, puis faites vos propres choix. Si vous voulez atteindre la meilleure sécurité, c'est la meilleure méthode. Copiez votre fichier de configuration de choix dans un dossier où PHP est capable de le lire, et renommez le `php.ini`. Par défaut, PHP recherche `php.ini` dans votre dossier Windows :
 - ♦ Sur Windows 9x/ME/XP, copiez le fichier `php.ini` dans votre dossier `%WINDIR%`, qui est généralement `c:\windows`.
 - ♦ Sur Windows NT/2000, copiez le fichier `php.ini` dans votre dossier `%WINDIR%` ou `%SYSTEMROOT%`, qui est typiquement `c:\winnt` ou `c:\winnt40` pour les serveurs NT/2000.
- Si vous utilisez NTFS sur Windows NT, 2000 ou XP, assurez vous que l'utilisateur qui fait tourner le serveur web a les droits de lecture de votre fichier `php.ini` (e.g. rendez le lisible par tout le monde).

Les étapes suivantes sont optionnelles :

- Editez votre nouveau fichier `php.ini`. Si vous envisagez d'utiliser OmniHTTPd, ignorez l'étape suivante. Configurez `doc_root` pour qu'il pointe sur la racine de votre serveur web. Par exemple :

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

- Choisissez les extensions que vous voulez que PHP démarre. Voyez la section sur les extensions Windows, pour les configurer, si elles ne sont pas intégrées. Notez que sur une nouvelle installation, il est recommandé de commencer par faire fonctionner PHP avant d'ajouter ou retirer des extensions via le `php.ini`.
- Sur PWS et IIS, vous pouvez utiliser la directive browscap pour qu'elle pointe sur :
`c:\windows\system\inetrv\browscap.ini` sur Windows 9x/Me, `c:\winnt\system32\inetrv\browscap.ini` sur NT/2000, et `c:\windows\system32\inetrv\browscap.ini` sur XP.

Copiez le fichier `php.ini-dist` dans votre dossier `%WINDIR%` sous Windows 95/98, ou vers votre dossier `%SYSTEMROOT%` sous Windows NT ou Windows 2000 et renommez-le en `php.ini`. Votre dossier `%WINDIR%` ou `%SYSTEMROOT%` est généralement :

- `c:\windows` pour Windows 9x/Me
- `c:\winnt` ou `c:\winnt40` pour les serveurs NT/2000

Il y a deux fichiers ini, distribués dans le fichier zip, `php.ini-dist` et `php.ini-optimized`. Nous vous recommandons d'utiliser `php.ini-optimized`, car nous l'avons optimisé d'un point de vue sécurité et vitesse d'exécution. Le mieux est tout de même de lire le fichier de configuration `php.ini` et de choisir vos options manuellement. Si vous voulez atteindre un meilleur niveau de sécurité, c'est par ce moyen que vous pourrez le faire, même si PHP fonctionne parfaitement avec les configurations par défaut.

Editez votre fichier `php.ini` :

- la famille des serveurs Windows, Personal Web server (PWS) 3 et 4 ou plus récents; Internet Information Server (IIS) 3 et 4 ou plus récents.
- les serveurs Apache Apache 1.3.x, et Apache 2.x.

- les serveurs Netscape/iPlanet .
- le serveur OmniHTTPd .
- le serveur Oreilly Website Pro .
- le serveur Sambar .
- le serveur Xitami .

3.3.9.3 Compilation des sources

Avant de commencer, il est bon de se poser la question suivante : "Pourquoi la compilation de ^{PHP} sous Windows est si difficile?". Deux raisons viennent immédiatement à l'esprit :

1. Windows ne dispose pas (encore) d'une grande communauté de développeurs qui partagent librement leurs sources. La conséquence directe est que les investissements nécessaires en infrastructure pour supporter ce type de développement n'ont pas été faits. Ce qui fait que le portage des utilitaires Unix a été la solution pour pallier ce manque. Ne soyez donc pas surpris de rencontrer cette parenté de temps en temps.
2. La majorité des instructions que vous allez rencontrer sont du type : "faire et oublier". Alors, asseyez-vous confortablement et suivez aussi scrupuleusement que possible les instructions.

3.3.9.3.1 Pré-requis

Pour compiler l'exécutable ^{PHP}, vous avez besoin d'un Microsoft Development Environment. Microsoft Visual C++ 6.0 est recommandé. Pour extraire le fichier téléchargé, vous avez besoin d'un outil de décompression (e.g.: Winzip). Si vous n'avez pas encore d'utilitaire de décompression, vous pouvez en télécharger un chez InfoZip .

Avant de commencer, vous devez télécharger...

- les win32 buildtools de ^{PHP}, sur <http://www.php.net/extra/win32build.zip> .
- le code source du résolveur de noms DNS utilisé par ^{PHP} sur http://www.php.net/extra/bindlib_w32.zip . C'est un remplacement pour la librairie ^{resolv.lib} incluse dans ^{win32build.zip} .
- Si vous prévoyez de compiler ^{PHP} comme module pour Apache, vous aurez aussi besoin des sources d'Apache .

Finalement, vous aurez besoin des sources ^{PHP} 4 elles-mêmes. Les dernières versions sont accessibles sur le serveur CVS anonyme . Vous pouvez aussi télécharger une version intermédiaire ou l'archive des sources la plus récente.

3.3.9.3.2 Compiler tout ensemble

Après téléchargement des paquets nécessaires, vous devez les extraire au bon endroit.

- Créez un dossier où tous les fichiers seront placés après l'extraction, comme par exemple : ^{c:\work} .
- Créez le dossier ^{win32build} dans votre dossier de travail (ici ^{c:\work}) et décompressez ^{win32build.zip} dedans.
- Créez le dossier ^{bindlib_w32} sous votre dossier de travail (ici ^{c:\work}) et décompressez ^{bindlib_w32.zip} dedans.
- Décompressez le code source de ^{PHP} dans votre dossier de travail (^{c:\work}).

Après avoir suivi ces instructions, vous devriez obtenir un dossier qui ressemble à ceci :

```

+-c:\work
|
| +-bindlib_w32
| |
| | +-arpa
| |
| | +-conf
| |
| | +-...
|
| +-php-4.x.x
| |
| | +-build
| |
| | +-...
| |
| | +-win32
| |
| | +-...
|
| +-win32build
| |
| | +-bin
| |
| | +-include
| |
| | +-lib

```

Créez les dossiers `c:\usr\local\lib` . Copiez le fichier `bison.simple` depuis `c:\work\win32build\bin` vers `c:\usr\local\lib` .

Note

Les utilisateurs de Cygwin peuvent omettre la dernière étape. Un environnement Cygwin correctement installé fournit les fichiers obligatoires de `bison.simple` et `bison.exe` .

3.3.9.3.3 Configuration de MVC ++

Lancez Microsoft Visual C++, et allez dans le menu "select Tools => Options". Dans le dialogue, sélectionnez l'onglet "directories". Modifiez les valeurs pour les items du menu Executables, Includes et Library files. Votre configuration doit alors ressembler à

- Fichiers exécutables : `c:\work\win32build\bin` , Utilisateurs Cygwin : `cygwin\bin`
- Fichiers include: `c:\work\win32build\include`
- Fichiers library : `c:\work\win32build\lib`

3.3.9.3.4 Compilation de resolv.lib

Vous devez compiler la librairie `resolv.lib` . Decidez si vous avez besoin des symboles de débogage (bindlib – Win32 Debug) ou non (bindlib – Win32 Release). Compilez la configuration ad hoc :

- Pour les utilisateurs de l'interface graphique, lancez VC++, puis sélectionnez le menu "File => Open Workspace" et "bindlib". Puis sélectionnez "Build=>Set Active Configuration" et sélectionnez la configuration voulue. Enfin, sélectionnez "Build=>Rebuild All".
- Pour les utilisateurs en ligne de commande, assurez-vous que vous avez enregistré les variables d'environnement C++, ou que vous avez exécuté `vcvars.bat` . Exécutez maintenant l'une des commandes suivantes :

```

◆ msdev bindlib.dsp /MAKE "bindlib - Win32 Debug"
◆ msdev bindlib.dsp /MAKE "bindlib - Win32 Release"

```

A ce stade, vous avez une librairie `resolv.lib` utilisable, soit dans votre dossier `Debug` , soit sans le dossier `Release` . Copiez ce fichier dans votre dossier `win32build\lib` , en remplaçant le fichier du même nom.

3.3.9.3.5 Compilation

La meilleure façon de compiler est de commander par la version CGI/exécutable.

- Pour les utilisateurs GUI, lancez VC++, puis sélectionnez le menu "File => Open Workspace" et sélectionnez "php4ts". Ensuite, sélectionnez le menu "Build=>Set Active Configuration", et sélectionnez la configuration voulue. Finalement, sélectionnez le menu "Build=>Rebuild All".
- Pour les utilisateurs en ligne de commande, assurez-vous que vous avez enregistré les variables d'environnement C++, ou que vous avez exécuté `vcvars.bat`. Exécutez maintenant l'une des commandes suivantes :
 - ◆ `msdev php4ts.dsp /MAKE "php4ts - Win32 Debug_TS"`
 - ◆ `msdev php4ts.dsp /MAKE "php4ts - Win32 Release_TS"`
 - ◆ A ce stade, vous avez un exécutable `php.exe` utilisable, soit dans votre dossier `Debug_TS` soit sans le dossier `Release_TS`.

Il est possible de faire des modifications particulières mineures en éditant le fichier `main/config.win32.h.in`.

Ensuite, vous pouvez compiler la version CLI qui est conçue pour un usage en ligne de commande. Les étapes sont les mêmes que pour la compilation en mode CGI, hormis le fait que vous devez sélectionner `php4ts_cli - Win32 Debug_TS` ou `php4ts_cli - Win32 Release_TS` dans le fichier de projet. Après une compilation réussie, vous pouvez exécuter le fichier `php.exe` dans l'un des dossiers `Release_TS\cli\` ou `Debug_TS\cli\`.

Note

Si vous voulez utiliser PEAR et l'installateur en ligne de commande, la version CLI-SAPI est obligatoire. Pour plus d'informations sur le projet PEAR et l'installateur, lisez la documentation sur le site de PEAR.

Afin de compiler le module SAPI (`php4isapi.dll`) pour l'intégrer dans Microsoft IIS, vous devez utiliser la configuration `php4isapi-whatever-config` et compiler la DLL nécessaire.

3.3.9.4 Installation des extensions sous Windows

Après avoir installé `PHP` et votre serveur web sous Windows, vous voudrez sûrement ajouter quelques extensions bien pratiques. La table suivante liste une partie des extensions disponibles. Comme indiqué dans le manuel, vous pouvez choisir quelles extensions vous voulez charger en décommentant la ligne `'extension=php_*.dll'` dans le fichier `php.ini`. Vous pouvez aussi charger dynamiquement un module avec la fonction `dl`.

Les fichiers DLLs des extensions `PHP` sont préfixés par `php_` en `PHP` 4, et `php3_` en `PHP` 3. Cela évite la confusion des extensions `PHP` et de leurs librairies.

Note

En `PHP` 4.3.1, les extensions BCMath, Calendar, COM, CType, FTP, MySQL, ODBC, Overload, PCRE, Session, Tokenizer, WDDX, XML et Zlib sont **intégrées**. Vous n'avez pas besoin d'ajouter d'autres extensions pour pouvoir utiliser ces fonctions. Voyez le fichier `README.txt` ou `install.txt` de votre distribution, ou bien cette table pour une liste des modules intégrés.

Editez votre fichier `php.ini` :

- Vous devez modifier l'option de `extension_dir` pour qu'elle pointe sur le dossier qui contient les extensions, là où vous avez placé vos fichiers `php_*.dll`. N'oubliez surtout pas l'anti-slash final. Par exemple :


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

- Pour activer les extensions dans le `php.ini`, il suffit de décommenter les lignes `extension=php_*.dll` dans le `php.ini`. Ceci est fait en effaçant le point-virgule initial de la ligne :

Activer l'extension Bzip2 sous Windows

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

// ... en
extension=php_bz2.dll
```

- Certaines de ces extensions requièrent des librairies DLL supplémentaires pour fonctionner correctement. Certaines d'entre elles sont disponibles dans la distribution. dans le dossier `dlls` mais certaines (comme Oracle (`php_oci8.dll`)), demandent des dlls qui ne sont pas dans la distribution.

Copiez les dlls fournies depuis le dossier `dlls` dans votre PATH Windows. Les bons emplacements sont typiquement

- ♦ c:\windows\system pour Windows 9x/Me
- ♦ c:\winnt\system32 pour Windows NT/2000
- ♦ c:\windows\system32 pour Windows XP

Si vous les avez déjà d'installées sur votre système, ne les écrasez que si `PHP` ne fonctionne pas correctement (et de toutes manières, faites une sauvegarde de ces DLL, en cas de problème).

Extension	Description	Notes
<code>php_bz2.dll</code>	Fonctions de compression <u>bzip2</u>	Aucune
<code>php_calendar.dll</code>	Fonctions de conversions calendaires <u>Calendar</u>	Disponible depuis PHP 4.0.3
<code>php_cpdf.dll</code>	Fonctions <u>ClibPDF</u>	Aucune
<code>php_crack.dll</code>	Fonctions de <u>CrackLib</u>	Aucune
<code>php3_crypt.dll</code>	Fonctions de Crypt	Inconnu
<code>php_ctype.dll</code>	Fonctions <u>ctype</u>	Aucune
<code>php_curl.dll</code>	<u>CURL</u> , Bibliothèque de fonctions "Client URL"	Requiert : <code>libeay32.dll</code> , <code>ssleay32.dll</code> (fournies)
<code>php_cybercash.dll</code>	Fonctions de paiement <u>Cybercash</u>	Aucune
<code>php_db.dll</code>	Fonctions <u>DBM</u>	Obsolètes. Utilisez DBA à la place (<code>php_dba.dll</code>)
<code>php_dba.dll</code>	<u>DBA</u> : DataBase Abstraction, fonctions d'abstraction de base de données (de type dbm)	Aucune
<code>php_dbase.dll</code>	Fonctions <u>dBase</u>	Aucune
<code>php3_dbm.dll</code>	Fonctions Berkeley DB2	Inconnu
<code>php_dbx.dll</code>	Fonctions <u>dbx</u>	
<code>php_domxml.dll</code>	Fonctions de <u>DOM XML</u>	Requiert : <code>libxml2.dll</code> (fournie)
<code>php_dotnet.dll</code>	Fonctions <u>.NET</u>	Aucune
<code>php_exif.dll</code>	Fonctions de lecture des en-têtes JPEG <u>Read EXIF</u>	Aucune
<code>php_fbsql.dll</code>	Fonctions de <u>FrontBase</u>	Aucune
<code>php_fdf.dll</code>	Fonctions <u>FDE</u> : Forms Data Format.	Requiert : <code>fdftk.dll</code> (fournies)
<code>php_filepro.dll</code>	Fonctions <u>filePro</u>	Accès en lecture seule
<code>php_ftp.dll</code>	Fonctions <u>FTP</u>	Disponible depuis PHP 4.0.3
<code>php_gd.dll</code>	Fonctions de traitement des images <u>GD</u>	Aucune
<code>php_gd2.dll</code>	Fonctions de traitement des images <u>GD2</u>	Aucune
<code>php_gettext.dll</code>	Fonctions <u>Gettext</u>	PHP <= 4.2.0 requiert : <code>gnu_gettext.dll</code> (fournie), PHP >= 4.2.3 requiert <code>libintl-1.dll</code> (fournie).
<code>php_hyperwave.dll</code>	Fonctions <u>HyperWave</u>	Aucune
<code>php_iconv.dll</code>	Fonctions de conversions de jeux de caractères <u>ICONV</u>	Requiert : <code>iconv-1.3.dll</code> (fournies)
<code>php_ifx.dll</code>	Fonctions <u>Informix</u>	Requiert : les bibliothèques Informix
<code>php_iisfunc.dll</code>	Fonctions de gestion de IIS	Aucune
<code>php_imap.dll</code>	Fonctions <u>IMAP</u> , POP3 et NNTP	PHP 3 : <code>php3_imap4r1.dll</code>
<code>php_ingres.dll</code>	Fonctions <u>Ingres II</u>	Requiert : librairies Ingres II
<code>php_interbase.dll</code>	Fonctions <u>InterBase</u>	Requiert : <code>gds32.dll</code> (fournies)
<code>php_java.dll</code>	Extension <u>Java</u>	Requiert : <code>jvm.dll</code> (fournies)
<code>php_ldap.dll</code>	Fonctions <u>LDAP</u>	PHP <= 4.2.0 requiert <code>libsasl.dll</code> (fournie), PHP >= 4.3.0 requiert <code>libeay32.dll</code> , <code>ssleay32.dll</code> (fournies).
<code>php_mbstring.dll</code>	Fonctions <u>chaînes multi-octets</u>	Aucune
<code>php_mcrypt.dll</code>	Fonctions de chiffrement <u>Mcrypt</u>	Requiert : <code>libmcrypt.dll</code>
<code>php_mhash.dll</code>	Fonctions <u>Mhash</u>	Aucune
<code>php_mime_magic.dll</code>	Fonctions <u>Mimetype</u>	Requiert : <code>magic.mime</code> (fournie)

php_ming.dll	Fonctions de génération de fichiers Flash avec <u>Ming</u>	Aucune
php_mysql.dll	Fonctions de <u>mSQL</u>	Requiert : msql.dll (fournies)
php3_msql1.dll	mSQL 1 client	Inconnu
php3_msql2.dll	mSQL 2 client	Inconnu
php_mssql.dll	Fonctions <u>MSSQL</u>	Requiert : ntwdbib.dll (fournies)
php3_mysql.dll	Fonctions <u>MySQL</u>	Intégré à PHP 4
php3_nsmail.dll	Fonctions de email Netscape	Inconnu
php3_oci73.dll	Fonctions Oracle	Inconnu
php_oci8.dll	Fonctions <u>Oracle 8</u>	Requiert : librairies clientes Oracle 8
php_openssl.dll	Fonctions <u>OpenSSL</u>	Requiert : libeay32.dll (fournies)
php_oracle.dll	Fonctions de <u>Oracle 7</u>	Requiert : librairies clientes Oracle 7
php_overload.dll	Fonctions relatives à <u>la surcharge d'objets</u>	Intégrée depuis PHP 4.3.0
php_pdf.dll	Fonctions de <u>PDF</u>	Aucune
php_pgsql.dll	Fonctions de <u>PostgreSQL</u>	Aucune
php_printer.dll	Fonctions d' <u>impressions</u>	Aucune
php_shmop.dll	Fonctions de <u>mémoire partagée</u>	Aucune
php_snmp.dll	Fonctions <u>SNMP</u>	NT uniquement!
php_sockets.dll	Fonctions de <u>socket</u>	Aucune
php_sybase_ct.dll	Fonctions <u>Sybase</u>	Requiert : librairies clientes Sybase
php_tokenizer.dll	<u>Tokenizer</u>	Intégrée depuis PHP 4.3.0
php_w32api.dll	Fonctions <u>w32api</u>	Aucune
php_xmllrpc.dll	Fonctions <u>XML-RPC</u>	PHP >= 4.2.1 requiert iconv.dll (fournie)
php_xslt.dll	Fonctions <u>XSLT</u>	Requiert : sablot.dll , expat.dll (fournies)
php_yaz.dll	Fonctions <u>YAZ</u>	Aucune
php_zip.dll	Fonctions de compression <u>Zip</u>	Accès en lecture seule
php_zlib.dll	Fonctions de compression <u>ZLib</u>	Aucune

3.3.10 Installation avec le serveur Apache

Cette section contient des notes spécifiques pour l'installation de PHP avec Apache, aussi bien pour la version Unix que Windows .

3.3.10.1 Détails pour l'installation de PHP sous Apache sous Unix.

Il y a deux méthodes d'installation de PHP avec Apache 1.3.x sous Windows. L'une passe par les CGI (php.exe), l'autre est d'utiliser la DLL de module Apache. Dans les deux cas, vous devrez arrêter le serveur Apache, et éditer le fichier `srm.conf` ou `httpd.conf` , pour configurer Apache avec PHP.

Vous pouvez sélectionner des options à ajouter au fichier `configure` à la ligne 8 depuis la liste complète des options de configuration .

Instructions d'installation avec Apache (version module)
<pre> 1. gunzip apache_1.3.x.tar.gz 2. tar xvf apache_1.3.x.tar 3. gunzip php-x.x.x.tar.gz 4. tar xvf php-x.x.x.tar 5. cd apache_1.3.x 6. ./configure --prefix=/www 7. cd ../php-x.x.x 8. ./configure --with-mysql --with-apache=../apache_1.3.x --enable-track-vars 9. make 10. make install 11. cd ../apache_1.3.x 12. for PHP 3: ./configure --activate-module=src/modules/php3/libphp3.a for PHP 4: ./configure --activate-module=src/modules/php4/libphp4.a 13. make 14. make install Au lieu de cette etape, vous pouvez aussi copier le binaire httpd et remplacer votre executable actuel. Assurez-vous tout de meme que le serveur est bien eteint. 15. cd ../php-x.x.x 16. for PHP 3: cp php3.ini-dist /usr/local/lib/php3.ini for PHP 4: cp php.ini-dist /usr/local/lib/php.ini Vous pouvez editer votre fichier php.ini pour modifier certaines options PHP. Si vous preferez placer ce fichier ailleurs, utilisez --with-config-file-path=/path lors de l'etape 8. 17. Editez votre fichier httpd.conf ou srm.conf file et ajoutez : Pour PHP 3: AddType application/x-httpd-php3 .php3 </pre>

Pour PHP 4: AddType application/x-httpd-php .php
 Vous pouvez choisir n'importe quelle extension que vous voulez ici. .php est uniquement une suggestion. Vous pouvez aussi inclure .html.
 18. Utilisez votre procédure habituelle pour démarrer votre serveur Apache. (vous devez l'eteindre et le redémarrer, pas seulement lui envoyer un signal HUP ou USR1.)

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.

Redémarrage d'apache

```
[1. Nombreuses variantes Linux SysV :
/etc/rc.d/init.d/httpd restart
2. Avec les scripts apachectl :
/path/to/apachectl stop
/path/to/apachectl start
3. httpdctl et httpsdctl (utilisant OpenSSL), similaire a apachectl:
/path/to/httpsdctl stop
/path/to/httpsdctl start
4. En utilisant mod_ssl, ou un autre serveur SSL, manuellement :
/path/to/apachectl stop
/path/to/apachectl startssl
```

Les exécutables `apachectl` et `http(s)dctl` peuvent être situés dans différents dossiers. Si votre système est pourvu de `locate`, de `whereis` ou de `which`, utilisez-les pour retrouver vos programmes.

Différents exemples de compilation PHP pour Apache suivent :

Compilation sur Apache avec l'extension PostgreSQL

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

Cette commande va créer une librairie 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 sur Apache avec l'extension PostgreSQL (librairie partagée)

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

Cette commande va créer une librairie partagée `libphp4.so` pour Apache, mais va aussi créer la librairie 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 `dl`.

Compilation sur Apache avec l'extension PostgreSQL (2)

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

Cette commande va créer une autre librairie partagée `libmodphp4.a`, un fichier `mod_php4.c` et quelques fichiers compagnons dans le dossier `src/modules/php4` du dossier 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`. 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 sur Apache avec l'extension PostgreSQL (2) (librairie partagée)

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

Identique à la version précédente, mais au lieu d'inclure le support PostgreSQL directement dans

l'exécutable final `httpd`, vous allez obtenir une librairie partagée `pgsql.so` que vous pouvez charger dans PHP soit grâce au fichier de configuration `php.ini` ou dynamiquement avec `dl`.

Lorsque vous faites le 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 fichier de configuration d'Apache actuel est fourni avec une section qui ressemble à ceci :
Configuration de l'utilisateur d'exécution d'Apache
User nobody Group "#-1"
A moins que vous ne changiez cette valeur par Group nogroup ou quelques 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 <code>--with-afx=/path/to/apxs</code> . Vous ne devez PAS utiliser la version d'afx qui est dans les sources d'Apache, mais celle qui est réellement installée sur votre système.

3.3.10.2 Installer PHP sous Windows avec Apache 1.3.x

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`), l'autre est d'utiliser les modules Apache DLL. Dans les deux cas, vous devez arrêter le serveur Apache, éditer votre fichier `srm.conf` ou `httpd.conf` pour configurer Apache.

Bien qu'il puisse y avoir quelques différences de configurations 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.

Si vous avez dézippé le package dans `C:\PHP\` comme indiqué dans le chapitre "[Installation manuelle sous Windows](#)", vous devez insérer les lignes suivantes dans votre fichier `srm.conf` ou `httpd.conf` pour qu'il fonctionne en CGI :

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

N'oubliez pas de redémarrer le serveur, avec la commande `NET STOP APACHE` suivie de `NET START APACHE`.

Attention
En utilisant la configuration en CGI, votre serveur sera ouvert à plusieurs attaques potentielles. Lisez donc la section sécurité des CGI pour savoir comment vous en défendre.

Si vous voulez utiliser PHP comme module Apache, vous devez déplacer le fichier `php4ts.dll` dans le dossier `windows/system` (pour Windows 9x/Me) ou `winnt/system32` (pour Windows NT/2000), en écrasant les anciennes versions. Puis, vous devez ajouter les deux lignes suivantes dans le fichier de configuration Apache :

```
LoadModule php4_module c:/php/sapi/php4apache.dll
AddType application/x-httpd-php .php .html
```

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 les alias classiques.

Note

Il se peut qu'après avoir utilisé l'installateur Apache pour Windows, vous deviez définir la directive `AddModule` pour `mod_php4.c` dans le fichier de configuration (`httpd.conf`). Cela peut être fait en ajoutant `AddModule mod_php4.c` à la liste des `AddModule`, juste après le début du fichier de configuration. C'est particulièrement important si la directive `ClearModuleList` est utilisée. Ne pas le faire reviendrait à ne pas déclarer PHP comme module Apache.

Il y a deux méthodes pour utiliser la fonctionnalité de coloration syntaxique, elles dépendent de votre installation. Si vous avez configuré Apache comme module ISAPI, alors il suffit d'ajouter la ligne suivante dans votre fichier de configuration : `AddType application/x-httpd-php-source .phps`

Si vous choisissez de configurer Apache pour utiliser PHP comme exécutable CGI, vous aurez besoin de la fonction `show_source`. Pour cela, créez simplement un script PHP et ajoutez-y le code suivant : `<?php show_source ("original_php_script.php"); ?>`. Substituez `original_php_script.php` par le nom du fichier que vous souhaitez afficher.

Note

Avec le couple Windows–Apache tous les antislashes des noms de chemins tels que "`c:\directory\file.ext`", doivent être convertis en slash c'est à dire "`c:/directory/file.ext`".

3.3.11 Installation avec les serveurs Apache 2.0

Cette section contient les notes et conseils d'installation de ^{PHP} avec le serveur Apache 2.0, aussi bien pour Unix que pour Windows versions.

Attention

N'utilisez pas Apache 2.0 et ^{PHP} sur un serveur de production, ni sur Windows, ni sur Unix.

Il est très recommandé de lire la documentation Apache pour avoir une meilleure connaissance du serveur web Apache 2.0.

3.3.11.1 Notes sur la compatibilité de PHP avec Apache 2.0

Les versions de ^{PHP} suivantes sont reconnues pour fonctionner avec la plus récente version d'Apache 2.0 :

- PHP 4.3.0 ou plus récent, disponible à <http://www.php.net/downloads.php> .
- Les dernières versions de développement. Le code source est disponible à <http://snaps.php.net/php4-latest.tar.gz> , et les exécutables compilés de Windows à <http://snaps.php.net/win32/php4-win32-latest.zip> .
- Une version non stable, téléchargeable sur <http://qa.php.net/> .
- Vous pouvez obtenir ^{PHP} sur le serveur CVS anonyme .

Ces versions de ^{PHP} sont compatibles avec Apache 2.0.40 et plus récent.

Note

Le support des SAPI d'Apache 2.0 a commencé avec PHP 4.2.0. PHP 4.2.3 est connu pour fonctionner avec Apache 2.0.39. N'essayez pas d'utiliser cette version de ^{PHP} avec une autre version

d'Apache 2.0. Nous ne recommandons pas l'utilisation de PHP 4.2.3 avec Apache 2.0.39.

Toutes les versions de ^{PHP} mentionnées ici fonctionnent avec Apache 1.3.x.

3.3.11.2 Installer PHP et Apache 2 sur Linux

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. ./configure --with-apxs2=/usr/local/apache2/bin/apxs
11. make
12. make install
13. cp php.ini-dist /usr/local/lib/php.ini

Editez votre fichier php.ini pour configurer les options PHP.
Si vous préférez placer ce fichier à un autre endroit, utilisez l'option
--with-config-file-path=/path en étape 10.

14. Editez votre fichier httpd.conf et vérifiez la présence des lignes suivantes :

LoadModule php4_module modules/libphp4.so
AddType application/x-httpd-php .php

Vous pouvez installer toutes les extensions que vous souhaitez. .php est
simplement celle que nous suggérons.

Le chemin à droite de la directive LoadModule doit indiquer le chemin
du module PHP de votre système. La directive ci-dessus est correcte si vous
avez suivi les autres étapes à la lettre.

15. Utilisez votre procédure standard pour redémarrer Apache :
/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 sûr, 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-threadée 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

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}.

3.3.11.3 Installation de PHP et Apache 2.0 sur Windows

Lisez donc les notes spécifiques à Windows pour Apache 2.0.

Attention

Apache 2.0 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 plate-formes pour l'instant.

Téléchargez la version la plus récente de Apache 2.0 et une version de PHP appropriée, tel que mentionné au début de ce chapitre. 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 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, et éditez le fichier `httpd.conf` pour configurer Apache avec le support ^{PHP}.

Vous devez insérer les lignes suivantes dans le fichier `httpd.conf` pour réaliser la configuration en mode **CGI** :

Configurer la version de développement de PHP avec Apache 2.0 en CGI

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

Note

A partir de PHP 4.3.0, le nom de l'exécutable CGI a changé, et il est devenu `php-cgi.exe`. Prenez-le en compte, et utilisez la bonne directive pour Apache.

Si vous souhaitez utiliser ^{PHP} comme module dans Apache 2.0, placez bien le fichier `php4ts.dll` dans votre dossier `winnt/system32`, en y écrasant tout autre fichier qui s'y trouverait. Puis vous devrez insérer les deux lignes suivantes dans votre fichier de configuration d'Apache `httpd.conf` pour y ajouter le module PHP d'Apache 2.0 :

Configurer PHP avec Apache 2.0 en module

```
LoadModule php4_module c:/php/sapi/php4apache2.dll
AddType application/x-httpd-php .php
```

Note

N'oubliez pas de remplacer `c:/php/` par votre chemin réel de ^{PHP} dans les exemples ci-dessus. Ne mélangez pas votre installation en module (DLL) avec **différentes versions de PHP**. Vous devez utiliser uniquement les DLL et extensions qui sont fournies avec la version de PHP que vous téléchargez.

3.3.12 CGI/ Installation pour exécution 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. Lisez attentivement le chapitre consacré à la sécurité si vous souhaitez utiliser cette solution.

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.

3.3.12.1 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.

3.3.12.2 Performances

Si vous avez compilé PHP comme programme CGI, vous pouvez évaluer les performances de PHP 3 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.

3.3.12.3 Utiliser les variables prédéfinies

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

3.3.13 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.

3.3.14 Installation sur le serveur 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.
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 [liste complète des options de configuration](#) pour une liste exhaustive.

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`.

3.3.15 Installation avec les serveurs IIS/PWS

Cette section contient des notes sur l'installation de PHP avec IIS (Microsoft Internet Information Server) : [PWS/IIS 3](#) , [PWS 4 ou plus récent](#) et [IIS 4 ou plus récent](#) .

3.3.15.1 Windows et PWS/IIS 3

La méthode recommandée pour configurer ces serveurs est d'utiliser le fichier INF inclus dans la distribution (`php_iis_reg.inf`). 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 :

Attention

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".
ex: `c:\php\php.exe` .

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.

- Répétez ces instructions pour toutes les extensions que vous voulez associer aux scripts

PHP.

- Naviguez jusqu'à : `HKEY_CLASSES_ROOT`
- Dans le menu edit, sélectionnez: `New->Key`.
- Donnez le nom de votre extension à la clé : ex: `.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`. ex: `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.

3.3.15.2 Windows et PWS 4 ou plus récent

Pour installer PHP sous Windows avec PWS 4 ou plus récent, vous avez deux options : l'une est d'avoir PHP sous forme de CGI, l'autre est d'utiliser les modules ISAPI, sous forme de DLL.

Si vous optez pour le CGI, faites ceci :

- Editez le fichier `pws-php4cgi.reg` (dans le dossier `sapi`) pour indiquer la localisation de votre fichier `php.exe`. Les slash doivent être échappés. Par exemple :
`[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map] ".php"="C:\\PHP\\php.exe"`
- Dans le gestionnaire PWS Manager, faites un clic droit sur les dossiers qui supporteront PHP, et sélectionnez "Properties". Cochez l'option "Execute" et confirmez.

Si vous optez pour les modules ISAPI, faites ceci :

- Editez le fichier `pws-php4isapi.reg` (dans le dossier `sapi`) pour indiquer la localisation de votre fichier `php4isapi.dll`. Les slash doivent être échappés. Par exemple :
`[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map] ".php"="C:\\PHP\\sapi\\php4isapi.dll"`
- Dans le gestionnaire PWS Manager, faites un clic droit sur les dossiers qui supporteront PHP, et sélectionnez "Properties". Cochez l'option "Execute" et confirmez.

3.3.15.3 Windows NT/2000 et IIS 4 ou plus récent

Pour installer PHP sous Windows NT/2000 server avec IIS 4 ou plus récent, vous avez deux options : l'une est d'avoir PHP sous forme de CGI, l'autre est d'utiliser les modules SAPI, sous forme de DLL.

Dans les deux cas, vous devez lancer la console "Microsoft Management" (elle peut aussi s'appeler "Internet Services Manager". Elle est située soit dans "Windows NT 4.0 Option Pack" ou dans

"Control Panel=>Administrative Tools" sous Windows 2000). Puis, faites un clic droit sur votre dossier web (qui apparaîtra probablement comme Default Web Server), et sélectionnez "Properties".

Si vous optez pour le CGI, faites ceci :

- Sous "Home Directory", "Virtual Directory", ou "Directory", cliquez sur le bouton "Configuration", et sélectionnez l'onglet "App Mappings".
- Cliquez sur "Add", puis dans la boîte "Executable", tapez : `c:\php\php.exe` (si vous avez installé PHP dans le dossier `c:\php\`).
- Dans la boîte "Extension", tapez le nom de l'extension que vous voulez associer aux scripts PHP. Laissez "Method exclusions" vide, et cochez "Script engine". Vous pouvez aussi préférer cocher la boîte 'check that file exists' – pour un coût faible, IIS (ou PWS) s'assurera que le script existe, et résoudra les problèmes d'authentification avant de démarrer PHP. Cela signifie que vous obtiendrez des messages d'erreur 404 intelligents, plutôt que d'avoir simplement une plainte du serveur comme quoi les CGI n'ont rien retourné.

Répétez les étapes 3 et 4 pour chaque extension que vous souhaitez associer aux scripts PHP. (`.php` et `.html` sont les plus répandues.)

- Configurer la sécurité nécessaire (dans "Internet Service Manager"), et si votre serveur NT utilise NTFS, ajoutez les droits adéquats pour `IUSR_`, au dossier qui contient `php.exe`.

Si vous optez pour les modules ISAPI, faites ceci :

- Si vous ne voulez pas effectuer des authentifications HTTP avec PHP, vous pouvez (et devez) sauter cette étape. Avec ISAPI Filters, ajoutez un nouveau filtre ISAPI. Utilisez PHP comme nom de filtre, et ajoutez simplement le chemin jusqu'à `php4isapi.dll`.
- Sous "Home Directory", cliquez sur le bouton "Configuration". Ajoutez une nouvelle entrée dans "Application Mappings". Utilisez le chemin jusqu'à `php4isapi.dll` comme "Executable", indiquez ".php" comme extension, laissez "Method exclusions" vide, et cochez "Script engine".
- Arrêtez totalement IIS (NET STOP iisadmin)
- Démarrez IIS (NET START w3svc)

3.3.16 Installation sous Netscape et iPlanet Enterprise Serveur

Cette section contient les notes et détails spécifiques à l'installation Netscape et iPlanet, aussi bien pour Sun Solaris que Windows.

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

3.3.16.1 Installer PHP avec Netscape sous Sun Solaris

Pour compiler PHP avec NES ou iPlanet, indiquez le bon dossier d'installation pour l'option de configuration `--with-nsapi = DIR` option. Le dossier par défaut est généralement `/opt/netscape/suitespot/`. Lisez aussi `/php-xxx-version/sapi/nsapi/nsapi-readme.txt`.

Note

Ces instructions, valables pour Sun Solaris 2.6 avec Netscape Enterprise Server 3.6, sont l'oeuvre de bhager at invacare dot com.

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 vortre 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 netscape. 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.

D'abord, vous devez ajouter des chemins dans la variable LD_LIBRARY_PATH pour que PHP trouve toutes les librairies partagées. Le mieux est de le faire dans le script de démarrage de votre serveur Netscape. Les utilisateurs Windows peuvent probablement ignorer cette étape. Le script de démarrage est situé dans : /path/to/server/https-servername/start

Vous pouvez aussi avoir besoin d'éditer les fichiers de configuration qui sont situés dans : /path/to/server/https-servername/config/ .

Instructions de configuration pour Netscape Entreprise Server

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 :

```
type=magnus-internal/x-httpd-php exts=php
```

2. 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_close,php4_execute,php4_auth_trans" shlib="/php4/nsapiPHP4.dll"
Init fn=php4_init errorString="Failed to initialize PHP!"
```

3. Configurez les valeurs par défaut :

```
<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"
```

```
</Object>
```

4. Créez un nouvel objet pour ^{PHP} :

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

5. Configuration d'authentification L'authentification PHP ne peut pas être utilisée avec d'autres authentifications. TOUTES LES FORMES D'AUTHENTIFICATION SONT PASSEES AU SCRIPT PHP. Pour configurer l'authentification PHP pour le serveur entier, ajoutez la ligne suivante :

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

6. Pour utiliser l'identification ^{PHP} sur un seul dossier, utilisez la ligne suivante :

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

Si vous utilisez Netscape Enterprise 4.x, alors, il faut utiliser ceci : placez ces lignes dans les informations d'initialisation des types Mime, et tout ce qui y est rattaché dans la configuration ci-dessus : (fourni par Graeme Hoose, GraemeHoose at BrightStation dot com).

```
Init fn="load-modules" shlib="/path/to/server4/bin/libphp4.so" funcs="php4_init,php4_close,php4_execute,php4_auth_trans"
Init fn="php4_init" LateInit="yes"
```

3.3.16.2 Installer PHP avec Netscape Enterprise sous Windows

Pour installer PHP sous forme de CGI (pour Netscape Enterprise Server, iPlanet, peut être Fastrack), suivez ces instructions :

- 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 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>

Pour installer PHP avec l'interface NSAPI (pour Netscape Enterprise Server, iPlanet, peut-être Fastrack), faîtes 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).
- Stoppez votre serveur web et éditez le fichier `obj.conf`. A la fin de la section Init, placez ces deux lignes (obligatoirement après l'init de type MIME) :

```
Init fn="load-modules" funcs="php4_init,php4_close,php4_execute,php4_auth_trans" shlib="c:/php/sapi/php4nsapi.dll"
Init fn="php4_init" errorString="Failed to initialise PHP!"
```

- Dans la section `<Object name="default">` , placez cette ligne, obligatoirement après toutes les lignes 'ObjectType' et avant les lignes 'AddLog' :

```
Service fn="php4_execute" type="magnus-internal/x-httpd-php"
```

- A la fin du fichier, créez un nouvel objet appelé `x-httpd-php` , en insérant ces lignes :

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

- Redémarrez votre serveur web, et validez les modifications
- Recommencez pour chaque serveur qui doit utiliser PHP.

Plus de détails sur la configuration de PHP en filtre NSAPI sont disponibles :
<http://benoit.noss.free.fr/php/install-php4.html>

3.3.16.3 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" [inikey=value inikey=value...]
```

où `XXX` est le code d'erreur HTTP. Effacez tout 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 la liste 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" [inikey=value inikey=value...]
```

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

3.3.16.4 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 librairie 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 librairie `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.

3.3.17 Installation pour les serveurs OmniHTTPd

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

3.3.17.1 OmniHTTPd 2.0b1 et plus récent pour Windows

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.

- Step 1: Installez le serveur OmniHTTPd.
- Step 2: Faites un clic droit sur l'icône bleue d'OmniHTTPd sur le système, et sélectionnez `Properties`
- Step 3: Cliquez sur `Web Server Global Settings`
- Step 4: Dans l'onglet 'External', entrez : `virtual = .php | actual = c:\path-to-php-dir\php.exe`, et utilisez le bouton "Add" (ajout).
- Step 5: Dans l'onglet ^{Mime}, entrez : `virtual = wwwserver/stdcgi | actual = .php`, et utilisez le bouton "Add" (ajout).
- Step 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} .

3.3.18 Installation Oreilly Website Pro Server

Cette section contient les conseils d'installation spécifiques à Oreilly Website Pro.

3.3.18.1 Oreilly Website Pro 2.5 et plus récent pour Windows

Cette liste décrit comment installer PHP comme CGI exécutable ou module ISAPI avec Oreilly Website Pro sous Windows.

- Editez les "Server Properties" et sélectionnez l'onglet "Mapping".
- Dans la "List" sélectionnez "Associations" et entrez le nom de l'extension voulue (".php") et le chemin jusqu'à l'exécutable (ex. c:\php\php.exe) ou la DLL ISAPI (ex. c:\php\sapi\php4isapi.dll).
- Sélectionnez "Content Types", ajoutez la même extension " .php " et entrez le "content type". Si vous choisissez la forme CGI, entrez "wwwserver/shellcgi"; si vous choisissez la forme module ISAPI, entrez "wwwserver/isapi" (sans les guillemets).

3.3.19 Serveurs Sambar

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

3.3.19.1 Installation pour Sambar sur Windows

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
*.php = c:\php\php4isapi.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.

3.3.20 Installation Xitami

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

3.3.20.1 Xitami pour Windows

Cette liste décrit comment installer ^{PHP} comme CGI exécutable ou module ISAPI avec Xitami sous Windows.

- 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} (souvent `c:\php\php.exe`).
- Cliquez sur le bouton "Save".
- Redémarrez le serveur pour prendre en compte les modifications.

3.3.21 Autres serveurs web

PHP peut être compilé pour fonctionner avec de nombreux autres serveurs web. Reportez-vous à la partie Options des serveurs web pour une liste complète des options de configuration. Les exécutables PHP CGI sont compatibles avec la majorité des serveurs supportant les interfaces CGI.

3.3.22 Des problèmes?

Lisez la FAQ

Certains problèmes sont récurrents : les plus communs sont listés dans la FAQ PHP, disponible dans ce manuel .

3.3.22.1 Autres problèmes

Si vous êtes complètement bloqués, quelqu'un sur la liste de diffusion PHP pourra probablement vous aider. Essayez de consulter les archives, au cas où quelqu'un aurait déjà rencontré votre problème. Les archives sont toujours accessibles à : <http://www.php.net/> . Pour souscrire à la liste de diffusion, envoyez un mail vide à php-install-subscribe@lists.php.net . L'adresse de la liste de diffusion : php-install@lists.php.net .

Si vous voulez obtenir de l'aide sur la liste de diffusion PHP, essayez d'être concis et clair, et pensez à donner tous les détails sur votre environnement (OS, version de PHP, serveur web, CGI ou module, safe mode , etc...), et n'hésitez pas à envoyer suffisamment de code pour que nous puissions reproduire l'erreur.

3.3.22.2 Rapports de Bug

Si vous pensez avoir trouvé un bug dans PHP, n'oubliez pas de le signaler. L'équipe de développement PHP ne le connaît peut être pas, et si vous ne le signaler pas, vos chances de voir le bug corrigé sont nulles. Vous pouvez rapporter des bugs grâce au système de suivi, accessible à <http://bugs.php.net/> .

Lisez les guides (Les bugs : ce qu'il faut faire, et ce qu'il ne faut pas faire) avant d'envoyer n'importe quel rapport!

3.4 Configuration

3.4.1 Le fichier de configuration

Le fichier de configuration (appelé `php3.ini` dans la version 3.0 du PHP, et simplement `php.ini` dans la version 4.0) 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 la version `CGI` le fichier est lu à chaque invocation.

La localisation par défaut du `php.ini` est une option de compilation (voir [FAQ](#)), mais elle peut être changée pour les versions `CGI` et `CLI` avec l'option de ligne de commande `-c` : voyez le chapitre sur l'utilisation de `PHP` en [ligne de commande](#). Vous pouvez aussi utiliser la variable d'environnement `PHPRC` pour indiquer des chemins supplémentaires pour rechercher le fichier `php.ini`.

Toutes les directives `PHP` ne sont pas documentées ci-dessous. Pour une liste de toutes les directives, lisez le `php.ini`, qui est très bien documenté. Vous pouvez jeter un oeil à sa toute dernière version [dans le CVS](#).

Note
La valeur par défaut de la directive <code>PHP</code> <code>register_globals</code> a été changée de on à off en <code>PHP</code> 4.2.0 .

Extrait du <code>php.ini</code>
<pre> ; 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 booléenne peuvent être spécifiées comme ceci : ; true, on, yes ; ou false, off, no, none register_globals = off magic_quotes_gpc = 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" </programlisting> </pre>

3.4.2 Comment modifier la configuration

3.4.2.1 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 et dans les fichiers `.htaccess`.

Dans la version 3.0, 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_`.

Dans la version 4.0, il y a de nombreuses directives Apache qui vous permettent de modifier la configuration de `PHP`.

```
php_value name value
```

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`.

Modification d'une valeur de type php_value

.

```
php_value auto_prepend_file none
```

```
php_flag name 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 name value
```

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`.

Modification d'une valeur de type php_admin_value

.

```
php_admin_value open_basedir 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_value` ne peut pas être modifiée en utilisant le fichier `.htaccess`.

Exemple de configuration Apache

```
<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>
```

Note

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 valeurs numériques peuvent aussi être utilisées dans le `php.ini`.

3.4.2.2 Autres interfaces de configuration de PHP

Indépendamment de l'interface de ^{PHP}, vous pouvez modifier certaines valeurs de la configuration durant l'exécution des scripts, grâce à la fonction `ini_set`. La table suivante fournit une présentation des différents endroits où la directive peut être modifiée.

Constante	Valeur	Signification
PHP_INI_USER	1	La directive peut être configurée dans les scripts.
PHP_INI_PERDIR	2	La directive peut être configurée dans le fichier <code>php.ini</code> , le fichier <code>.htaccess</code> ou encore dans le fichier <code>httpd.conf</code>
PHP_INI_SYSTEM	4	La directive peut être configurée dans le fichier <code>php.ini</code> ou dans <code>httpd.conf</code>
PHP_INI_ALL	7	La directive peut être modifiée n'importe où.

Vous pouvez voir l'état de votre configuration en utilisant la fonction `phpinfo`. Vous pouvez aussi accéder aux valeurs de votre configuration de manière individuelle en utilisant les fonctions `ini_get` or `get_cfg_var`.

3.4.3 Directives de configuration

3.4.3.1 Options Httpd

Nom	Valeur par défaut	Modifications
async_send	"g"	PHP_INI_ALL

3.4.3.2 Options du langage

Nom	Valeur par défaut	Modifications
short_open_tag	On	PHP_INI_SYSTEM PHP_INI_PERDIR
asp_tags	Off	PHP_INI_SYSTEM PHP_INI_PERDIR
precision	"14"	PHP_INI_ALL
y2k_compliance	Off	PHP_INI_ALL
allow_call_time_pass_reference	On	PHP_INI_SYSTEM PHP_INI_PERDIR
expose_php	On	PHP_INI_SYSTEM

Voici un éclaircissement sur l'utilisation des directives de configuration.

short_open_tag bool

Active le support de la forme courte des balises ^{PHP} : `<? ?>` . Si vous voulez utiliser ^{PHP} avec du XML, vous pouvez désactiver cette option afin que vous puissiez utiliser `<?xml ?>` . Sinon vous pouvez l'afficher avec PHP comme ceci : `<?php echo '<?xml version="1.0"'; ?>` . Si cette directive est désactivée, vous devez utiliser la forme longue des balises ^{PHP} `<?php ?>` .

Note
Cette directive affecte aussi les raccourcis <code><?=></code> , qui est l'équivalent de <code><? echo</code> . Utiliser ce raccourci impose l'activation de la directive <code>short_open_tag</code> .

asp_tags bool

Active l'utilisation des balises de type ASP `<% %>`, en plus des traditionnelles balises `<?php ?>` . Cela inclut l'utilisation du raccourci `<%= $value %>`. Pour plus d'informations, reportez-vous à [inclusion dans le HTML](#) .

Note
Le support des balises ASP a été ajouté dans la version 3.0.4.

precision integer

Le nombre de chiffres significatifs à afficher avec les [nombre à virgule flottante](#) .

y2k_compliance bool

Active la compatibilité an 2000 (Cela posera des problèmes avec les navigateurs qui ne sont pas compatibles)

allow_call_time_pass_reference bool

Cette directive force le passage des arguments par référence au moment de l'appel des fonctions. Cette méthode est obsolète, et il est probable qu'elle soit abandonnée dans les futures versions de PHP et Zend. La méthode qui est recommandée est d'indiquer le passage par référence dans la déclaration de la fonction. Nous vous recommandons de tenter de désactiver cette directive et d'adapter vos scripts, afin d'en garantir leur portabilité (vous verrez une alerte s'afficher à chaque fois que vous utilisez cette fonctionnalité, et les arguments seront passés par valeur plutôt que par référence).

Voir aussi [Les références](#) .

expose_php bool

Cette directive active ou non le fait que ^{PHP} est marqué comme installé sur le serveur web (en ajoutant sa propre signature dans les en-têtes du serveur web). Cet affichage n'est pas une menace en terme de sécurité, mais il permet de déterminer

l'utilisation de ^{PHP} sur votre serveur.

3.4.3.3 Limitation des ressources

Nom	Valeur par défaut	Modifications
memory_limit	"8M"	PHP_INI_ALL

Voici un éclaircissement sur l'utilisation des directives de configuration.

memory_limit integer

Cette directive fixe la quantité maximale de mémoire, exprimée en octets, qu'un script peut réserver. Cela évite que des scripts mal écrits consomment toute la mémoire sur le serveur. Afin d'utiliser cette directive, vous devez l'avoir activée au moment de la compilation. De ce fait, vous devez trouver dans la ligne de configuration l'option `--enable-memory-limit`. Notez que vous pouvez lui donner la valeur de `-1` si vous ne souhaitez pas de limitation de consommation de mémoire.

Voir aussi max_execution_time.

3.4.3.4 Gestion des données

Nom	Par défaut	Modifiable
track-vars	"On"	PHP_INI_??
arg_separator.output	"&"	PHP_INI_ALL
arg_separator.input	"&"	PHP_INI_SYSTEM PHP_INI_PERDIR
variables_order	"EGPCS"	PHP_INI_ALL
register_globals	"Off"	PHP_INI_PERDIR PHP_INI_SYSTEM
register_argc_argv	"On"	PHP_INI_PERDIR PHP_INI_SYSTEM
post_max_size	"8M"	PHP_INI_SYSTEM PHP_INI_PERDIR
gpc_order	"GPC"	PHP_INI_ALL
auto_prepend_file	""	PHP_INI_SYSTEM PHP_INI_PERDIR
auto_append_file	""	PHP_INI_SYSTEM PHP_INI_PERDIR
default_mimetype	"text/html"	PHP_INI_ALL
default_charset	"iso-8859-1"	PHP_INI_ALL
always_populate_raw_post_data	"0"	PHP_INI_SYSTEM PHP_INI_PERDIR
allow_webdav_methods	"0"	PHP_INI_SYSTEM PHP_INI_PERDIR

Voici un éclaircissement sur l'utilisation des directives de configuration.

track_vars bool

Si activé, les variables de l'environnement, GET, POST, Cookie et du serveur sont disponibles respectivement dans les tableaux associatifs `$_ENV`, `$_GET`, `$_POST`, `$_COOKIE` et `$_SERVER`.

Notez que depuis PHP 4.0.3, ^{track_vars} est toujours activée.

arg_separator.output string

Le séparateur utilisé par ^{PHP} dans les URL générées par ^{PHP} pour séparer les arguments.

arg_separator.input string

Liste de séparateur(s) utilisé par ^{PHP} pour analyser les URL d'entrées, et en déduire les variables.

Note
Chacun des caractères de cette directive est considéré comme un séparateur.

variables_order string

Etablit l'ordre de priorité des méthodes GET/POST/COOKIE et des valeurs issues de l'environnement et du serveur. Par défaut, cette directive est établie à "EGPCS". En affectant "GP" à cette directive, ^{PHP} ignorera les cookies, le serveur et l'environnement

et écrasera toute variable issue de la méthode GET par les valeurs des variables de même nom, issues de la méthode POST.

Voir aussi register_globals .

register_globals bool

Cette option active l'enregistrement des variables EGPCS (Environnement, GET, POST, Cookie, Serveur), en tant que variables globales. Par exemple, si `register_globals = on`, l'URL suivante `http://www.example.com/test.php?id=3` va créer la variable `$id` dans votre script. Ou bien, `$DOCUMENT_ROOT`, issue de `$_SERVER['DOCUMENT_ROOT']`. Vous pouvez désactiver cette fonction si vous ne voulez pas truffer vos scripts avec des valeurs utilisateurs. Depuis ^{PHP} 4.2.0, cette directive vaut par défaut **off**. Il est recommandé de lire la liste des Variables prédéfinies, pour découvrir les super-globales : `$_ENV`, `$_GET`, `$_POST`, `$_COOKIE` et `$_SERVER`. Lisez aussi le chapitre sur la sécurité Utilisation des variables super-globales pour plus de détails.

Notez que `register_globals` ne peut être modifiée durant l'exécution du script (avec `ini_set`). Cependant, vous pouvez utiliser le fichier `.htaccess` si votre hôte les supporte, comme décrit ci-dessus. Par exemple, vous pouvez inscrire dans votre fichier `.htaccess` : `php_flag register_globals on`.

Note

`register_globals` est combiné avec la directive variables_order .

register_argc_argv bool

Indique à ^{PHP} s'il doit déclarer les variables `argv` et `argc` (qui contiendront alors les mêmes informations passées par méthode GET).

Voir aussi utiliser PHP en ligne de commande . De plus, cette directive a été introduite en PHP 4.0.0 et a toujours été à "on" jusque là.

register_long_arrays boolean

Indique à ^{PHP} s'il doit enregistrer les anciens tableaux `$HTTP_*_VARS`, comme variables prédéfinies. Lorsque cette option est activée (par défaut), les anciens tableaux tels que `$HTTP_GET_VARS` seront définis. Si vous ne les utilisez pas, il est recommandé de les désactiver, pour des raisons de performances. A la place, utilisez les super globales, telles que `$_GET`.

Cette directive a été introduite avec PHP 5.0.0.

post_max_size integer

Configure la taille maximale de données par méthode POST. Cette configuration affecte aussi le téléchargement de fichiers. Pour autoriser l'upload de grands fichiers, il faut que cette configuration soit plus grande encore, et plus grande que upload_max_filesize .

Si la limitation de mémoire est activée dans votre script de configuration memory_limit va aussi affecter le téléchargement de fichiers. En général, memory_limit doit être plus grand que `post_max_size` .

gpc_order string

Cette directive est obsolète. Voyez plutôt variables_order .

auto_append_file string

Spécifie le nom d'un fichier qui sera automatiquement ajouté après le fichier principal. Le fichier est inclus comme s'il avait été appelé avec la fonction `include`, donc include_path est utilisé.

Le mot réservé `none` désactive l'auto-appepending.

Note

Si le script s'arrête par la fonction `exit`, `auto-append` **ne fonctionnera pas**.

`auto_prepend_file` string

Spécifie le nom d'un fichier qui sera automatiquement ajouté avant le fichier principal. Le fichier est inclus comme s'il avait été appelé avec la fonction `include`, donc `include_path` est utilisé.

Le mot réservé `none` désactive l'auto-`appending`.

`default_mimetype` string`default_charset` string

Depuis la version 4.0b4, ^{PHP} fournit toujours une indication du jeu de caractères pour les fichiers générés, dans l'en-tête HTTP `Content-type`. Pour désactiver l'envoi de cette indication, laissez cette directive vide.

`always_populate_raw_post_data` bool

Force le remplissage de la variable `$HTTP_RAW_POST_DATA`.

`allow_webdav_methods` bool

Cette directive autorise la gestion de requêtes HTTP WebDAV par ^{PHP} : PROPFIND, PROPPATCH, MOVE, COPY, etc... Si vous voulez les données postées par ces requêtes, vous devez aussi activer `always_populate_raw_post_data`.

Voir aussi `magic_quotes_gpc`, `magic-quotes-runtime` et `magic_quotes_sybase`.

3.4.3.5 Dossiers

Nom	Par défaut	Modifiable
<code>include_path</code>	<code>PHP_INCLUDE_PATH</code>	<code>PHP_INI_ALL</code>
<code>doc_root</code>	<code>PHP_INCLUDE_PATH</code>	<code>PHP_INI_SYSTEM</code>
<code>user_dir</code>	NULL	<code>PHP_INI_SYSTEM</code>
<code>extension_dir</code>	<code>PHP_EXTENSION_DIR</code>	<code>PHP_INI_SYSTEM</code>
<code>cgi.force_redirect</code>	"1"	<code>PHP_INI_SYSTEM</code>
<code>cgi.redirect_status_env</code>	""	<code>PHP_INI_SYSTEM</code>
<code>fastcgi.impersonate</code>	"0"	<code>PHP_INI_SYSTEM</code>

Voici un éclaircissement sur l'utilisation des directives de configuration.

`include_path` string

Spécifie une liste de dossiers, où les fonctions `require`, `include` et `fopen_with_path()` iront chercher les fichiers. Le format est le même que celui de la variable d'environnement ^{PATH} : une liste de dossiers, séparés par des deux points (:) sous UNIX, et par des points virgules (;) sous Windows.

UNIX `include_path``include_path=".: /php/includes"`Windows `include_path``include_path=".;c:\php\includes"`

Utiliser un point (`.`) dans l'`include_path` permet de réaliser des inclusions relatives au dossier courant.

`doc_root` string

Le dossier racine de ^{PHP}. Cette directive n'est utilisée que si elle est définie. Si ^{PHP} est configuré en `safe mode`, aucun fichier en dehors de ce dossier ne sera accessible. Si ^{PHP} n'a pas été compilé avec l'option `FORCE_REDIRECT`, vous DEVRIEZ utiliser cette option, si vous exécutez ^{PHP} en tant que CGI avec serveur web autre que IIS. La

solution alternative est d'utiliser la directive cgi.force_redirect ci-dessous.

user_dir string

Répertoire où sont stockés les fichiers ^{PHP} dans le répertoire d'un utilisateur. Par exemple, ^{public_html}.

extension_dir string

Définit le répertoire dans lequel le PHP doit chercher les extensions lors du chargement dynamique. Voir aussi enable_dl et ^{dl}.

extension string

Définit les extensions qui doivent être chargées lors du démarrage du PHP.

cgi.force_redirect bool

cgi.force_redirect est nécessaire pour assurer la sécurité de ^{PHP}, lorsqu'il fonctionne en tant que CGI pour la plupart des serveurs web. Même si laissé vide, ^{PHP} va activer cette option par défaut. Vous pouvez la désactiver **à vos risques et périls**.

Note

Utilisateurs Windows : vous pouvez désactiver cette option sans problème avec IIS et en fait, vous DEVEZ la désactiver. Pour faire fonctionner OmniHTTPD ou Xitami, vous devrez aussi la désactiver.

cgi.cgi.redirect_status_env string

Si la directive cgi.force_redirect est activée, et que vous ne fonctionnez pas sous Apache ou Netscape (iPlanet), vous POURRIEZ devoir allouer une variable d'environnement que ^{PHP} ira rechercher pour s'assurer qu'il peut continuer son exécution.

Note

Allouer cette variable PEUT causer des problèmes de sécurité, RENSEIGNEZ VOUS AVANT DE LE FAIRE.

cgi.rfc2616_headers int

Indique à ^{PHP} quel type d'en-tête utiliser lorsqu'il envoie les en-têtes de réponse HTTP. Si cette option vaut 0, ^{PHP} envoie un en-tête ^{Status:} qui est supporté par Apache et d'autres serveurs web. Lorsque cette option vaut 1, ^{PHP} va envoyer des en-têtes compatibles RFC 2616. Laissez le à 0 si vous ne savez pas de quoi il s'agit.

fastcgi.impersonate string

FastCGI sous IIS (sur Windows NT) supporte la possibilité de traiter les tokens de sécurité du client. Cela permet à IIS de définir le contexte de sécurité à utiliser. mod_fastcgi sous Apache ne supporte pas encore cette fonctionnalité (17/03/2002). Utilisez la valeur de 1 pour l'activer sous IIS. La valeur par défaut est zéro.

3.4.3.6 Chargement de fichiers

Nom	Par défaut	Modifiable
file_uploads	"1"	PHP_INI_SYSTEM
upload_tmp_dir	NULL	PHP_INI_SYSTEM
upload_max_filesize	"2M"	PHP_INI_SYSTEM PHP_INI_PERDIR

Voici un éclaircissement sur l'utilisation des directives de configuration.

file_uploads bool

Cette directive indique si les chargements de fichiers sont supportés sur le serveur. Voir aussi upload_max_filesize, upload_tmp_dir et post_max_size.

upload_tmp_dir string

Indique le répertoire utilisé lors du chargement d'un fichier sur le serveur. Ce répertoire doit être accessible en lecture pour l'utilisateur qui exécute le script ^{PHP}, ou par le serveur web.

upload_max_filesize integer

La taille maximale d'un fichier chargé sur le serveur. La valeur est en octets. Par

défaut, elle est de 2 méga-octets.

3.4.3.7 Options de SQL

Nom	Par défaut	Modifiable
<code>sql.safe_mode</code>	<code>"0"</code>	<code>PHP_INI_SYSTEM</code>

Voici un éclaircissement sur l'utilisation des directives de configuration.

```
sql.safe_mode bool
```

3.4.3.8 Configuration des débogueurs

```
debugger.host string
```

Nom DNS ou adresse IP de l'hôte du débogueur.

```
debugger.port string
```

Numéro de port du débogueur.

```
debugger.enabled bool
```

Indique si le débogueur est activé ou pas.

4 Référence

4.1 La syntaxe de base

4.1.1 Passer du HTML au PHP

Lorsque ^{PHP} commence à traiter un fichier, il ne fait qu'afficher le texte qu'il rencontre, jusqu'à ce qu'il rencontre une balise spéciale qui lui indiquera qu'il peut interpréter le code qui suit. L'analyseur ^{PHP} va alors exécuter ce code, jusqu'à ce qu'il rencontre une balise de fin de code ^{PHP}. A ce moment là, il retourne en mode texte, et il affiche simplement le contenu. C'est ce mécanisme qui vous permet d'inclure du code ^{PHP} dans des pages HTML : tout ce qui est placé hors des balises ^{PHP} est affiché sans modification, tandis que le contenu est exécuté.

Il y a quatre jeux de balises qui servent à délimiter des blocs de code ^{PHP}. Seules deux d'entre elles sont toujours actives : c'est `<?php. . .?>` et `<script language="php">. . .</script>`. Tandis que les autres peuvent être activées et désactivées depuis le fichier de configuration `php.ini`. Les balises courtes et les balises ^{ASP} peuvent être pratiques, mais elles ne sont pas aussi portables que les versions longues. De plus, si vous envisagez d'inclure du ^{PHP} dans des fichiers XML ou XHTML, vous devrez utiliser la forme `<?php. . .?>` pour vous conformer au XML.

Le passage du HTML à PHP

```
<?php
1. <?php echo("Si vous voulez afficher du XML ou du XHTML, faites comme ceci.\n"); ?>

2. <? echo ("Ceci est un exemple d'affichage à l'écran en PHP, sous forme d'expression SGML.\n"); ?>
   <?= expression ?>   Raccourci de "<? echo expression ?>"

3. <script language="php">
   echo ("Certains éditeurs HTML (comme FrontPage)
      n'acceptent pas les expressions telles que celle-ci.");
</script>

4. <% echo ("Vous pouvez aussi utiliser le style ASP comme délimiteur."); %>
   <%= $variable; # ceci est un raccourci pour "<%echo .." %>
?>
```

La première méthode est généralement utilisée, car elle permet une implémentation aisée de ^{PHP} avec la prochaine génération de XHTML.

La deuxième forme n'est valable que si vous l'avez activée. Soit en faisant appel à la fonction `short_tags()` (uniquement en ^{PHP} 3), soit en utilisant l'option d'exécution `short_open_tag` dans le fichier de configuration `php.ini`, soit encore en utilisant l'option de compilation `--enable-short-tags`. Même si les balises courtes sont activées dans le fichier de configuration `php.ini-dist`, leur utilisation est déconseillée.

La quatrième forme est seulement disponible si vous l'avez activée en utilisant l'option d'exécution `asp_tags`.

Note

Le support des balises ^{ASP} a été ajouté en ^{PHP} 3.0.4.

Note

L'utilisation des balises courtes doit être évitée lors du développement d'applications et de bibliothèques destinées à être distribuées ou déployées sur des serveurs qui sont hors de votre contrôle, car ces balises ne seront pas forcément supportées par le serveur de destination. Pour

rendre vos scripts portables et faciles à distribuer, évitez les balises courtes.

La marque de fermeture d'un bloc (??) comprend le caractère de nouvelle ligne qui le suit immédiatement, s'il y en a un. De plus, la balise fermante implique un point-virgule. Vous n'êtes pas obligés de mettre le point-virgule à la fin d'un bloc de ^{PHP}.

^{PHP} vous permet d'utiliser des structures telles que :

Méthode avancée

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

Cela fonctionne comme on peut s'y attendre, car ^{PHP} affiche le texte entre les balises ?? et <?php sans modification. L'exemple ci-dessus est très simple, mais c'est un bon système à garder en tête pour afficher de grandes quantités de texte : c'est même plus efficace que d'utiliser les fonctions echo ou print.

4.1.2 Le séparateur d'instructions

Les instructions sont séparées par un point virgule à chaque fin d'instruction, comme en langage C ou en Perl.

La balise de fin (??) implique la fin d'une instruction, et donc ajoute implicitement un point virgule. Les deux exemples suivants sont équivalents.

Le séparateur d'instructions

```
<?php
  echo "Ceci est un test";
??
<?php echo "Ceci est un test" ?>
```

4.1.3 Commentaires

^{PHP} supporte les commentaires comme en C, C++ et Shell Unix. Par exemple :

Commentaires en PHP

```
<?php
  echo "Ceci est un test"; // Ceci est un commentaire sur une ligne comme en C++
  /* Ceci est un commentaire sur plusieurs lignes,
   comme en C et C++ */
  echo "Ceci est encore un test";
  echo "Enfin, le test final"; # Ceci est un commentaire comme en Shell Unix
??
```

Commentaires en PHP (2)

```
<h1>Ceci est un <?php # echo "simple";?> exemple.</h1>
<p>L'entête ci-dessus affiche 'Ceci est un exemple'.
```

Faites attention à ne pas emboîter les commentaires de type langage C, ce qui arrive de temps en temps lorsque vous voulez commenter une grande partie de code.

Commentaires emboîtés
<pre><?php /* echo "Ceci est un test"; /* Ce commentaire va poser un problème */ */ ?></pre>

Les commentaires `//` commentent jusqu'à la fin de la ligne ou bien jusqu'à la fin du bloc (le premier des deux). Cela signifie que du code HTML placé après `// ?>` sera affiché, car ^{PHP} retourne alors en mode texte juste après la balise `?>`.

4.2 Les types

4.2.1 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

Note
Dans ce manuel, vous rencontrerez souvent le type ^{mixed} . C'est un pseudo-type, qui indique que le paramètre peut indifféremment prendre plusieurs types.

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

Note
Si vous voulez vérifier le type et la valeur d'une variable ou d'une <u>expression</u> , utilisez la fonction <u>var_dump</u> .
Si vous souhaitez simplement une représentation lisible pour du débogage, utilisez la fonction <u>gettype</u> . Pour vérifier la présence de certains types, n'utilisez pas <u>gettype</u> , mais plutôt les fonctions ^{is_} <u>type</u> . Par exemple :
Exemples de vérification de type de variable
<pre><?php</pre>

```

$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'incrémente 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";
}
?>

```

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 affecté. Pour plus d'informations, voir le paragraphe [transtypage](#).

4.2.2 Booléens

C'est le type le plus simple. Un booléen exprime les valeurs de `TRUE` ou `FALSE`.

4.2.2.1 Syntaxe

Vous pouvez utiliser les constantes `' TRUE '` et `' FALSE '` pour spécifier une valeur de type `bool`. Ces constantes sont insensibles à la casse.

Utilisation des constantes 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 le passer à une [structure de contrôle](#).

Utilisation des constantes TRUE et FALSE avec une structure de contrôle

```

// == 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";
}

```

4.2.2.2 Conversion en booléen

Reportez-vous au chapitre "[Définition du type](#)" pour plus d'informations sur les conversions.

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

- Le booléen `FALSE`
- L'entier 0 (zéro)
- Le nombre à virgule flottante 0.0 (zéro)
- La chaîne vide, et la chaîne "0"
- Le tableau vide (aucun élément)
- L'objet vide (aucun élément)
- La constante spéciale `NULL`

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

Attention
-1 est considéré comme vrai!
Conversion en booléen
<pre> <?php echo gettype((bool) ""); // bool(false) echo gettype((bool) 1); // bool(true) echo gettype((bool) -2); // bool(true) echo gettype((bool) "foo"); // bool(true) echo gettype((bool) 2.3e5); // bool(true) echo gettype((bool) array(12)); // bool(true) echo gettype((bool) array()); // bool(false) ?> </pre>

4.2.3 Entiers

Un entier est un nombre de l'ensemble des entiers naturels $\mathbb{Z}$: $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$. Il est possible de spécifier les nombres entiers (`integer`) de toutes les manières suivantes : décimale (base 10), hexadécimale (base 16), octale (base 8) éventuellement précédé du signe moins (`-`).

4.2.3.1 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 (`-` or `+`).

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> \$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 </pre>

```

hexadecimal : 0[xX][0-9a-fA-F]+
octal       : 0[0-7]+
integer     : [+]?decimal
            | [+]?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.

4.2.3.2 Dépassement de capacité des entiers

Si un nombre est hors de l'intervalle de validité des entiers, il sera interprété comme un ^{float}.

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)

// this goes also for hexadecimal specified integers:
var_dump( 0x80000000 );
// affiche : float(2147483648)

$million = 1000000;
$large_number = 50000 * $million;
var_dump($large_number);
// affiche : float(500000000000)
?>

```

De même, si une fonction ou un opérateur retourne un entier qui est hors des limites de validité des entiers, il sera aussi automatiquement converti en ^{float}.

En ^{PHP}, il n'y a pas de division entière. ^{1/2} sera un nombre à virgule flottante (^{float}) de valeur ^{0.5}.

Attention

Malheureusement, il y a un bug 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.

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 les fonctions round, floor ou ceil.

Divisions entières

```

<?php
var_dump(25/7);
// float(3.5714285714286)
var_dump((int) (25/7));
// int(3)
var_dump(round(25/7));
// float(4)
?>

```

4.2.3.3 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}.

Reportez-vous à la section définition de type pour plus d'informations sur les conversions.

4.2.3.3.1 Depuis un booléen

FALSE devient 0 (zéro), et TRUE devient 1 (un).

4.2.3.3.2 Depuis un nombre à virgule flottante

Lors de conversion entre un nombre à virgule flottante 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 $\pm 2.15e+9 = 2^{31}$), le résultat est indéfini, car les nombres à virgule flottante n'ont pas assez de précision pour fournir une valeur exacte pour un entier.

Attention
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
<pre><?php echo (int) ((0.1+0.7) * 10); // affiche 7! ?></pre>
Pour plus d'informations, reportez-vous aux <u>alertes</u> liées aux nombres à virgule flottante.

4.2.3.3.3 A partir de chaînes de caractères

Reportez-vous à la section des conversions de chaînes.

4.2.3.3.4 Conversion d'autres types

La conversion d'autres types en entier est indéfinie. Actuellement, ^{PHP} convertit d'abord la valeur en booléen.

Attention
Mais, ne vous fiez pas à ce comportement, car il est susceptible de changer sans préavis!

Voir aussi : Nombres de grande taille et Nombres à virgules flottantes.

4.2.4 Les nombres à virgule flottante

Les nombres à virgule flottante (connus aussi sous le vocable de "double" ou "float" "nombre réels") peuvent être spécifiés en utilisant la syntaxe suivante :

4.2.3 Entiers

Exemple de nombres à virgule flottante

```
<?php
$a = 1.234;
$a = 1.2e3;
?>
```

Attention

Précision des nombres à virgule flottante

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, $\text{floor}((0.1+0.7)*10)$ retournera 7 au lieu de 8 car le résultat de la représentation interne est $7.999999999\dots$.

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 $0.3333333\dots$ 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.

4.2.4.2 Conversion d'un nombre à virgule flottante

Pour savoir quand et comment des chaînes sont converties en nombre à virgule flottante, 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.

4.2.5 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.

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.

4.2.5.1 Syntaxe

Une chaîne peut être spécifiée de trois manières différentes :

- guillemets simples
- guillemets doubles
- syntaxe ^{HereDoc}

4.2.5.1.1 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.

Note
En ^{PHP} 3, une alerte sera affichée si cela arrive avec un niveau de rapport d'erreur de <code>E_NOTICE</code> .

Note
Contrairement aux autres syntaxes, les variables présentes dans la chaîne ne seront PAS remplacées par leurs valeurs.

Exemples d'interpolation des chaînes de caractères
<pre><?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 'Etes vous sûr de vouloir effacer le dossier C:*.??'; // affiche : Etes vous sûr de vouloir effacer le dossier C:*.?? echo 'Etes vous sûr de vouloir effacer le dossier C:*.??'; // affiche : Etes 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 ?></pre>

4.2.5.1.2 Guillemets doubles

Si la chaîne est entourée de guillemets doubles, ^{PHP} va comprendre certaines séquences de caractères :

Séquence	Valeur
<code>\n</code>	Nouvelle ligne (linefeed, LF ou 0x0A (10) en ASCII)
<code>\r</code>	Retour à la ligne (carriage return , CR ou 0x0D (13) en ASCII)
<code>\t</code>	Tabulation horizontale (HT ou 0x09 (9) en ASCII)
<code>\\</code>	Anti-slash
<code>\\$</code>	Caractère \$
<code>\"</code>	Guillemets doubles
<code>\[0-7]{1,3}</code>	Une séquence de caractères qui permet de rechercher un nombre en notation octale.
<code>\x[0-9A-Fa-f]{1,2}</code>	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.

4.2.5.1.3 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. Cette séquence se termine par l'identifiant initial, placé en premier sur une nouvelle 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

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 blanc entre le retour à la ligne et l'identifiant, ou bien entre l'identifiant et le `;`.

Le plus dur est peut être qu'il ne faut pas qu'il y ait un retour à la ligne (`\r`) à la fin de cette ligne, mais seulement un retour chariot (`\n`). Puisque Microsoft Windows utilise la séquence `\r\n` comme terminaison de ligne, la syntaxe `heredoc` risque de ne pas fonctionner là. Cependant, la plupart des éditeurs ^{PHP} fournissent une sauvegarde au format UNIX.

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.

4.2.5.1.4 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.

4.2.5.1.4.1 Syntaxe simple

Dès qu'un signe dollar `$` est rencontré, l'analyseur ^{PHP} va lire autant de caractère 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!";
// OK, car ", " n'est pas autorisé dans les noms de variables
echo "Il a goûté plusieurs $boissons";
// Pas OK, car 's' peut faire partie d'un nom de variable, et PHP recherchera alors $boissons
echo "Il a goûté plusieurs ${boisson}s";
// OK
?>
```

Similairement, vous pouvez aussi 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 de truc comme 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.

4.2.5.1.4.2 Syntaxe complexe

La syntaxe est dite "complexe" car elle permet l'utilisation d'expressions complexes, et non pas parcequ'elle serait obscure. Nuance.

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. Par exemple, vous pouvez utiliser "{\$}" pour obtenir un "{\$}" littéral. Voici quelques exemples :

Syntaxe complexe de chaîne

```
<?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 $foo[bar] est faux
// hors d'une chaîne. En d'autres terme, cela va fonctionner
// car PHP recherche d'abord une constante appelée foo, mais
// il générera une note E_NOTICE (undefined constant).
echo "Ceci est faux : {$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}}";
?>
```

4.2.5.1.5 Accès aux caractères d'une chaîne

Les caractères d'une chaîne sont accessibles en spécifiant leur offset (le premier caractère est d'offset 0) entre accolade, après le nom de la variable.

Note

Pour assurer la compatibilité ascendante, vous pouvez toujours accéder aux caractères avec des crochets. Mais cette syntaxe est obsolète en ^{PHP} 4.

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};
?>
```

4.2.5.2 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](#).

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 régulières Perl](#) et [Expressions régulières 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 .

4.2.5.3 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.

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 à virgule flottante).

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 . 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 .

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 .

4.2.5.4 Conversion de chaînes de caractères

Lorsqu'une chaîne de caractère 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).

Lorsque la première expression est une chaîne de caractères, le type de la variable dépend de la seconde expression.

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 = 1 + "10 gros cochons"; // $foo est du type integer (11)
$foo = "10.0 cochons " + 1;   // $foo est du type integer (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 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";
?>
```

4.2.6 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 champs de ce manuel, mais vous trouverez ici un exemple de toutes ces structures.

4.2.6.1 Syntaxe

4.2.6.1.1 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 string soit un integer 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 ou bien une chaîne. 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'`).

Une valeur peut être n'importe quoi.

Création d'un tableau

```
<?php
array("untableau" => array(6 => 5, 13 => 9, "a" => 43));

echo $arr["untableau"][6];    // 5
echo $arr["untableau"][13];   // 9
echo $arr["untableau"]["a"];  // 43

?>
```

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);

?>
```

Utiliser `TRUE` comme clé revient à utiliser le type `integer` de valeur `1`. Utiliser `FALSE` comme clé revient à utiliser le type `integer` de valeur `0`. Utiliser `NULL` comme clé revient à utiliser le type `string` 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`.

4.2.6.1.2 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 ] = value ; $arr[] = value ;` // `key` est soit une chaîne, soit un entier // `value` peut être n'importe quoi. Si `$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

?>
```


4.2.6.2 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 réindexé. Si vous utilisez uniquement des index numériques (commençant à zéro, et incrémentés de un), vous pouvez effectuer la réindexation 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.

4.2.6.3 Attention aux tableaux

4.2.6.3.1 Pourquoi est ce que `$foo[bar]` est invalide?

Dans vos vieux scripts, vous pouvez avoir utilisé la syntaxe suivante :

Accès aux index de tableaux, ancienne méthode

```
<?php
$foo[bar] = 'ennemi';
echo $foo[bar];
?>
```

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. Actuellement, cela fonctionne car la constante indéfinie est transformée en chaîne ayant la même valeur.

D'autres exemples :

Exemples de conflits de constantes indéfinies

```
<?php
// Affichons toutes les erreurs
error_reporting(E_ALL);

$arr = array('fruit' => 'pomme', 'légume' => 'carotte');

// Correct
print $arr['fruit']; // pomme
print $arr['légume']; // 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 'légume'.
define('fruit','légume');

// 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 concatenation 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 informationnelle";
?>
```

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 il sont déjà réservés.

Note

Pour insister, à l'intérieur d'une chaîne de caractères (`string`) à guillemets doubles, il est correct de ne pas utiliser les guillemets simples dans les index, et `"$foo[bar]"` est valide. Voyez les exemples ci-dessus pour plus de détails sur la syntaxe des variables dans les chaînes.

4.2.6.4 Conversion en tableau

Pour les types scalaires (`integer`, `float`, `string`, `boolean` et `resource`), 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 éléments de tableau. Les clés créées sont les noms des membres.

Si vous convertissez la valeur `NULL` en tableau, vous obtiendrez un tableau vide.

4.2.6.5 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','bleu','vert','jaune');
foreach ( $couleurs as $couleur ){
    echo "Aimez-vous la couleur $couleur?\n";
}
/* Affiche :
Aimez-vous la couleur rouge?
Aimez-vous la couleur bleu?
Aimez-vous la couleur vert?
Aimez-vous la couleur jaune?
*/
?>
```

Notez qu'il n'est pas possible actuellement de modifier les valeurs d'un tableau avec une telle boucle. Une solution pour cela est :

Collection

```
<?php
foreach( $couleurs as $cle => $couleur ){
// ne marche pas
// $couleur = strtoupper($couleur);
// marche :
    $couleurs[$cle] = strtoupper($couleur);
}
print_r($couleur);
/* Affiche :
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);
/* Affiche:
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 .

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 multi-dimensionnels, ou récursifs.

Tableaux multi-dimensionnels, 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
$juices["pomme"]["vert"] = "bon";
?>
```

4.2.7 Les objets

4.2.7.1 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();
?>
```

4.2.7.2 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. Pour les autres valeurs, un membre de l'objet appelé `scalar` contiendra la valeur.

Transtypage

```
<?php
$obj = (object) 'au revoir';
```

```
echo $obj->scalar; // affiche 'au revoir'
?>
```

4.2.8 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.

4.2.8.1 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.

4.2.8.2 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.

Note

Les connexions persistantes représentent un cas particulier, elles ne seront **PAS** détruites. Voyez [connexions persistantes](#) .

4.2.9 La valeur NULL

La valeur spéciale ^{NULL} représente l'absence de valeur. Une variable avec la valeur ^{NULL} n'a pas de valeur.

4.2.9.1 Syntaxe

Il y a seulement une valeur de type ^{NULL}, et c'est la constante ^{NULL}, insensible à la casse.

La valeur NULL

```
<?php
$var = Null;
?>
```

Note

La valeur ^{NULL} a été introduite en ^{PHP} 4.

4.2.10 Pseudo-types utilisés dans cette documentation

4.2.10.1 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 chaîne de caractères et des tableau.

4.2.10.2 number

`number` indique qu'un paramètre peut être du type entier ou nombre à virgule flottante.

4.2.10.3 callback

Certaines fonctions comme par exemple `call_user_function` ou `usort` accepte 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 classe.

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

```
<?php
// simple callback example
function foobar() {
    echo "hello world!";
}
call_user_function("foobar");

// method callback examples
class foo {
    function bar() {
        echo "hello world!";
    }
}

$foo = new foo;

call_user_function(array($foo, "bar")); // object method call
call_user_function(array("foo", "bar")); // static class method call
?>
```

4.2.11 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 évaluées.

Conversion automatique
<pre><?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) ?></pre>

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 type ci-dessous. Si vous voulez changer le type d'une variable, intéressez-vous à aux fonctions de conversion de chaînes .

Si vous voulez forcer le type d'une variable, vous pouvez vous reporter à la section transtypage . Si vous voulez changer le type d'une variable, utilisez settype .

Pour tester les exemples de cette section, il suffit d'en faire un copier/coller, et d'insérer les lignes dans un script ^{PHP} .

Exploitation des exemples
<pre><?php echo "\\$foo==\\$foo; le type est " . gettype(\$foo) . "
\n"; ?></pre>

Note				
Le comportement de la conversion automatique est actuellement indéfinie.				
<table border="1"> <thead> <tr> <th>Conversion automatique indéfinie</th> </tr> </thead> <tbody> <tr> <td> <pre><?php \$a = "1"; // \$a est une chaîne \$a[0] = "f"; // Mais que sont les index de chaîne? ?></pre> </td> </tr> </tbody> </table> Etant donné que ^{PHP} supporte l'indexation de chaîne avec des offsets 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} interprète 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 : <table border="1"> <thead> <tr> <th>Conversion automatique indéfinie(2)</th> </tr> </thead> <tbody> <tr> <td> <pre><?php \$a = "abc"; // \$a est une chaîne \$a{1} = "f"; // \$a vaut maintenant "afc" ?></pre> </td> </tr> </tbody> </table>	Conversion automatique indéfinie	<pre><?php \$a = "1"; // \$a est une chaîne \$a[0] = "f"; // Mais que sont les index de chaîne? ?></pre>	Conversion automatique indéfinie(2)	<pre><?php \$a = "abc"; // \$a est une chaîne \$a{1} = "f"; // \$a vaut maintenant "afc" ?></pre>
Conversion automatique indéfinie				
<pre><?php \$a = "1"; // \$a est une chaîne \$a[0] = "f"; // Mais que sont les index de chaîne? ?></pre>				
Conversion automatique indéfinie(2)				
<pre><?php \$a = "abc"; // \$a est une chaîne \$a{1} = "f"; // \$a vaut maintenant "afc" ?></pre>				

Voyez la section [Accès aux caractères d'une chaîne](#) pour plus de détails.

4.2.11.1 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
<pre><?php \$foo = 10; // \$foo est un entier \$bar = (double) \$foo; // \$bar est un double ?></pre>

Les conversions autorisées sont:

- (int) , (integer) – type entier
- (bool) , (boolean) – booléen
- (real) , (double) , (float) – type double
- (string) – type chaîne
- (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
<pre><?php \$foo = (int) \$bar; \$foo = (int) \$bar; ?></pre>

Note		
Au lieu de transtyper une variable en chaîne, vous pouvez aussi l'insérer entre deux guillemets doubles :		
<table border="1"> <thead> <tr> <th>Transtypage en chaîne</th> </tr> </thead> <tbody> <tr> <td> <pre><?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"; } ?></pre> </td> </tr> </tbody> </table>	Transtypage en chaîne	<pre><?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"; } ?></pre>
Transtypage en chaîne		
<pre><?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"; } ?></pre>		

Le transtypage n'a pas toujours de résultat prévisible. Pour plus d'informations, voyez :

- [Conversion en booléen](#)
- [Conversion en entier](#)
- [Conversion en nombre à virgule flottante](#)
- [Conversion en chaîne de caractères](#)
- [Conversion en tableau](#)
- [Conversion en objet](#)
- [Conversion en ressource](#)

4.3 Les variables

4.3.1 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 (`\x7f-\xff`).

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
$ma#239;s = 'jaune';        // valide; 'i' est ASCII 239.
?>
```

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 à [Expressions](#).

^{PHP} 4 permet aussi d'assigner les valeurs aux variables **par référence**. Cela signifie que la nouvelle variable ne fait que référencer (en d'autres termes, "devient un alias de", ou encore "pointe sur") la variable originale. Les modifications de la nouvelle variable affecteront l'ancienne, et vice versa. Cela signifie aussi qu'aucune copie n'est faite : l'assignation est donc beaucoup plus rapide. Cela se fera notamment sentir dans des boucles, ou lors d'assignation de grands objets (tableaux).

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 Pierre"; // 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.
?>
```

^{PHP} suit les conventions de Perl pour la gestion des opérateurs arithmétiques, et non pas celles du langage C. Par exemple, en Perl `'Z'+1` devient `'AA'`, tandis que en langage C `'Z'+1` devient `'['` (`ord('Z')` == 90, `ord('[')` == 91).

Opérations arithmétiques et caractères

```
<?php
$i = 'w';
for($n=0; $n<6; $n++)
    echo ++$i . "\n";

/*
Affiche ceci :

X
Y
Z
AA
AB
AC
*/
?>
```

4.3.2 Variables prédéfinies

^{PHP} fournit un grand nombre de variables prédéfinies. 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 prédéfinies 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 prédéfinies de ^{PHP}, comme les [tableaux superglobaux](#), est recommandé.

Depuis la version 4.1.0, ^{PHP} fournit un jeu de tableaux prédéfinis, 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 les détails de leur contenu, et une présentation approfondie sur les variables prédéfinies ^{PHP}, et leur nature, reportez-vous à la section [variables prédéfinies](#). De plus, vous noterez que les anciennes variables prédéfinies (`$_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 <u>variable variables</u> .

Si certaines variables de variables_order ne sont pas définies, leur tableau prédéfini ^{PHP} correspondant est laissé vide.

Tableaux superglobaux de ^{PHP}

\$GLOBALS

Contient une référence sur chaque variable qui est actuellement disponible dans l'environnement d'exécution global. Les clés de ce tableau sont les noms des variables globales.

\$ _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 par le protocole HTTP en méthode GET. 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.

\$ _SESSION

Les variables qui sont actuellement enregistrées dans la session attachée au script. C'est la nouvelle version de l'ancienne variable `$HTTP_SESSION_VARS` . Voir le chapitre gestion des sessions pour plus d'informations.

4.3.3 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
<?php

```
$a = 1;
include "b.inc";
?>
```

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)

```
<?php
$a = 1; /* portée globale */
function test() {
    echo $a; /* portée locale */
}
test();
?>
```

Le script n'affichera rien à l'écran car la fonction `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.

4.3.3.1 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 prédéfini `$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 `$GLOBALS` est un tableau associatif avec le nom des variables globales comme clef et les valeurs des éléments du tableau comme valeur des variables. Notez que `$GLOBALS` existe dans tous les

contextes, car ^{\$GLOBALS} est un superglobal . Voici un exemple des super globaux :

Les variables super globales
<pre> <?php function test_global() { // La plupart des variables prédéfinies ne sont pas des "superglobales" et // requiert le mot clé 'global' pour être disponible dans une fonction. global \$HTTP_POST_VARS; print \$HTTP_POST_VARS['name']; // Les superglobales sont accessibles dans tous les contextes // et ne requièrent pas 'global'. Les superglobales sont disponibles // depuis <TT>PHP</TT> 4.1.0 print \$_POST['name']; } ?> </pre>

4.3.3.2 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
<pre> <?php function test() { \$a = 0; echo \$a; \$a++; } ?> </pre>

Cette fonction est un peu inutile car à chaque fois qu'elle est appelée, elle initialise ^{\$a} à 0 et affiche "0". L'incrémentation 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)
<pre> <?php function test() { static \$a = 0; echo \$a; \$a++; } ?> </pre>

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
<pre> <?php function test() { </pre>

```

static $count = 0;
$count++;
echo $count;
if ($count < 10) {
    test();
}
$count--;
}
?>

```

4.3.3.3 Les références avec la 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. Ce peut vous mener à des comportement inattendus, comme 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);
?>

```

Exécuter cet exemple vous donnera :

```

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 un 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();
?>
```

Exécuter cet exemple vous donnera :

```
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.

4.3.4 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 affectée et utilisée 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.

4.3.5 Variables externes à PHP

4.3.5.1 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">
  Name: <input type="text" name="username"><br>
  Email: <input type="text" name="email"><br>
  <input type="submit" name="submit" value="Submit me!">
</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
<TT>PHP</TT>
4.1.0

    print $_POST['username'];
    print $_REQUEST['username'];

    import_request_variables('p', 'p_');
    print $p_username;

// Disponibles depuis
<TT>PHP</TT>
3.

    print $HTTP_POST_VARS['username'];

// Disponibles si la directive register_globals = on. Depuis
//
<TT>PHP</TT>
4.2.0 la valeur par défaut de cette directive est register_globals = off.
// Utiliser ou présumer cette méthode est découragé.

    print $username;
?>
```

Utiliser un formulaire de type GET est similaire, hormis le fait que vous deviez utiliser les variables prédéfinies du 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.exemple.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') {
    print '<pre>';

    print_r($_POST);
    print '<a href="'. $_SERVER['PHP_SELF'] .' ">Essayez à nouveau</a>';

    print '</pre>';
} else {
?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    Name: <input type="text" name="personal[name]"><br>
    Email: <input type="text" name="personal[email]"><br>
    Beer: <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.

4.3.5.1.1 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.

4.3.5.2 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'authentification de l'utilisateur. Vous pouvez établir un cookie grâce à la fonction [setcookie](#). Les cookies font partie intégrante des entêtes 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`. Tout cookie envoyé depuis le client sur le serveur sera automatiquement stocké sous forme de variable, comme pour la méthode POST ou GET.

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 ("MonCookie[]", "test", time()+3600);
?>
```

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 caddie, vous devez implémenter un compteur et l'incrémenter au fur et à mesure. C'est-à-dire:

Exemple avec `setcookie`

```
<?php
$compte++;
SetCookie ("Compte", $compte, time()+3600);
SetCookie ("Caddie[$compte]", $item, time()+3600);
?>
```

4.3.5.3 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 suivi 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`).

4.3.5.4 Détermination du type des variables

Parceque ^{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_long`, `is_double`, `is_string`, `is_array` et `is_object`.

4.4 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]*`

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.

4.4.1 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`, `integer`, `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.
?>
```

4.4.2 Constantes prédéfinies

PHP fournit un grand nombre de constantes prédéfinies. 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.

Une liste des constantes prédéfinies est disponible dans la section Constantes prédéfinies.

4.5 Les expressions

Les expressions sont la partie la plus importante du ^{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 :

```
<?php
function foo () {
    return 5;
}
?>
```

Considérant que vous êtes familier 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 3 types de variables scalaires : les valeurs entières, les nombres à virgule flottante et les chaînes de caractères (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 décrément). 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 décrément. 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'incrémentement et de décrémentement comme c'est le cas dans le langage C. En ^{PHP}, comme en C, il y a deux types d'opérateurs d'incrémentement (pré-incrémentement et post-incrémentement). Les deux types d'opérateur d'incrémentement jouent le même rôle (c'est-à-dire qu'ils incrémentent la variable). La différence vient de la valeur de l'opérateur d'incrémentement. L'opérateur de pré-incrémentement, 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-incrémentement'). L'opérateur de post-incrémentement, 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-incrémentement').

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'incrémentement 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 :

```
<?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 :

```
<?php
function double($i) {
    return $i*2;
}

$b = $a = 5;          /* assigne la valeur 5 aux variables $a et $b */
$c = $a++;            /* post-incrémentement de la variable $a et assignation de
                       la valeur à la variable $c */
$d = $d = ++$b;        /* Pré-incrémentement, 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 <emphasis>après</emphasis>
                    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 allions 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 écrirons ^{expr} pour indiquer toute expression ^{PHP} valide.

4.6 Les opérateurs

4.6.1 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 ("`+`").

Le tableau suivant dresse une liste de la priorité des différents opérateurs dans un ordre croissant de priorité.

Associativité	Opérateurs
gauche	,
gauche	or
gauche	xor
gauche	and
droite	print
gauche	= += -= *= /= .= %= &= = ^= ~= <<=>=>
gauche	? :
gauche	
gauche	&&
gauche	
gauche	^
gauche	&
non-associatif	== != === !==
non-associatif	< <= > >=

gauche	<< >>
gauche	+ - .
gauche	* / %
droite	! ~ ++ -- (int) (double) (string) (array) (object) @
droite	[
non-associatif	new

Note

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`.

4.6.2 Les opérateurs arithmétiques

Vous rappelez-vous des opérations élémentaires apprises à l'école ?

Exemple	Nom	Résultat
<code>\$a + \$b</code>	Addition	Somme de <code>\$a</code> et <code>\$b</code> .
<code>\$a - \$b</code>	Soustraction	Différence de <code>\$a</code> et <code>\$b</code> .
<code>\$a * \$b</code>	Multiplication	Produit de <code>\$a</code> et <code>\$b</code> .
<code>\$a / \$b</code>	Division	Quotient de <code>\$a</code> et <code>\$b</code> .
<code>\$a % \$b</code>	Modulo	Reste de <code>\$a</code> divisé par <code>\$b</code> .

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. Si l'un des opérandes est un nombre à virgule flottante, ou bien le résultat d'une opération qui retourne une valeur non entière, un nombre à virgule flottante sera retourné.

Voir aussi le manuel sur les [fonctions mathématiques](#).

4.6.3 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

```
<?php
$a = ($b = 4) + 5;
// $a est maintenant égal à 9, et $b vaut 4.
?>
```

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

```
<?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!";
?>
```

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. ^{PHP 4} supporte aussi l'assignation par référence, 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. D'un autre côté, la recopie est très rapide.

4.6.4 Opérateurs sur les bits

Les opérateurs sur les bits vous permettent de manipuler les bits dans un entier.

Exemples avec les opérateurs sur les bits

```
<?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 ascii : s #0 #4 #0 #0
// 'a' ^ 'e' = #4
?>
```

Exemple	Nom	Résultat
<code>\$a & \$b</code>	ET (AND)	Les bits positionnés à 1 dans \$a ET dans \$b sont positionnés à 1.
<code>\$a \$b</code>	OU (OR)	Les bits positionnés à 1 dans \$a OU \$b sont positionnés à 1.
<code>\$a ^ \$b</code>	Xor	Les bits positionnés à 1 dans \$a OU dans \$b 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).

4.6.5 Opérateurs de comparaison

Les opérateurs de comparaison, comme leur nom l'indique, vous permettent de comparer deux valeurs.

Exemple	Nom	Résultat
<code>\$a == \$b</code>	Egal	Vrai si \$a est égal à \$b .
<code>\$a === \$b</code>	Identique	Vrai si \$a est égal à \$b et qu'ils sont de même type (PHP 4 seulement).
<code>\$a != \$b</code>	Différent	Vrai si \$a est différent de \$b .
<code>\$a <> \$b</code>	Différent	Vrai si \$a est différent de \$b .
<code>\$a < \$b</code>	Plus petit que	Vrai si \$a est plus petit strictement que \$b .
<code>\$a > \$b</code>	Plus grand	Vrai si \$a est plus grand strictement que \$b .
<code>\$a <= \$b</code>	Inférieur ou égal	Vrai si \$a est plus petit ou égal à \$b .
<code>\$a >= \$b</code>	Supérieur ou égal	Vrai si \$a est plus grand ou égal à \$b .

Un autre opérateur conditionnel est l'opérateur ternaire (":?"), qui fonctionne comme en langage C.

```
<?php
// Exemple d'utilisation pour l'opérateur ternaire
$action = (empty($_POST['action'])) ? 'default' : $_POST['action'];

// L'expression ci-dessus est identique à
if (empty($_POST['action'])) {
    $action = 'default';
} else {
    $action = $_POST['action'];
}
```

```
}
?>
```

L'expression $(\text{expr1}) ? (\text{expr2}) : (\text{expr3})$ renvoie la valeur de l'expression expr2 si l'expression expr1 est vraie, et l'expression expr3 si l'expression expr1 est fausse.

Voir aussi [strcasecmp](#), [strcmp](#) et le chapitre sur les [types](#).

4.6.6 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.

```
<?php
/* Erreur intentionnelle (le fichier n'existe pas): */
$my_file = @file ('non_persistent_file') or
    die ("Impossible d'ouvrir le fichier : L'erreur est : '$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.

Note

La plupart des fonctions d'accès aux bases de données ne retournent pas d'erreur ^{PHP}. Il faut y accéder avec une fonction du type `base_de_donnees_get_error()`.

Voir aussi [error_reporting](#).

4.6.7 Opérateur d'exécutions

PHP supporte un opérateur d'exécution : guillemets obliques (`"`"`). Notez bien la différence entre les guillemets simples (sur la touche 4), et ceux-ci (sur la touche de la livre anglaise). ^{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).

```
<?php
$output = `ls -al`;
echo "<pre>$output</pre>";
?>
```

Note

Cet opérateur est désactivé lorsque le safe mode est activé ou bien que la fonction shell_exec est désactivée.

Voir aussi system, passthru, exec, shell_exec, popen et escapeshellcmd.

4.6.8 Opérateurs d'incrément/Décrément

PHP supporte les opérateurs de pré et post incrémentation et décrémentation, comme en C.

Exemple	Nom	Résultat
<code>++\$a</code>	Pré-incrémente	Incrémente \$a de 1, puis retourne \$a .
<code>\$a++</code>	Post-incrémente	Retourne \$a , puis l'incrémente de 1.
<code>--\$a</code>	Pré-décrémente	Décrémente \$a de 1, puis retourne \$a .
<code>\$a--</code>	Post-décrémente	Retourne \$a , puis décrémente \$a de 1.

Voici un exemple simple

```
<?php
echo "<h3>Post-incrémentation</h3>";
$a = 5;
echo "Devrait valoir 5: " . $a++ . "<br>\n";
echo "Devrait valoir 6: " . $a . "<br>\n";
echo "<h3>Pré-incrémentation</h3>";
$a = 5;
echo "Devrait valoir 6: " . ++$a . "<br>\n";
echo "Devrait valoir 6: " . $a . "<br>\n";
echo "<h3>Post-décrémentation</h3>";
$a = 5;
echo "Devrait valoir 5: " . $a-- . "<br>\n";
echo "Devrait valoir 4: " . $a . "<br>\n";
echo "<h3>Pré-décrémentation</h3>";
$a = 5;
echo "Devrait valoir 4: " . --$a . "<br>\n";
echo "Devrait valoir 4: " . $a . "<br>\n";
?>
```

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";

/*
   Produit un résultat identique à celui-ci :
X
Y
Z
AA
AB
AC
*/
?>
```

4.6.9 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.

! \$a	NON (Not)	Vrai si \$a est faux.
\$a && \$b	ET (And)	Vrai si \$a ET \$b sont vrais.
\$a \$b	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](#).

4.6.10 Opérateurs de chaînes

Il y a deux opérateurs de chaînes. 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.

```
<?php
$a = "Bonjour ";
$b = $a . "Monde!";
// $b contient "Bonjour Monde!"
$a = "Bonjour ";
$a = $a . "Monde!";
// $a contient "Bonjour Monde!"
?>
```

4.6.11 Opérateurs de tableaux

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

```
<?php
$a = array("a" => "pomme", "b" => "banane");
$b = array("a" => "poire", "b" => "fraise", "c" => "cerise");

$c = $a + $b;

var_dump($c);
?>

array(3) {
  ["a"]=>
  string(5) "pomme"
  ["b"]=>
  string(6) "banane"
  ["c"]=>
  string(6) "cerise"
}
```

4.7 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.

4.7.1 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 ()`

```
<?php
if (expression)
    commandes
?>
```

Comme nous l'avons vu dans le paragraphe consacré aux expressions, ^{expr} est convertie en sa valeur booléenne. Si l'expression ^{expr} 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) {
    print "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.

4.7.2 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) {
    print "a est plus grand que b";
} else {
    print "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`.

4.7.3 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` ()

```
<?php
if ($a > $b) {
    print "a est plus grand que b";
} elseif ($a == $b) {
    print "a est égal à b";
} else {
    print "a est plus petit que b";
}
?>
```

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`.

4.7.4 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, `endwhile;`, `endfor;`, `endforeach;`, ou `endswitch;`.

Instruction `if` () alternative

```
<?php if ($a == 5): ?>
A égal 5
<?php endif; ?>
```

Dans l'exemple ci-dessus, le bloc HTML "A = 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`

```
<?php
if ($a == 5):
```

```

    print "a égale 5";
    print "...";
elseif ($a == 6):
    print "a égale 6";
    print "!!!";
else:
    print "a ne vaut ni 5 ni 6";
endif;
?>

```

Voir aussi while , for , et if pour d'autres exemples.

4.7.5 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 ()`

```

<?php
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

```

<?php
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) {
    print $i++; /* La valeur affiche est $i avant l'incréméntation
                (post-incréméntation) */
}
/* exemple 2 */
$i = 1;
while ($i <= 10):
    print $i;
    $i++;
endwhile;
?>

```

4.7.6 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 {
    print $i;
} while ($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) {
        print "i n'est pas suffisamment grand";
        break;
    }
    $i *= $factor;
    if ($i < $minimum_limit) {
        break;
    }
    print "i est bon";
    ...process 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.

4.7.7 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()

```
<?php
for (expr1; expr2; expr3) statement
?>
```

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

```
<?php
/* exemple 1 */
for ($i = 1; $i <= 10; $i++) {
    print $i;
}
/* exemple 2 */
for ($i = 1;;$i++) {
    if ($i > 10) {
        break;
    }
    print $i;
}
/* exemple 3 */
$i = 1;
for (;;) {
    if ($i > 10) {
        break;
    }
    print $i;
    $i++;
}
/* exemple 4 */
for ($i = 1; $i <= 10; print $i, $i++);
?>
```

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

```
<?php
for (expr1; expr2; expr3): statement; ...; endfor;
?>
```

Les autres langages ont l'instruction `foreach` pour accéder aux éléments d'un tableau. PHP 3 ne dispose pas d'une telle fonction; PHP 4 en dispose (voir [foreach](#)). En PHP 3, vous pouvez combiner `while` avec `list` et `each` pour obtenir le même résultat. Reportez-vous aux exemples de la documentation.

4.7.8 foreach

PHP 4 (mais pas PHP 3) inclut 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()

```
<?php
foreach(array_expression as $value) commandes
foreach(array_expression as $key => $value) 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`.

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

De plus, notez que `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.

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
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
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) {
    print "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 $v) {
    print "\$a[$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 => $v) {
    print "\$a[$k] => $v.\n";
}

/* exemple foreach 4 : tableaux multi-dimensionnels */
```

```

$a[0][0] = "a";
$a[0][1] = "b";
$a[1][0] = "y";
$a[1][1] = "z";

foreach($a as $v1) {
    foreach ($v1 as $v2) {
        print "$v2\n";
    }
}

/* exemple foreach 5 : tableaux dynamiques */

foreach(array(1, 2, 3, 4, 5) as $v) {
    print "$v\n";
}

?>

```

4.7.9 break

L'instruction `break` permet de sortir d'une structure `for`, `while`, `foreach` 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 "At 10; quitting<br>\n";
            break 2; /* Termine le switch et la buocle while. */
        default:
            break;
    }
}

?>

```

4.7.10 continue

L'instruction `continue` est utilisée dans une boucle afin d'éluder les instructions de l'itération courante afin de passer directement à l'itération suivante.

Note

Notez qu'en PHP, la structure `switch` est considérée comme une boucle par `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 (!($% 2)) { // évite les membres impairs  
    continue;  
}  
  
fonction_quelconque($valeur);  
  
$i = 0;  
while ($i++ < 5) {  
    echo "Dehors<br>\n";  
    while (1) {  
        echo "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&Milieu<br>\n";  
        while (1) {  
            echo "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&Intérieur<br>\n";  
            continue 3;  
        }  
        echo "Ceci n'est jamais atteint.<br>\n";  
    }  
    echo "Ceci non plus.<br>\n";  
}  
?>
```

4.7.11 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 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 ($i == 0) {
    print "i égale 0";
}
if ($i == 1) {
    print "i égale 1";
}
if ($i == 2) {
    print "i égale 2";
}
switch ($i) {
    case 0:
        print "i égale 0";
        break;
    case 1:
        print "i égale 1";
        break;
    case 2:
        print "i égale 2";
        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:
        print "i égale 0";
    case 1:
        print "i égale 1";
    case 2:
        print "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 case vide

```
<?php
switch ($i) {
    case 0:
    case 1:
    case 2:
        print "i est plus petit que 3 mais n'est pas négatif";
        break;
    case 3:
        print "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:
        print "i égale 0";
        break;
    case 1:
        print "i égale 1";
        break;
    case 2:
        print "i égale 2";
        break;
    default:
        print "i n'est ni égal à 2, ni à 1, ni à 0.";
}
?>
```

Une autre chose à mentionner est que l'instruction ^{case} peut être une expression à 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:
        print "i égale 0";
        break;
    case 1:
        print "i égale 1";
        break;
    case 2:
        print "i égale 2";
        break;
    default:
        print "i n'est ni égal à 2, ni à 1, ni à 0";
endswitch;
?>

```

4.7.12 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 `declare`

```

<?php
declare (directive) statement
?>

```

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 `statement` du bloc de `declare` sera exécutée. Comment elle sera exécutée, et quels effets cela aura dépend de la directive utilisée dans le bloc `directive`.

4.7.12.1 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 débbugger, 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](#).

4.7.13 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.

4.7.14 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 $somefile;

require ('unfichier.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 subisse 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

La version Windows de ^{PHP} ne supporte 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](#) .

4.7.15 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.

Lorsqu'un fichier est inclus, le code qu'il contient hérite des variables qui existent dans ce [contexte](#) . Toute variable accessible à la ligne qui réalise l'exécution est aussi accessible dans le fichier inclus.

Exemple avec [include](#)

```
vars.php
<?php
$couleur = 'verte';
$fruit = 'pomme';
?>

test.php
<?php
echo "Une $couleur $fruit"; // Une
include 'vars.php';
echo "Une $color $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

La version Windows de ^{PHP} ne supporte 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.exemple.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.exemple.com comme du PHP
include 'http://www.exemple.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.exemple.com/file.php?foo=1&bar=2';

$foo = 1;
$bar = 2;
include 'file.txt'; // OK.
include 'file.php'; // OK.

?>

```

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.

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; // prints 'PHP'

$bar = include 'noreturn.php';

echo $bar; // prints 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. Un autre moyen d'inclure des fichiers dans des variables est d'utiliser la fonction fopen ou file ou d'utiliser include avec les fonctions de contrôle de sortie .

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 .

4.7.16 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 inclut 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}.

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 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!</pre>

Attention
La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si <code>allow_url_fopen</code> est activé.

Voir aussi `require`, `include`, `include_once`, `get_required_files`, `get_included_files`, `readfile` et `virtual`.

4.7.17 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}.

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 Windows.
<code>include_once</code> est insensible à la casse sous Windows
<pre>include_once("a.php"); // ceci inclut le fichier a.php include_once("A.php"); // ceci inclut encore le fichier a.php sous Windows!</pre>

Attention
La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si <code>allow_url_fopen</code> est activé.

Voir aussi `require`, `include`, `require_once`, `get_required_files`, `get_included_files`, `readfile` et `virtual`.

4.8 Les fonctions

4.8.1 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 .

En ^{PHP} 3, les fonctions doivent être définies avant qu'elles ne soient utilisées. Ce n'est plus le cas en ^{PHP} 4.

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 fonctions

```
<?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();
?>
```

^{PHP} ne supporte pas le surchargement de fonction, ni la destruction ou la redéfinition de fonctions déjà déclarées.

^{PHP} 3 ne supporte pas un nombre variable d'arguments (voir [valeurs par défaut d'arguments](#) pour plus d'informations). ^{PHP} 4 supporte les deux : 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.

4.8.2 Les arguments de fonction

Des informations peuvent être passées à une fonction en utilisant un tableau d'arguments, dont chaque élément est séparé par une virgule. Un élément peut être une variable ou une constante.

^{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](#). Fonctionnellement, on peut arriver au même résultat en passant un tableau comme argument :

Nombre variable d'argument sous forme de tableau

```
<?php
function takes_array($input) {
    echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
?>
```

4.8.2.1 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.';
}
$string = 'Ceci est une chaîne';
add_some_extra($string);
echo $string; // affiche 'Ceci est une chaîne, et un peu plus.'
?>
```

Si vous souhaitez passer une variable par référence à une fonction mais de manière ponctuelle, vous pouvez ajouter un '&' devant l'argument dans l'appel de la fonction:

Passage de valeur par référence, ponctuellement

```
<?php
function foo ($bar) {
    $bar .= ', et un peu plus.';
}
$string = 'Ceci est une chaîne';
```

```
foo ($str);
echo $str;    // affiche 'Ceci est une chaîne'
foo (&$str);
echo $str;    // affiche 'Ceci est une chaîne, et un peu plus.'
?>
```

4.8.2.2 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_apero ($type = "ricard") {
    return "Servir un verre de $type.\n";
}
echo servir_apero();
echo servir_apero("whisky");
?>
```

La fonction ci-dessus affichera :

```
Servir un verre de ricard.
Servir un verre de whisky.
```

La valeur par défaut d'un argument doit obligatoirement être une constante, et ne peut être ni une variable, ni un membre de classe.

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'affiche du code ci-dessus est le suivant :

```
Warning: Missing argument 2 in call to faireunyaourt() in
/usr/local/etc/httpd/htdocs/PHP 3test/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'affichage de cet exemple est le suivant :

```
Préparer un bol de acidophilus framboise.
```

4.8.2.3 Nombre d'arguments variable

PHP 4 supporte 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.

4.8.3 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.

Retourner des valeurs de fonction

```
<?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 plusieurs valeurs 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();
?>
```

4.8.4 Variables fonctions

^{PHP} supporte le concept de variables fonctions. 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 variables fonctions ne peuvent pas fonctionner avec les éléments de langage comme les `echo`, `unset`, `isset`, `empty`, etc... C'est une des différences majeures entre les fonctions ^{PHP} et les éléments de langage.

Exemple de variable fonction

```
<?php
function foo() {
    echo "dans foo(<br>\n";
}
function bar( $arg = '' ) {
    echo "Dans bar(); l'argument était '$arg'.<br>\n";
}
$func = 'foo';
$func();
$func = 'bar';
$func( 'test' );
```

`?>`

Voir aussi [call_user_func](#) , [les variables variables](#) et [function_exists](#) .

4.8.5 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 coder en PHP .

Voir aussi [function_exists](#) , [l'index des fonctions](#) , [get_extension_funcs](#) et [dl](#) .

4.9 Les classes et les objets

4.9.1 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 Caddie {
// Éléments 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
<TT>TRUE</TT>
;
        } else {
            return FALSE;
        }
    }
}
?>
```


L'exemple ci-dessus définit la classe `Caddie` 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

Les notes suivantes ne sont valables que pour PHP 4.

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.

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.

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 Caddie {
    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 Caddie {
    var $date_du_jour;
    var $name;
    var $owner;
    var $items;
    function Caddie() {
        $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 Caddie;
$cart->add_item("10", 1);
$another_cart = new Caddie;
$another_cart->add_item("0815", 3);
?>
```

L'instruction ci-dessus crée l'objet `$cart` de la classe `Caddie`. La fonction `add_item()` est appelée afin d'ajouter l'article numéro 10 dans le panier. 3 articles numéro 0815 sont ajoutés au caddie `$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

vosre 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 $ncart->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'.

4.9.2 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 Caddie_nomme extends Caddie {
    var $owner;
    function set_owner ($name) {
        $this->owner = $name;
    }
}
?>
```

L'exemple ci-dessus définit la classe `Caddie_nomme` qui possède les mêmes variables que la classe `Caddie` 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 Caddie_nomme; // Création d'un panier nominatif
$ncart->set_owner ("kris"); // Affectation du nom du panier
print $ncart->owner; // Affichage du nom du panier
$ncart->add_item ("10", 1); // (héritage des fonctions de la classe père)
?>
```

4.9.3 Constructor : constructeur

Attention

En ^{PHP} 3 et ^{PHP} 4, les constructeurs se comportent différemment. La sémantique de ^{PHP} 4 est fortement recommandée.

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. En ^{PHP} 3, une fonction devient le constructeur si elle porte le même nom que la classe. En ^{PHP} 4, une fonction devient un constructeur si elle porte le même nom que la classe dans laquelle elle est définie. La différence est subtile, mais cruciale.

Constructeur de classe

```
<?php
class Auto_Caddie extends Caddie {
    function Auto_Caddie () {
        $this->add_item ("10", 1);
    }
}
// Cette syntaxe est valable en PHP 3 et 4
?>
```

L'exemple ci-dessus définit la classe ^{Auto_Caddie} qui hérite de la classe Caddie 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
// Cette syntaxe est valable en PHP 3 et 4
class Constructor_Cart extends Cart {
    function Constructor_Cart ($item = "10", $num = 1) {
        $this->add_item ($item, $num);
    }
}
// Création du caddie
$default_cart = new Constructor_Cart;
// Création d'un vrai caddie
$different_cart = new Constructor_Cart ("20", 17);
?>
```

Attention

En ^{PHP} 3, les classes dérivées et les constructeurs ont un certains nombre de limitations. Les exemples suivants doivent être lus avec beaucoup d'attention pour comprendre ces limitations.

Constructeur de classe en PHP 4

```
<?php
class A {
    function A() {
        echo "Je suis le constructeur de A.<br>\n";
    }
}
class B extends A {
    function C() {
        "Je suis une fonction standard.<br>\n";
    }
}
// Aucun constructeur n'est appelé en PHP 3!!
$b = new B;
?>
```

En ^{PHP} 3, aucun constructeur ne sera appelé dans l'exemple ci-dessus. La règle en ^{PHP} 3 est : 'Un constructeur est une méthode qui a le même nom que la classe'. Le nom de la classe est B, et il n'y

a pas de fonction qui s'appelle B() dans la classe B. Rien ne se passe.

Ceci est corrigé en ^{PHP} 4, avec l'introduction d'une nouvelle règle : Si une classe n'a pas de constructeur, le constructeur de la classe de base est appelé, s'il existe. L'exemple ci-dessus affichera : ' Je suis le constructeur de A.
' en ^{PHP} 4.

Constructeur de classe en PHP 4

```
<?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 {
    function C() {
        echo "Je suis une fonction standard.<br>\n";
    }
}
// Cette syntaxe va appeler B() comme constructeur.
$b = new B;
?>
```

En ^{PHP} 3, 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. La règle de ^{PHP} 3 est 'Un constructeur est une méthode qui a le même nom que la classe'. ^{PHP} 3 ne se soucie guère si la fonction est définie dans la classe B ou si elle a été héritée.

Ceci est corrigé en ^{PHP} 4, avec l'introduction d'une nouvelle règle : 'Un constructeur est une méthode de même nom que la classe, définie dans la classe elle-même'. Donc, en ^{PHP} 4, la classe B n'a pas de constructeur par elle-même, et le constructeur de la classe A aura été appelé, affichant : ' Je suis le constructeur de A.
'.

Attention

Ni ^{PHP} 3 ni ^{PHP} 4 n'appelle 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.

Note

Il n'y a pas de destructeurs en ^{PHP} 3 et ^{PHP} 4. Vous pouvez utiliser la fonction [register_shutdown_function](#) à la place, pour simuler un destructeur.

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} .

4.9.4 Opérateur ::

Attention

La documentation suivante n'est valable que pour ^{PHP} 4 et plus récent.

Parfois, il est pratique de faire référence aux fonctions est 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`.

4.9.5 parent

Il arrive que vous ayez à écrire du code qui fasse 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();
?>
```

4.9.6 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 sauver toutes les variables d'un objet. Le nom de la classe sera sauvé mais par 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 sauvant la définition de la classe A dans un fichier séparé, et en l'incluant dans les deux pages page1.php et page2.php.

```
<?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`.

4.9.7 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 sauveés.

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 sauveés.

4.9.8 Références dans un constructeur

Créer des références dans un constructeur peut conduire à des résultats étranges. Ce tutoriel vous guide pour éviter ces problèmes.

```
<?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 `=&` :

```
<?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('crée dans le constructeur');
$bar2->echoName();
$globalref[1]->echoName();
/* affiche :
   crée dans le constructeur
   crée dans le constructeur
   crée 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 :
   crée dans le constructeur
   modifié */
// 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.

```

<?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();
/*
output:
class a: 10
class b: 10
class b: 10
class a: 11
class b: 11
class b: 11
*/
?>

```

4.9.9 Comparer des objets en PHP 4

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 exécutons le code suivant :

Exemple de comparaison d'objets en PHP 4

```

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);

```

Cet exemple 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

```
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);
```

Qui va donner l'affichage suivant :

```
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
```

4.9.10 Comparer des objets en PHP 5

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.

En ^{PHP} 5, la comparaison d'objets est plus compliquée qu'en ^{PHP} 4, afin d'être plus proche du comportement des langages orienté objet (bien que ^{PHP} n'en soit pas un).

Lors de l'utilisation de l'opérateur de comparaison `==`, les objets sont comparées de manière simple, à savoir : deux objets sont égaux si ils ont les mêmes attributs et valeurs, et qu'ils sont des instances de la même classe, dans le même espace de nom.

D'un autre coté, lors de l'utilisation de l'opérateur d'identité `===`, les objets sont identiques uniquement si ils font référence au même objet de la même classe, dans le même espace de nom.

Un exemple va illustrer ces règles.

Exemple de comparaison d'objets en PHP 5

```
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;
    }
}

namespace Other {

    class Flag {
        var $flag;

        function Flag($flag=true) {
            $this->flag = $flag;
        }
    }
}

$o = new Flag();
$p = new Flag();
$q = $o;
$r = new Other::Flag();

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 de même nom, mais dans des espaces de noms différents\n";
compareObjects($o, $r);
```

Cet exemple 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 de même nom, mais dans des espaces de noms différents

```
o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE
```

4.10 Les références

4.10.1 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 : ici, 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.

4.10.2 Que font les références ?

Les références vous permettent de faire pointer deux variables sur le même contenu. Par exemple, lorsque vous faites :

```
<?php
$a =& $b
?>
```

cela signifie que `$a` et `$b` pointent sur la même variable.

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.

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):

```
<?php
$bar =& new fooclass();
$foo =& find_var ($bar);
?>
```

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.

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 :

```
<?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](#).

4.10.3 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 :

```
<?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).

4.10.4 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 :

```
<?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.

Les objets suivants peuvent être passés par référence :

- Une variable, i.e. `foo($a)`
- Un nouvel objet, i.e. `foo(new fooobj())`
- Une référence, retournée 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 :

```
<?php
function bar() // Notez l'absence de &
{
    $a = 5;
    return $a;
}
foo(bar);
foo($a = 5) // Expression, pas une variable
foo(5) // Constante, pas une variable
?>
```

Ces fonctionnalités sont valables à partir de PHP 4.0.4.

4.10.5 Retourner des références

Retourner des références est toujours utile lorsque vous voulez utiliser une fonction pour savoir à quoi est liée une variable. Lorsque vous retournez une variable par paramètre, utilisez le code suivant

```
<?php
function &find_var($param) {
    // ...code...
    return $found_var;
}
$foo =& find_var ($bar);
$foo->x = 2;
?>
```

Dans cet exemple, la propriété de l'objet est retournée dans `find_var` et lui sera affectée, et non pas à la copie, comme cela sera le cas avec une syntaxe par 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).

4.10.6 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,

```
<?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.

4.10.7 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 :

4.10.7.1 Références globales

Lorsque vous déclarez une variable `global $var`, vous créez en fait une référence sur une variable globale. Ce qui signifie que

```
<?php
$var =& $GLOBALS["var"];
?>
```

Et que, si vous détruisez la variable `$var`, la variable globale ne sera pas détruite.

4.10.7.2 `$this`

Dans une méthode d'objet `$this` est toujours une référence sur l'objet courant.

5 Sécurité

5.1 Sécurité

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 le 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 comme 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. Comment créer cet environnement et le sécuriser 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.

5.1.1 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 n'est authentifié que 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.

5.1.2 Binaires CGI

5.1.2.1 Faiblesses connues

Utiliser le 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 placer l'interpréteur à l'intérieur du répertoire cgi-bin. Même si le binaire 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, le 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 le 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.

5.1.2.2 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).

5.1.2.3 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 du 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 du PHP, comme exposé ci-dessous.

5.1.2.4 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 du 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 du 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".

5.1.2.5 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 :

```
#!/usr/local/bin/php
```

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`.

5.1.3 Module Apache

Lorsque le 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"). 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 authentification. 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.

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 professionnels 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 restriction Apache, pour restreindre les activités anonymes liées aux internautes.

5.1.4 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'authentification. 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 du 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 de noms de fichiers sécurisée

```

<?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.

5.1.5 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.

5.1.5.1 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 par exemple 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.

5.1.5.2 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.

5.1.5.3 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 par exemple 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
<pre>// Stockage du mot de passe hashé \$query = sprintf("INSERT INTO users(name,pwd) VALUES('%s', '%s')", addslashes(\$username), md5(\$password)); \$result = pg_exec(\$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_exec(\$connection, \$query); if (pg_numrows(\$result) > 0) { echo "Bienvenue, \$username!"; } else { echo "Identification échouée pour \$username."; }</pre>

5.1.5.4 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 par exemple les identifications et authentifications, 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)

```
$offset = argv[0]; // Attention, aucune validation!
$query = "SELECT id, name FROM products ORDER BY name LIMIT 20 OFFSET $offset;";
// avec PostgreSQL
$result = pg_exec($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 :

```
// 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 d'ignorer le reste de la requête, en utilisant les symboles `--` pour mettre en commentaire.

Un moyen possible 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

```
$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

```
,
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}, l'injection SQL va arriver.

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 nomination pour stocker des noms d'utilisateurs et des mots de passe.

Modifier un mot de passe ... et gain de droits!

```
$query = "UPDATE usertable SET pwd='$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 sera alors devenue :

Une requête et son injection

```
// $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)

```
$query = "SELECT * FROM products WHERE id LIKE '%$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

```
$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.

5.1.5.4.1 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 régulière 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 guillemet toutes les valeurs non numériques qui sont passées à la base de données, et protégez-vous des guillemets dans les valeurs avec la fonction addslashes ou addslashes. Voyez le premier exemple. Comme les exemples l'on montré, les guillemets statiques peuvent être facilement contournés.
- 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 coté 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.

5.1.6 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'authentification 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, utilisez la constante `E_NONE`, pour isoler votre code.

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");
}
?>
```

5.1.7 Utilisation des variables super-globales

Une fonctionnalité de PHP qui peut être utilisée pour améliorer la sécurité est de configurer PHP en désactivant l'option `register_globals`. En supprimant la possibilité que les variables envoyées par les internautes soient injectées automatiquement dans le script PHP, vous pouvez restreindre la quantité de variables non-protégées. Les intrus devront prendre beaucoup plus de temps pour corrompre les mécanismes d'envoi de données, et vos variables internes seront nettement mieux protégées.

Bien que cela augmente d'autant les efforts à fournir pour écrire un script PHP, les bénéfices peuvent en être nettement plus intéressants.

Travailler avec `register_globals = on`

```
<?php
if ($username) {
// attention, cette valeur peut être parasitée via GET/POST/COOKIES
    $good_login = 1;
}
if ($good_login == 1) {
// attention, cette valeur peut être parasitée via GET/POST/COOKIES
    fpassthru ("/données/très/très/sensibles/index.html");
}
?>
```

Travailler avec `register_globals = off`

```
<?php
if($_COOKIE["username"]){
// ne peut provenir que d'un cookie, corrompu ou pas
    $good_login = 1;
// Impossible à parasiter
    fpassthru ("/données/très/très/sensibles/index.html");
}
?>
```

En utilisant intelligemment ceci, il est même possible de détecter les tentatives de corruption. Si vous savez à l'avance d'où la variable doit venir (GET ou POST ou COOKIE), vous pouvez tester les données. Même si cela ne vous garantit pas contre la corruption de ces données, cela impose aux pirates de bien savoir comment corrompre les données.

Détection de corruption de variables

```
<?php
if ($_COOKIE['username'] &&
    !$_POST['username'] &&
    !$_GET['username'] ) {
// D'autres vérifications pour vérifier l'origine du nom d'utilisateur fourni
}
```

```

$good_login = 1;
fpassthru ("/données/très/très/sensibles/index.html");
} else {
mail("admin@example.com", "Tentative de piratage", $_SERVER['REMOTE_ADDR']);
echo "Problème de sécurité, l'administrateur est alerté.";
exit;
}
?>

```

Bien entendu, désactiver l'option `register_globals` ne signifie pas que votre code devient ouvert à tous. Mais il faut aussi vérifier toutes les données qui vous sont fournies par les utilisateurs, et plutôt deux fois qu'une.

5.1.8 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).

5.1.9 Masquer PHP

Quelques astuces permettent de masquer PHP, et cela entrave les pirates qui recherchent des faiblesses dans votre système. En désactivant l'option `expose_php` 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

```
# Faire que le code PHP ressemble à un autre langage
AddType application/x-httpd-php .asp .py .pl
```

Ou masquez le complètement :

Masquer PHP avec des types inconnus

```
# Faire que le code PHP ressemble à un autre langage qui n'existe pas
AddType application/x-httpd-php .bop .foo .133t
```

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

```
# Faire que le code PHP ressemble à du html
AddType application/x-httpd-php .htm .html
```

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.

5.1.10 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.

6 Caractéristiques

6.1 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 ^{\$PHP_AUTH_TYPE} contenant respectivement le nom d'utilisateur, le mot de passe et le type d'identification. Actuellement, seule l'identification simple ("Basic") est supportée. 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

```
<?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>";
}
?>
```

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 buggés, 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 contrediront, et échoueront).

Notez cependant que les manipulations ci-dessus n'empêchent pas quiconque possède une page non authentifiée 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'] == $_SERVER['PHP_AUTH_USER'])) {
    authenticate();
}
else {
    echo "<p>Welcome: {$_SERVER['PHP_AUTH_USER']}<br>";
    echo "Old: {$_REQUEST['OldAuth']}";
    echo "<form action='{$_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='Re Authenticate'>\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 aussi que tout ceci ne fonctionne pas sous Microsoft IIS et que les limitations de PHP en version CGI sont dues aux limitations de IIS.

Note

Si le safe mode est activé, l'uid du script es ajouté automatiquement dans l'en-tête ^{realm}, comme faisant partie de l'en-tête ^{WWW-Authenticate}.

6.2 Cookies

PHP supporte les cookies 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'authentifier et de suivre les visiteurs. Vous pouvez envoyer un cookie avec la commande setcookie. 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.

Tous les cookies qui sont envoyés au client seront automatiquement retournés au script PHP, et

transformés en variable, exactement comme pour GET et POST. Si vous souhaitez affecter plusieurs valeurs à un seul cookie, ajoutez [] au nom du cookie. Pour plus détails, reportez-vous à la fonction [setcookie](#).

En PHP 4.1.0 et plus récent, le tableau superglobal `$_COOKIE` sera aussi prédéfini avec les cookies transmis par l'internaute. `$HTTP_COOKIE_VARS` est aussi prédéfini dans les versions de PHP plus anciennes, lorsque la directive `track_vars` est activée.

Pour plus de détails, y compris des notes sur les bogues des navigateurs, voyez la fonction [setcookie](#).

6.3 Gestion des chargements de fichier

6.3.1 Chargements de fichiers par méthode POST

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). Cette fonctionnalité permet de charger des fichiers textes ou binaires. Avec l'authentification et les fonctions de manipulation des fichiers, vous avez un contrôle total sur le chargement et la gestion des fichiers chargés.

Note
Notes de configuration
Voir aussi les directives file_uploads , upload_max_filesize , upload_tmp_dir et post_max_size , dans <code>php.ini</code>

Notez bien que PHP supporte aussi le chargement par la méthode PUT comme dans le navigateur Netscape Composer et les clients Amaya du W3C. Reportez-vous au chapitre sur le [support de la méthode PUT](#).

Un écran de chargement de fichiers peut être constitué en créant un formulaire de la manière suivante :

Formulaire de chargement de fichier
<pre><FORM ENCTYPE="multipart/form-data" ACTION="_URL_" METHOD="POST"> <INPUT TYPE="hidden" name="MAX_FILE_SIZE" value="1000"> Envoyez ce fichier : <INPUT NAME="userfile" TYPE="file"> <INPUT TYPE="submit" VALUE="Send File"> </FORM></pre>

Le paramètre `_URL_` doit pointer sur un fichier PHP. L'option `MAX_FILE_SIZE` cachée doit précéder le nom du fichier à charger, et représente la taille maximale du fichier à charger. La valeur est donnée en octets. Dans ce script, les valeurs suivantes doivent être définies pour assurer un chargement correct.

Attention
Le champ <code>MAX_FILE_SIZE</code> est là à titre de conseil au navigateur. 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.

Les variables définies après un téléchargement de fichiers diffèrent beaucoup de version en version. La variable `$FILES` existe depuis PHP 4.1.0. Le tableau `$HTTP_POST_FILES` existe depuis PHP 4.0.0. Ces tableaux contiennent toutes les données sur les fichiers téléchargés. L'utilisation de la variable

`$FILES` est recommandée. Si la directive `PHP register_globals` vaut **on**, les variables reliées existeront aussi. `register_globals` vaut par défaut **off** depuis `PHP 4.2.0`.

Le contenu du tableau `$FILES` est détaillé dans notre exemple ci-dessous. Notez que l'un suppose que le nom de la variable du fichier téléchargé est **userfile**, tel que défini dans le formulaire ci-dessus.

```
$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".
$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. error code a été introduit
    en PHP 4.2.0
```

Note

Dans les versions de `PHP` antérieure à la version 4.1.0, la variable `$FILES` s'appelait `$HTTP_POST_FILES` et ce n'était pas une variable auto-globale comme l'est `$FILES`. `PHP 3` ne supporte pas `$HTTP_POST_FILES`.

Lorsque `register_globals` vaut **on** dans le `php.ini`, des variables supplémentaires sont rendues disponibles. Par exemple `$userfile_name` vaut `$FILES['userfile']['name']`, `$userfile_type` vaut `$FILES['userfile']['type']`, etc... Gardez à l'esprit que depuis `PHP 4.2.0`, `register_globals` vaut **off** par défaut. Il n'est pas prudent de supposer que cette directive vaut **on**.

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

Les exemples ci-dessous sont valables avec les versions de `PHP` plus récentes que la 4.0.2. Voyez les fonctions `is_uploaded_file` et `move_uploaded_file`.

```
<?php
// En PHP < 4.1.0, $HTTP_POST_FILES doit être utilisé
//   à la place de $FILES.
// En PHP plus ancien que 4.0.3, utilisez copy() et is_uploaded_file()
//   à la place de move_uploaded_file

$uploadaddir = '/var/www/uploads/';

print "<pre>";
if (move_uploaded_file($FILES['userfile']['tmp_name'], $uploadaddir . $FILES['userfile']['name'])) {
    print "Le fichier est valide, et a été téléchargé
        avec succès. Voici plus d'informations :\n";
    print_r($FILES);
} else {
    echo "Attaque par upload potentielle. Voici plus d'informations :\n";
    print_r($FILES);
}
?>
```

Le script PHP qui reçoit le fichier chargé doit pouvoir gérer le fichier de manière appropriée. Vous pouvez utiliser la variable `$file_size` pour recalculer tous les fichiers qui sont trop gros ou trop petits. Vous pouvez utiliser la variable `$file_type` pour recalculer les fichiers qui n'ont pas le bon type. Quelques soient les actions, ce script doit pouvoir supprimer le fichier du dossier temporaire, ou le déplacer ailleurs. Depuis `PHP 4.2.0`, vous pouvez utiliser l'information dans `$FILES['userfile']['error']` et adapter

vosre politique en fonction des code d'erreur . Quelque soit votre politique, vous devriez soit effacer le fichier du dossier temporaire, soit le déplacer.

Le fichier sera automatiquement effacé du fichier temporaire à la fin du script, s'il n'a pas été déplacé ou renommé.

6.3.2 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.

Note

Ces constantes sont apparues en PHP 4.3.0.

6.3.3 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 fichier `php3.ini` , ou par `php3_upload_max_filesize` dans les directives Apache. La valeur par défaut est 2 méga-octets.

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'en-tête MIME. Tant que c'est le cas, ^{CERN httpd} ne pourra pas effectuer de chargements de fichiers.

6.3.4 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 téléchargement 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="Send files">
</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.

6.3.5 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 stdin */
$putdata = fopen("php://stdin","r");

/* Ouvre un fichier pour écriture */
$f = fopen("myputfile.ext","w");

/* Lit les données par bloc de 1ko
   et les écrit dans le fichier */
while ($data = fread($putdata,1024))
    fwrite($f,$data);

/* Ferme les flux */
fclose($f);
fclose($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 W3C Amaya. Les requêtes de type PUT sont beaucoup plus simples que les chargements de fichiers, et elles ressemblent à :

```
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 :

```
Script PUT /put.php3
```

Elle indique à Apache qu'il doit envoyer les requêtes de chargement par méthode PUT au script `put.php3`. Bien entendu, cela présuppose que vous avez activé PHP pour qu'il prenne en charge les fichiers de type `.php3`, et que PHP est actif.

Dans le fichier `put.php3`, vous pouvez mettre ceci :

```
<?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.

6.4 Utilisation des fichiers à distance

Aussi longtemps que le support des gestionnaires d'URL ("URL fopen wrapper") sont activés dans le `php.ini`, 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 ne supportaient 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.

Connaitre 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 par exemple '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 authentification basique).

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 svslog.

6.5 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 `php3_max_execution_time`, dans le fichier Apache `.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_timeout`. Cette fonction retournera vrai 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, les deux fonctions `connection_timeout` et `connection_aborted` vont retourner `TRUE`. Vous pouvez aussi vérifier les deux états en un seul appel avec la fonction `connection_status`. Cette fonction va retourner un champ de bits, avec les états. Si les deux états sont actifs, l'état retourné prendra la valeur 3.

6.6 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, 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's 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.

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](#), [imap_popen](#), [ingres_pconnect](#), [mysql_pconnect](#), [mssql_pconnect](#), [mysql_pconnect](#), [oci_plogon](#), [odbc_pconnect](#), [ora_plogon](#), [pfsockopen](#), [pg_pconnect](#) et [sybase_pconnect](#).

6.7 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".

6.7.1 Sécurité et Safe Mode

Nom	Par défaut	Modifiable
safe_mode	"0"	PHP_INI_SYSTEM
safe_mode_gid	"0"	PHP_INI_SYSTEM
safe_mode_include_dir	NULL	PHP_INI_SYSTEM
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_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [ini_set](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

safe_mode bool

Active ou non le mode de sécurité de ^{PHP}. Lisez le chapitre [sécurité](#) pour plus d'informations.

safe_mode_gid bool

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 string

Les vérifications basées sur le ^{UID} ou ^{GID} sont ignorées lorsque les fichiers sont issus du dossier et de ses sous-dossiers, indiqué dans cette directive. 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", si 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/"

safe_mode_exec_dir string

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.

safe_mode_allowed_env_vars string

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 string

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 string

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.

Sous Windows, séparez les dossiers par des deux points. Sur les autres systèmes, séparez les dossiers avec un point-virgule. Lorsque ^{PHP} est utilisé comme module Apache, les chemins de la directive `open_basedir` des dossiers parents sont automatiques 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"` si 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 est de permettre l'ouverture de tous les fichiers.

`disable_functions` string

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` string

Cette directive vous permet de désactiver certaines classes 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}.

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 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 :

```
<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
```

6.7.2 Fonctions désactivées par le Safe Mode

Voici une liste non-exhaustive des fonctions désactivées par le `Safe Mode`.

Fonction	Limitations
<code>dbmopen</code>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<code>dbase open</code>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<code>filepro</code>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<code>filepro rowcount</code>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<code>filepro retrieve</code>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
Fonctions <code>ifx_</code>	restrictions <code>sql_safe_mode</code> , (<code>!= Safe Mode</code>)
Fonctions <code>ingres_</code>	restrictions <code>sql_safe_mode</code> , (<code>!= Safe Mode</code>)
Fonctions <code>mysql_</code>	restrictions <code>sql_safe_mode</code> , (<code>!= Safe Mode</code>)
<code>pg loimport</code>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

<u>posix_mkfifo</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>putenv</u>	Obéis aux directives <code>safe_mode_protected_env_vars</code> et <code>safe_mode_allowed_env_vars</code> . Voir aussi la documentation de <u>putenv</u>
<u>move_uploaded_file</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>chdir</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>dl</u>	Cette fonction est désactivée par le <code>safe-mode</code>
<u>opérateur guillemets obliques</u>	Cette fonction est désactivée par le <code>safe-mode</code>
<u>shell_exec</u> (équivalent fonctionnel des guillemets obliques)	Cette fonction est désactivée par le <code>safe-mode</code>
<u>exec</u>	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.
<u>system</u>	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.
<u>passthru</u>	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.
<u>popen</u>	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.
<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. (on source and 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 sticky bit
<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: only the target is checked)
<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>getallheaders</u>	En Safe Mode, les en-têtes commençant par 'authorization' (sensible à la casse) ne seront pas retournés. Attention : cette protection est mise en défaut par la fonction <u>getallheaders</u> sur les serveurs AOL!
<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'authentification.
<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 le USER. (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)
Toutes les fonctions qui utilisent les wrappers de fichiers. (php4/main/fopen_wrappers.c)	??

6.8 Utiliser PHP en ligne de commande

Depuis la version 4.3, ^{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 pas mal de différences entre

le type `CLI SAPI` et les autres `SAPI`, qui sont expliqués dans ce chapitre.

Le `CLI SAPI` a été publié pour la première fois avec la version `PHP 4.2.0`, mais il est resté expérimental, et doit être explicitement activé avec la commande `--enable-cli`, lorsque vous exécutez le script `./configure`. Depuis PHP 4.3.0, le `CLI SAPI` n'est plus expérimental, et est donc **toujours** compilé et installé comme binaire PHP (appelé aussi `php.exe` sous Windows).

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`.

- 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 <code>CLI SAPI</code>	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 <u>la bufferisation de sortie</u> si vous voulez retarder un affichage, ou bien en manipuler ultimement le contenu.
<code>max_execution_time</code>	0 (unlimited)	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 <code>TRUE</code> à cette directive, vous aurez toujours accès à la variable argc (représentant le nombre d'arguments passés à l'application) et argv (le tableau contenant les arguments passés) dans le <code>CLI SAPI</code> . Depuis <code>PHP 4.3.0</code> , les variables <code>\$argc</code> et <code>\$argv</code> sont définies et remplies avec les valeurs appropriées, en utilisant <code>CLI SAPI</code> . Avant cette version, la création de ces variables était liée au comportement des versions <code>CGI</code> et <code>MODULE</code> , qui requièrent l'activation de la directive <code>PHP register_globals</code> . Indépendamment de la version ou de la valeur de <code>register_globals</code> , vous pouvez toujours accéder à <code>\$SERVER</code> et <code>\$HTTP_SERVER_VARS</code> . Par exemple: <code>\$SERVER['argv']</code>

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 par exemple `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 <code>\$stdin = fopen('php://stdin', 'r');</code>
STDOUT	Un descripteur de fichier déjà disponible vers stdout. Cela évite de l'ouvrir avec <code>\$stdout = fopen('php://stdout', 'w');</code>
STDERR	Un descripteur de fichier déjà disponible vers stderr. Cela évite de l'ouvrir avec <code>\$stderr = fopen('php://stderr', 'w');</code>

Etant donné ce qui précède, vous n'avez pas besoin d'ouvrir un fichier 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.

- **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 pourra être

```
$ 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
<code>CGI SAPI</code> supporte le même comportement de <code>CLI SAPI</code> , en lui passant l'option <code>-C</code> , lorsque vous l'invoquez en ligne de commande.

La liste des options de ligne de commande fournies par `PHP` est disponible en exécutant `PHP` avec l'option `-h` :

```
Usage: php [options] [-f] <file> [args...]
php [options] -r <code> [args...]
php [options] [-- args...]
-s Display colour syntax highlighted source.
-w Display source with stripped comments and whitespace.
-f <file> Parse <file>.
-v Version number
-c <path>|<file> Look for php.ini file in this directory
-a Run interactively
-d foo[=bar] Define INI entry foo with value 'bar'
-e Generate extended information for debugger/profiler
-z <file> Load Zend extension <file>.
-l Syntax check only (lint)
-m Show compiled in modules
-i PHP information
-r <code> Run PHP <code> without using script tags <?..?>
-h This help

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 :

```
php mon_script.php

php -f mon_script.php
```

Les deux méthodes (en utilisant `-f` ou pas) exécutent 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.

- 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` fonctionne sans ces balises, et les ajouter conduira alors à une erreur d'analyse.

- Donner du code `PHP` dans l'entrée standard (`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 peut aussi les recevoir. Le nombre d'arguments n'est pas limité par PHP, mais par le shell, qui a une limite en terme de nombre de caractères qui peuvent être passés. Généralement, vous n'atteindrez pas cette limite, mais cela reste possible. Les arguments passés au script seront transmis via la variable tableau `$argv` . L'index zéro contiendra toujours le nom du script appelé. Si le code a été fourni en ligne de commande, ou via `stdin` , ce premier élément vaudra `-` . L'autre variable globale fournie est `$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. Vous pouvez aussi ajouter après `#!/usr/bin/php` les options nécessaires. Vous pouvez aussi le faire en ligne de commande. De cette manière, le script peut être exécuté comme un script shell 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 755 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.

Option	Description
<code>-s</code>	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, sans les en-têtes. <pre><code> [...] </code></pre> Note Cette option ne fonctionne pas avec l'option <code>-r</code>.
<code>-w</code>	Affiche la source sans les commentaires et les espaces. Note Cette option ne fonctionne pas avec l'option <code>-r</code>.
<code>-f</code>	Analyse et exécute le fichier donné après l'option <code>-f</code>. Cette option est optionnelle, et peut être omise. Le seul nom du fichier est suffisant.
<code>-v</code>	Affiche les versions de PHP, PHP SAPI, et Zend sur le terminal. Par exemple : <pre>\$ php -v PHP 4.3.0-dev (cli), Copyright (c) 1997-2002 The PHP Group Zend Engine v1.2.1, Copyright (c) 1998-2002 Zend Technologies</pre>
<code>-c</code>	Avec cette option, vous pouvez spécifier le nom du dossier dans lequel PHP va recherche le fichier <code>php.ini</code>, ou encore vous pouvez spécifier un fichier de configuration directement (qui n'a pas obligatoirement à être appelé <code>php.ini</code>) : <pre>\$ php -c /custom/directory/ mon_script.php \$ php -c /custom/directory/custom-file.ini mon_script.php</pre>
<code>-a</code>	Exécute PHP de manière interactive.
<code>-d</code>	Cette option permet de modifier n'importe quelle directive de configuration du fichier <code>php.ini</code>. La syntaxe est : <pre>-d configuration_directive[=value]</pre> Exemples: <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" # 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) "" # 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>
<code>-e</code>	Génère des informations étendues pour le profilage et le débogage.
<code>-z</code>	Charge une extension Zend. Si seulement un fichier est fourni, PHP essaie de charger cette extension dans le dossier courant par défaut des librairie sur votre système (généralement spécifié avec <code>/etc/ld.so.conf</code> 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 uniquement les extensions dans ce dossier.
<code>-l</code>	Cette option permet de faire une vérification syntaxique sur le code PHP fourni. En cas de réussite, le message <code>No syntax errors detected in <filename></code> (Aucune d'erreur de syntaxe n'ont été détectées dans <code><nom_de_fichier></code>) est affiché sur la sortie standard, et le script shell retourne <code>0</code>. En cas d'erreur, le

	message <code>Errors parsing <filename></code> (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 . Cette option ne détecte pas les erreurs fatales (comme par exemple les fonctions non définies). Utilisez <code>-f</code> si vous voulez tester aussi ces erreurs. Note Cette option ne fonctionne pas avec l'option <code>-r</code> .
<code>-m</code>	Cette option liste les extensions PHP et Zend compilées et chargées : <pre>\$ php -m [PHP Modules] xml tokenizer standard session posix pcre overload mysql mbstring ctype [Zend Modules]</pre>
<code>-i</code>	Cette option appelle la fonction <code>phpinfo</code> , et affiche le résultat. Si PHP ne fonctionne pas correctement, il est recommandé d'utiliser la commande <code>php -i</code> et de voir si il n'y a pas d'erreurs affichées avant ou après la table d'information. N'oubliez pas que le résultat de cette option est au format HTML , et donc grand et illisible directement.
<code>-r</code>	Cette option permet l'exécution de ^{PHP} directement dans la ligne de commande. Les balises de ^{PHP} (<code><?php</code> et <code>?></code>) ne sont pas demandées, et causeront une erreur d'analyse. Note De l'attention 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 lignes de commande, réalisées par le shell. Exemple conduisant à une erreur d'analyse : <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, grâce aux 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 : \$ 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. \$ php -r '\$foo = get_defined_constants(); var_dump(\$foo);' array(370) { ["E_ERROR"]=> int(1) ["E_WARNING"]=> int(2) ["E_PARSE"]=> int(4) ["E_NOTICE"]=> int(8) ["E_CORE_ERROR"]=> [...] }</pre> Si vous utilisez un shell différent de sh/bash, vous pouvez rencontrer d'autres problèmes. N'hésitez pas à ouvrir un rapport de bugs, ou d'envoyer un mail à phpdoc@lists.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.
<code>-h</code>	Avec cette option, vous pouvez obtenir des informations sur la liste des options supporté par votre PHP, avec des explications succinctes.

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 de configuration en début de scripts, 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 PHP sous Windows, ce qui rend les scripts facilement portables. Un exemple complet est disponible ci-dessous :

Script prévu pour être exécuté en ligne de commande (script.php)

```
#!/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];
}
?>
```

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 il n'y a pas d'en-têtes HTTP qui soient affichés. 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.

7 Index des fonctions

7.1 Apache

7.1.1 Introduction

Ces fonctions sont disponibles lorsque PHP est utilisé comme module Apache 1.x.

7.1.2 Installation

Pour savoir installer PHP pour Apache 1.x, reportez-vous à [Apache section](#) , dans le chapitre d'installation.

7.1.3 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	Défaut	Modifiable	Fonction
<code>engine</code>	On	PHP_INI_ALL	Active ou désactive l'analyse des fichiers PHP
<code>child_terminate</code>	Off	PHP_INI_ALL	Spécifie si les scripts PHP peuvent réclamer la fin des processus fils en fin de requête. Reportez-vous à apache_child_terminate
<code>last_modified</code>	Off	PHP_INI_ALL	Envoie la date de modification des scripts PHP dans l'en-tête HTTP Last-Modified:
<code>xbit_hack</code>	Off	PHP_INI_ALL	Analyse avec PHP les fichiers exécutables, indépendamment de leur extension.

Voici un éclaircissement sur l'utilisation des directives de configuration.

`engine` booléen

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.

7.1.4 Types de ressources

Cette extension ne définit aucune ressource.

7.1.5 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)

- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [apache_child_terminate](#)
- [apache_lookup_uri](#)
- [apache_note](#)
- [apache_request_headers](#)
- [apache_response_headers](#)
- [apache_setenv](#)
- [ascii2ebcdic](#)
- [ebcdic2ascii](#)
- [getallheaders](#)
- [virtual](#)

7.1.7 `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. Les propriétés de la classe renvoyée sont les suivantes :

- status
- the_request
- status_line
- method
- content_type
- handler
- uri
- filename
- path_info
- args
- boundary
- no_cache
- no_local_copy
- allowed
- send_bodyct
- bytes_sent
- byterange
- clength
- unparsed_uri
- mtime
- request_time

Note

`apache_lookup_uri` ne fonctionne que lorsque ^{PHP} est installé sous la forme d'un module Apache.

7.1.8 `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. Si la fonction est appelée avec un argument, elle renvoie la valeur courante de la variable ^{note_name}. Si `apache_note` est appelée avec deux arguments, `apache_note` affecte à la note ^{note_value} la valeur ^{note_name} et `apache_note` retournera la valeur précédente de la variable ^{note_name}.

7.1.9 `apache_request_headers()` : Lit tous les en-têtes HTTP de la requête

array **apache_request_headers** (void)

[apache_request_headers](#) retourne un tableau associatif avec tous les en-têtes ^{HTTP} de la requête courante. Cela n'est possible que lorsque ^{PHP} fonctionne comme module ^{Apache}.

Note

Avant la version PHP 4.3.0, [apache_request_headers](#) était appelée [getallheaders](#). Depuis ^{PHP} 4.3.0, [getallheaders](#) est un alias de [apache_request_headers](#).

Exemple avec [apache_request_headers](#)

```
<?php
$headers = apache_request_headers();

foreach ($headers as $header => $value) {
    echo "$header: $value <br />\n";
}
?>
```

Note

Vous pouvez aussi obtenir les valeurs des variables ^{CGI} communes en les lisant dans l'environnement, ce qui fonctionne, que vous soyez ou non en module ^{Apache}. Utilisez la fonction [phpinfo](#) pour connaître la liste des variables d'environnement disponibles.

7.1.10 apache_response_headers() : Lit tous les en-têtes de réponse HTTP

array **apache_response_headers** (void)

[apache_response_headers](#) retourne un tableau avec tous les en-têtes de réponse Apache. Cette fonctionnalité n'est disponible qu'en PHP version 4.3.0 et plus récent.

Voir aussi [getallheaders](#) et [headers_sent](#).

7.1.11 apache_setenv() : Modifie une variable de ^{subprocess_env} Apache

int **apache_setenv** (string variable , string value , bool walk_to_top)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.1.12 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.

Voir aussi [ebcdic2ascii](#)

7.1.13 `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.

Voir aussi [ascii2ebcdic](#)

7.1.14 `getallheaders()` : Lit tous les en-têtes de la requête HTTP

```
array getallheaders ( void )
```

`getallheaders` est un alias de [apache_request_headers](#). Elle retourne un tableau associatif avec tous les en-têtes HTTP émis par la requête courante. Lisez le manuel de [apache_request_headers](#) pour plus de détails.

Note

En PHP 4.3.0, `getallheaders` est devenu un alias de [apache_request_headers](#). En fait, la fonction a été renommée. Cela est dû au fait qu'elle ne fonctionne que lorsque ^{PHP} est compilé comme module Apache.

Voir aussi [apache_request_headers](#).

7.1.15 `virtual()` : Effectue une sous-requête Apache

```
int virtual ( string  filename )
```

`virtual` est une fonction spécifique au serveur Apache. Elle est équivalente à la directive "`<!--#include virtual...-->`" lorsque vous utilisez le module `include` d'Apache. Cette fonction effectue une sous-requête Apache. C'est très utile lorsque vous utilisez des scripts CGI, des fichiers `.html` ou n'importe quel type de fichier qui doit être analysé par 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 les fichiers PHP, il est conseillé d'utiliser les fonctions [include](#) et [require](#). `virtual` ne peut pas être utilisé pour inclure un fichier qui est lui-même un fichier PHP.

7.2 Tableaux

7.2.1 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}.

7.2.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.2.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.2.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.2.5 Types de ressources

Cette extension ne définit aucune ressource.

7.2.6 Constantes prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

CASE_LOWER (*entier*)
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 (*entier*)
CASE_UPPER est utilisé avec [array_change_key_case](#) et sert à convertir tous les index d'un tableau en majuscule.

SORT_ASC (*entier*)
SORT_ASC trie en ordre ascendant

SORT_DESC (*entier*)
SORT_DESC trie en ordre descendant

SORT_REGULAR (*entier*)
SORT_REGULAR compare normalement les valeurs d'un tri

SORT_NUMERIC (*entier*)

SORT_NUMERIC

compare numériquement les valeurs d'un tri

SORT_STRING (*entier*)

SORT_STRING

compare alphabétiquement les valeurs d'un tri.

7.2.7 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 prédefinies](#)
- [Voir aussi](#)
- [array_change_key_case](#)
- [array_chunk](#)
- [array_combine](#)
- [array_count_values](#)
- [array_diff_assoc](#)
- [array_diff](#)
- [array_fill](#)
- [array_filter](#)
- [array_flip](#)
- [array_intersect_assoc](#)
- [array_intersect](#)
- [array_key_exists](#)
- [array_keys](#)
- [array_map](#)
- [array_merge_recursive](#)
- [array_merge](#)
- [array_multisort](#)
- [array_pad](#)
- [array_pop](#)
- [array_push](#)
- [array_rand](#)
- [array_reduce](#)
- [array_reverse](#)
- [array_search](#)
- [array_shift](#)
- [array_slice](#)
- [array_splice](#)
- [array_sum](#)
- [array_unique](#)
- [array_unshift](#)
- [array_values](#)
- [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](#)

7.2.9 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
        )
)

```

7.2.10 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 égal, 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);
?>

```

Le script ci-dessus affichera :

```

Array
(
    [vert] => avocat
    [rouge] => pomme
    [jaune] => banane
)

```

Voir aussi array_merge , array_walk et array_values .

7.2.11 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));

```

```
?>
```

Le résultat du programme ci-dessus est :

```
Array
(
    [1] => 2
    [bonjour] => 2
    [monde] => 1
)
```

Note

[array_count_values](#) a été ajoutée en ^{PHP} 4.0.

Voir aussi [count](#) , [array_unique](#) , [array_values](#) et [count_chars](#) .

7.2.12 array_diff_assoc() : Calcule la différence de deux tableaux, en prenant en compte les clés

array **array_diff_assoc** (array array1 , array array2 , array ...)

[array_diff_assoc](#) retourne un tableau contenant les valeurs du tableau ^{array1} qui ne sont présente dans aucun autre argument. Notez que les clés sont utilisées durant la comparaison, contrairement à [array_diff](#) .

Exemple avec [array_diff_assoc](#)

```
<?php
$array1 = array ("a" => "vert", "b" => "brun", "c" => "bleu", "rouge");
$array2 = array ("a" => "vert", "jaune", "rouge");
$result = array_diff_assoc ($array1, $array2);

/* Le résultat est :
Array
(
    [b] => brun
    [c] => bleu
    [0] => rouge
)
*/
?>
```

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 représentation 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](#) .

7.2.13 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. Notez que les clés sont préservées.

Exemple avec [array_diff](#)

```
<?php
$array1 = array ("a" => "vert", "rouge", "bleu", "rouge");
$array2 = array ("b" => "vert", "jaune", "rouge");
$result = array_diff ($array1, $array2);
?>
```

`$result` contient `array("bleu");` . Les valeurs multiples dans `$array1` seront toutes traitées de la même façon.

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.

Attention

Cette fonction était inutilisable en PHP 4.0.4!

Voir aussi [array_diff_assoc](#) , [array_intersect](#) et [array_intersect_assoc](#) .

7.2.14 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` .

Exemple avec [array_fill](#)

```
<?php
$a = array_fill(5, 6, 'banane');
print_r($a);
?>
```

`$a` vaut :

```
Array
(
    [5] => banane
    [6] => banane
    [7] => banane
    [8] => banane
    [9] => banane
    [10] => banane
)
```

Voir aussi [str_repeat](#) et [range](#) .

7.2.15 array_filter() : Filtre les éléments d'un tableau

array **array_filter** (array input , callback function)

[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"));
?>
```

Le programme 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.

Voir aussi [array_map](#) , [array_reduce](#) et [array_walk](#) .

7.2.16 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, et les valeurs sont les clés. [array_flip](#) ne fonctionne que sur des entiers et des chaînes, et affichera une erreur s'il détecte une clé ou une valeur de type invalide (tableau, objet, booléen, nombre à virgule flottante).

Notez bien que les valeurs de `trans` doivent être des clés valides, c'est à dire qu'elles doivent être des entiers (`integer`) ou des chaînes de caractères (`string`). Une alerte sera émise si une valeur a 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
$trans = array_flip ($strs);
$original = strtr ($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](#) .

7.2.17 array_intersect_assoc() : Calcule l'intersection de deux tableaux avec des tests sur les index

array **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. 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);

/* $result_array ressemble à ceci :

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_diff](#) et [array_intersect](#) .

7.2.18 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. Notez que les clés sont préservées.

Exemple avec [array_intersect](#)

```
<?php
$array1 = array ("a" => "vert", "rouge", "bleu");
$array2 = array ("b" => "vert", "jaune", "rouge");
$result = array_intersect ($array1, $array2);
print_r($result);
?>
```

```
$result vaut :
Array
(
    [a] => vert
    [0] => rouge
)
```

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.

Attention

Cette fonction était buguée en ^{PHP} 4.0.4!

Voir aussi [array_intersect_assoc](#), [array_diff](#) et [array_diff_assoc](#).

7.2.19 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.

Exemple avec [array_key_exists](#)

```
<?php
$search_array = array("premier" => 1, "second" => 4);
if (array_key_exists("premier", $search_array)) {
    echo "L'élément 'premier' existe dans le tableau";
}
?>
```

Note

Cette fonction s'appelait [key_exists](#) en ^{PHP} version 4.0.6.

Voir aussi [isset](#), [array_keys](#) et [in_array](#).

7.2.20 array_keys() : Retourne toutes les clés d'un tableau

```
array array_keys ( array input , mixed search_value )
```

[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.

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));
?>
```

Le résultat du script d'exemple est :

```
Array
```

```
(
[0] => 0
[1] => couleur
)
Array
(
[0] => 0
[1] => 3
[2] => 4
)
Array
(
[0] => couleur
[1] => taille
)
```

Note

[array_keys](#) a été ajoutée en ^{PHP} 4. Ci-dessous, voici une implémentation qui fonctionnera sous ^{PHP} 3 :

Implémentation de [array_keys](#) pour les utilisateurs de PHP 3

```
<?php
function array_keys ($arr, $term="") {
    $t = array();
    while (list($k,$v) = each($arr)) {
        if ($term && $v != $term) {
            continue;
        }
        $t[] = $k;
    }
    return $t;
}
?>
```

Voir aussi [array_values](#) et [array_key_exists](#) .

7.2.21 array_map() : Applique sur fonction sur des tableaux

array **array_map** (mixed 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](#) .

Exemple avec [array_map](#)

```
<?php
function cube($n) {
    return $n*$n*$n;
}
$a = array(1, 2, 3, 4, 5);
$b = array_map("cube", $a);
?>
```

Cela conduit à ce que la variable ^{\$b} contienne :

```
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);
?>
```

Cela 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);
?>
```

Le programme 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](#) et [array_walk](#) .

7.2.22 array_merge_recursive() : Combine plusieurs tableaux ensembles, récursivement

array **array_merge_recursive** (array array1 , array array2 , array ...)

[array_merge_recursive](#) rassemble tous les éléments de plusieurs tableaux ensembles, en ajoutant les éléments de l'un à la suite des éléments du précédent. [array_merge_recursive](#) retourne le tableau résultant.

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);
?>
```

Le résultat sera

```
Array
(
    [couleur] => Array
        (
            [favorie] => Array
                (
                    [0] => rouge
                    [1] => vert
                )
            [0] => rouge
        )
    [0] => 5
    [1] => 10
)
```

Voir aussi [array_merge](#) .

7.2.23 array_merge() : Fusionne plusieurs tableaux

array **array_merge** (array array1 , array array2 , array ...)

array_merge rassemble les éléments de plusieurs tableaux ensembles, 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.

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);
?>
```

La `$result` vaut :

```
Array
(
    [couleur] => vert
    [0] => 2
    [1] => 4
    [2] => a
    [3] => b
    [forme] => trapézoïde
)
```

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.

Voir aussi [array_merge_recursive](#) et [array_combine](#) .

7.2.24 array_multisort() : Tri multi-dimensionnel

bool array_multisort (array ar1 , mixed arg , mixed ... , array ...)

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 sont préservées.

Les tableaux passés en arguments 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, mais les arguments suivants peuvent être des tableaux ou une ou deux options de tri, prises dans 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`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Trier plusieurs tableaux

```
<?php
$ar1 = array ("10", 100, 100, "a");
$ar2 = array (1, 3, "2", 1);
array_multisort ($ar1, $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.

Classer un tableau multi-dimensionnel

```
<?php
$ar = array (array ("10", 100, 100, "a"), array (1, 3, "2", 1));
array_multisort($ar[0], SORT_ASC, SORT_STRING,
                $ar[1], SORT_NUMERIC, SORT_DESC);
?>
```

Dans cet exemple, après le tri, le premier tableau contient 10, 100, 100, "a" (tri alphabétique, ordre croissant); Le deuxième tableau contient 1, 3, "2", 1 (tri numérique, ordre décroissant).

7.2.25 array_pad() : Complète un tableau jusqu'à la longueur spécifiée, avec une valeur.

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.

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](#).

7.2.26 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);
?>
```

Après ceci, `$stack` n'a plus que 3 éléments: "orange", "banane" et "pomme" :

```
Array
(
    [0] => orange
    [1] => banane
    [2] => pomme
)
```

tandis que `$fruit` contient "framboise".

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 exacte de cette fonction.

Voir aussi [array_push](#), [array_shift](#) et [array_unshift](#).

7.2.27 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 passées en paramètres 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);
?>
```

`$stack` vaut alors :

```
Array
(
    [0] => orange
    [1] => banane
    [2] => pomme
    [3] => bleuet
)
```

Voir aussi `array_pop`, `array_shift` et `array_unshift`.

7.2.28 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, il n'est 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() * 100000000);
$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](#) .

7.2.29 array_reduce() : Réduit itérativement un tableau

mixed **array_reduce** (array input , callback function , int initial)

[array_reduce](#) applique itérativement la fonction `callback` 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.

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 (= 1*2*3*4*5*10), et `$d` contiendra ¹ .

Voir aussi [array_filter](#) , [array_map](#) , [array_unique](#) et [array_count_values](#) .

7.2.30 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](#).

7.2.31 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

Avant ^{PHP} 4.2.0, `array_search` retournait `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`.

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 exacte de cette fonction.

Voir aussi [array_keys](#), [array_values](#), [array_key_exists](#) et [in_array](#).

7.2.32 array_shift() : Dépile 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. 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

```
<?php
$stack = array ("orange", "banane", "pomme", "bleuet");
$fruit = array_shift ($stack);
print_r($stack);
?>
```

`$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](#) .

7.2.33 array_slice() : Extrait une portion de tableau

array **array_slice** (array array , int offset , int length)

[array_slice](#) retourne une série d'éléments du tableau `array` commençant à l'offset `offset` et représentant `length` éléments.

Si `offset` est positif, 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.

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"
?>
```

Voir aussi [array_splice](#) et [unset](#) .

7.2.34 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.

Si `offset` est positif, la série commencera à cet offset dans le tableau `input` . Si `offset` est négatif, cette série commencera à l'offset `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 `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 `offset` .

Conseil : si le tableau de remplacement 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, -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`.

7.2.35 array_sum() : Calcule la somme des valeurs du tableau

mixed **array_sum** (array array)

`array_sum` retourne la somme des valeurs du tableau, sous forme d'un entier ou d'un nombre à virgule flottante.

Exemple avec <code>array_sum</code>
<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> Le résultat de ce script sera : <pre>sum(a) = 20 sum(b) = 6.9</pre>

Note
Dans les versions de ^{PHP} antérieure à la version 4.0.6, cette fonction modifie le tableau passé en argument et convertit ses arguments en chaînes de caractères (ce qui conduisait généralement à 0).

7.2.36 array_unique() : Dédoublonne un tableau

array **array_unique** (array array)

`array_unique` prend le tableau `array` et retourne un nouveau tableau, complètement dédoublonné.

Note
Deux éléments sont considérés comme égaux si et seulement si <code>(string) \$elem1 === (string) \$elem2</code> . En

clair : lorsque la représentation en chaîne de caractères est identique.

Attention

Cette fonction était inutilisable en ^{PHP} 4.0.4!

Notez que les clés sont préservées. `array_unique` conserve la clé de la première valeur rencontrée, et ignore toutes les suivantes.

Exemple avec `array_unique`

```
<?php
$input = array ("a" => "vert", "rouge", "b" => "vert", "bleu", "rouge");
$result = array_unique ($input);
print_r($result);
?>
```

Cela va afficher :

```
Array
(
    [a] => vert
    [0] => rouge
    [1] => bleu
)
```

`array_unique` et les types de valeurs

```
<?php
$input = array (4,"4","3",4,3,"3");
$result = array_unique ($input);
var_dump($result);
?>
```

Le résultat du programme sera (^{PHP} 4.0.6) :

```
array(2) {
    [3]=>
    int(4)
    [4]=>
    int(3)
}
```

7.2.37 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 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);
?>
```

`$queue` vaut :

```
Array
(
    [0] => pomme
    [1] => framboise
    [2] => orange
    [3] => banane
)
```

Voir aussi [array_shift](#) , [array_push](#) et [array_pop](#) .

7.2.38 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));
?>
```

Cet exemple affiche :

```
Array
(
    [0] => XL
    [1] => vert
)
```

Note

[array_values](#) a été ajoutée en ^{PHP} 4. Ci-dessous, voici une implémentation pour ceux qui utilisent toujours ^{PHP} 3.

[array_values](#) pour les utilisateurs PHP 3

```
<?php
function array_values ($arr) {
    $t = array();
    while (list($k, $v) = each ($arr)) {
        $t[] = $v;
    }
    return $t;
}
?>
```

Voir aussi [array_keys](#) .

7.2.39 array_walk() : Exécute une fonction sur chacun des membres d'un tableau.

bool **array_walk** (array array , callback function , mixed userdata)

[array_walk](#) exécute la fonction `func` avec chaque élément du tableau `arr` . Les éléments sont passés en tant que premier argument de la fonction `func` . `func` doit être une fonction définie par l'utilisateur, et non pas une fonction native ^{PHP} . Vous ne pouvez pas utiliser [array_walk](#) directement avec [strtolower](#) , il faut absolument passer par une fonction utilisateur. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

Si `func` a besoin de plus d'un argument, une alerte sera générée pour chaque appel de `func` . Ces alertes sont supprimées en ajoutant le suffixe '@' avant l'appel de [array_walk](#) ou simplement en utilisant [error_reporting](#) .

Note

Si `func` doit travailler avec les véritables valeurs du tableau, spécifiez que le premier paramètre de `func` doit être passé par référence. Alors, les éléments seront directement modifiés dans le tableau.

Note

Passer les clés et `userdata` à `func` a été ajouté en ^{PHP} 4.0.

En ^{PHP} 4, `reset` doit être appelé si nécessaire, car `array_walk` ne réinitialise pas automatiquement le tableau.

Exemple avec `array_walk`

```
<?
<TT>PHP</TT>

$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";
}

array_walk ($fruits, 'test_print');
reset ($fruits);

array_walk ($fruits, 'test_alter', 'fruit');
reset ($fruits);

array_walk ($fruits, 'test_print');
?>
```

L'affichage de ce script sera :

```
d. citron<br>
a. orange<br>
b. banane<br>
c. pomme<br>
d. fruit: citron<br>
a. fruit: orange<br>
b. fruit: banane<br>
c. fruit: pomme<br>
```

Voir aussi `each` et `list` .

7.2.40 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.

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);
?>
```

qui affichera :

```
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);
?>
```

qui affichera :

```
Array
(
    [1] => Janvier
    [2] => Février
    [3] => Mars
)
```

Voir aussi [array_pad](#), [list](#) et [range](#).

7.2.41 arsort() : Trie un tableau en ordre inverse

```
void arsort ( array array , int sort_flags )
```

`arsort` trie un tableau 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.

Exemple avec `arsort`

```
<?php
$fruits = array("d"=>"papaye", "a"=>"orange", "b"=>"banane", "c"=>"ananas");
arsort ($fruits);
for (reset ($fruits); $key = key($fruits); next($fruits)) {
    echo "fruits[$key] = ".$fruits[$key]."\n";
}
?>
```

Cet exemple va afficher:

```
fruits[d] = papaye
fruits[a] = orange
fruits[b] = banane
fruits[c] = ananas
```

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 le manuel pour la fonction [sort](#).

Voir aussi [asort](#), [rsort](#), [ksort](#) et [sort](#).

7.2.42 asort() : Trie un tableau en ordre

```
void asort ( array array , int sort_flags )
```

[asort](#) trie un tableau 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.

Exemple avec asort <pre><?php \$fruits = array("d"=>"papaye", "a"=>"orange", "b"=>"banane", "c"=>"ananas"); asort(\$fruits); for(reset(\$fruits); \$key = key(\$fruits); next(\$fruits)) { echo "fruits[\$key] = ".\$fruits[\$key]."\n"; } ?></pre>

Cet exemple 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](#).

7.2.43 compact() : Crée un tableau contenant les variables et leur valeur

```
array compact ( mixed varname , mixed ... )
```

[compact](#) accepte différents paramètres. Les paramètres peuvent être des variables contenant des chaînes, ou un tableau de chaînes, qui peut contenir d'autres tableaux de noms, que [compact](#) traitera récursivement.

Pour chacun des arguments, [compact](#) recherche une variable avec une variable de 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.

Exemple avec `compact`

```
<?php
$ville = "Montréal";
$province = "Québec";
$evenement = "Conférence PHP Québec";
$location_vars = array("ville", "province");
$result = compact("evenement", $location_vars);
?>
```

Après ce programme, `$result` contiendra :

```
Array
(
    [evenement] => Conférence PHP Québec
    [ville] => Montréal
    [province] => Québec
)
```

Voir aussi `extract` .

7.2.44 count() : Compte le nombre d'éléments d'une variable

```
int count ( mixed var , int mode )
```

`count` retourne le nombre d'éléments dans `var` , qui est généralement un tableau (et tout le reste n'aura qu'un élément).

If `var` n'est pas un tableau, ¹ sera retourné (exception : `count( NULL )` égale ⁰).

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 .

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;
?>
```

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

?>
```

Note

sizeof est un alias de count .

Voir aussi is_array , isset et strlen .

7.2.45 current() : Transforme une variable en 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 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 .

7.2.46 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);
?>
```

`$bar` contient maintenant les paires suivantes :

```
Array
(
    [1] => ghislain
    [value] => ghislain
    [0] => 0
    [key] => 0
)
```

```
<?php
$foo = array( "Mehdi" => "Didou", "Damien" => "Dams");
$bar = each($foo);
print_r($bar);
?>
```

`$bar` contient maintenant les paires suivantes :

```
Array
(
    [1] => Didou
    [value] => Didou
    [0] => Mehdi
    [key] => Mehdi
)
```

`each` est typiquement utilisé en conjonction avec `list` pour passer en revue un tableau. Par exemple, avec `$_POST` :

Passer en revue `$_POST` avec `each`

```
echo "Valeurs transmises via la méthode POST :<br />\n";
reset ($_POST);
while (list ($key, $val) = each ($_POST)) {
    echo "$key => $val<br />\n";
}
```

Après chaque `each` , le pointeur de tableau est déplacé au prochain élément, ou sur le dernier élément, lorsqu'on arrive à la fin.

Voir aussi `key` , `list` , `current` , `reset` , `next` et `prev` .

7.2.47 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.

Exemple avec `end`

```
<?php
$fruits = array('pomme','banane','myrtille');
print end($fruits); // myrtille
?>
```

Voir aussi `current` , `each` , `prev` , `next` et `reset` .

7.2.48 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`, et créer une nouvelle variable.

`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`
Ecrase 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 option 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`. Le résultat préfixé n'est pas un nom de variable valide, il ne sera pas importé dans la table des symboles.

`extract` retourne le nombre de variables réellement importées dans la table des symboles.

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");
print "$couleur, $taille, $forme, $wddx_taille\n";
?>
```

L'exemple ci-dessus affiche :

```
bleu, grand, sphere, moyen
```

La variable `$taille` n'a pas été réécrite, car on avait spécifié le paramètre `EXTR_PREFIX_SAME`, qui a permis la création `$wddx_size`. Si `EXTR_SKIP` avait été utilisé, alors `$wddx_size` n'aurait pas été créé. Avec `EXTR_OVERWRITE`, `$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](#).

7.2.49 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 type du paramètre `needle` correspond au type de la valeur trouvée dans `haystack`.

Note

Si `needle` est une chaîne, la comparaison est faite en tenant compte de la casse.

Note

Dans les versions antérieure à 4.2.0, `needle` ne pouvait pas être un tableau.

Exemple avec `in_array`

```
<?php
$os = array ("Mac", "NT", "Irix", "Linux");
if (in_array ("Linux", $os)) {
    print "Trouvé Linux";
}
if (in_array ("mac", $os)) {
    print "Trouvé mac";
}
?>
```

La seconde condition échoue, car `in_array` est sensible à la casse. Le script retourne :
Trouvé Linux

Exemple avec `in_array` et le mode strict

```
<?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";
?>
```

Ce script va afficher :

```
1.13 a été trouvé avec recherche stricte
```

Exemple avec [in_array](#) et deux tableaux en paramètres

```
<?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";
?>

// Cela affichera :

'ph' a été trouvé
'o' a été trouvé
```

Voir aussi [array_search](#) , [array_key_exists](#) et [isset](#) .

7.2.50 key() : Retourne une clé d'un tableau associatif

mixed **key** (array array)

[key](#) retourne l'index de la clé courante dans un tableau.

Voir aussi [current](#) et [next](#) .

7.2.51 krsort() : Trie un tableau en sens inverse et suivant les clés

int **krsort** (array array , int sort_flags)

[krsort](#) trie un tableau 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.

Exemple avec [krsort](#)

```
<?php
$fruits = array("d"=>"papaye", "a"=>"orange", "b"=>"banane", "c"=>"ananas");
krsort($fruits);
for(reset($fruits); $key = key($fruits); next($fruits)) {
    echo "fruits[$key] = ".$fruits[$key]."\n";
}
?>
```

Cet exemple va afficher :

```
fruits[a] = orange
fruits[b] = banane
fruits[c] = ananas
fruits[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](#) .

Voir aussi [array_multisort](#) , [arsort](#) , [asort](#) , [ksort](#) , [natSORT](#) , [natcasesort](#) , [rsort](#) , [sort](#) , [uasort](#) , [uksort](#) et [usort](#) .

7.2.52 ksort() : Trie un tableau suivant les clés

int **ksort** (array array , int sort_flags)

[ksort](#) trie un tableau suivant les clés, en maintenant la correspondance entre les clés et les valeurs.

Cette fonction est pratique pour les tableaux associatifs.

Exemple avec `ksort`

```
<?php
$fruits = array("d"=>"papaye", "a"=>"orange", "b"=>"banane", "c"=>"ananas");
ksort($fruits);
reset($fruits);
while (list ($key, $val) = each ($fruits)) {
    echo "$key => $val\n";
}
?>
```

Cet exemple 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`.

Voir aussi `array-multisort`, `arsort`, `asort`, `krsort`, `natsort`, `natcasesort`, `rsort`, `sort`, `uasort`, `uksort` et `usort`.

Note

Le second paramètre a été ajouté en ^{PHP} 4.0.

7.2.53 list() : Transforme une liste de variables en tableau

void **list** (*mixed* ...)

Tout comme `array`, `list` n'est pas une véritable fonction, mais un élément de langage, qui permet d'assigner une série de variables 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) = $info;
print "$drink est $color et la $power le rend particulier.\n";

// Liste de certaines variables
list($drink, , $power) = $info;
print "$drink contient de la $power.\n";

// Ou bien, n'utilisons que le troisième
list( , , $power) = $info;
print "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)) {
```

```

    print ( " <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 utilisez 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.

7.2.54 natcasesort() : Tri d'un tableau avec l'algorithme à "ordre naturel" insensible à la casse

```
void natcasesort ( array array )
```

natcasesort implémente un algorithme de tri qui traite les chaînes alpha-numériques comme un être humain : c'est ce qui est appelé l'"ordre naturel".

natcasesort est la version insensible à la casse de natsort. Voir aussi natsort pour un exemple illustré.

Pour plus de détails, rendez-vous sur le site de Martin Pool : [Natural Order String Comparison](#).

Voir aussi sort, natsort, strnatcmp et strnatcasecmp.

7.2.55 natsort() : Tri d'un tableau avec l'algorithme à "ordre naturel"

```
void natsort ( array array )
```

natsort implémente un algorithme de tri qui traite les chaînes alpha-numériques comme un être humain : 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 :

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);
?>
```

Le code 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](http://www.martinpool.net/natsort/) .

Note

Si vous souhaitez conserver la relation clé/valeur, utilisez plutôt `usort($arr, 'strnatcmp')` .

Voir aussi [natcasesort](#) , [strnatcmp](#) et [strnatcasecmp](#) .

7.2.56 next() : Avance le pointeur interne d'un tableau

mixed **next** (array array)

`next` retourne l'élément suivant du tableau, ou `FALSE` s'il n'y a plus d'éléments. Le pointeur de interne de tableau est avancé d'un élément.

`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, il faut utiliser `each` .

Exemples avec `next`

```
<?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](#) .

7.2.57 pos() : Retourne l'élément courant d'un tableau

mixed **pos** (array array)

pos est une fonction alias de current .

Voir aussi end , next , prev et reset .

7.2.58 prev() : Recule le pointeur courant de tableau

mixed **prev** (array array)

prev repositionne le pointeur interne de tableau à la dernière place qu'il occupait, ou bien 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 il fait reculer le pointeur plutôt que de l'avancer.

Exemples avec next

```
<?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 .

7.2.59 range() : Crée un tableau contenant un intervalle d'éléments

array **range** (int low , int high , int 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. Si 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.

Voir aussi shuffle et array_fill .

7.2.60 reset() : Remet le pointeur interne de tableau au début

mixed **reset** (array array)

reset replace le pointeur de tableau ^{array} au premier élément.

reset retourne la valeur du premier élément.

Voir aussi current , each , next et prev .

7.2.61 rsort() : Trie en ordre inverse

void **rsort** (array array , int sort_flags)

rsort effectue un tri en ordre décroissant (du plus grand au plus petit).

Exemple avec rsort

```
<?php
$fruits = array("papaye", "orange", "banane", "ananas");
rsort($fruits);
for (reset($fruits); list($key,$value) = each($fruits); ) {
    echo "fruits[$key] = ", $value, "\n";
}
?>
```

Cet exemple va afficher:

```
fruits[0] = papaye
fruits[1] = orange
fruits[2] = banane
fruits[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 , sort et usort .

7.2.62 shuffle() : Mélange les éléments d'un tableau

void **shuffle** (array array)

shuffle mélange les éléments d'un tableau.

Note

Depuis ^{PHP} 4.2.0, il n'est plus besoin d'initialiser le générateur de nombres aléatoires avec srand ou mt_srand car c'est fait automatiquement.

Exemple avec shuffle

```
<?php
$numbers = range (1,20);
srand ((float)microtime()*1000000);
shuffle ($numbers);
while (list (, $number) = each ($numbers)) {
    echo "$number ";
}
?>
```

Voir aussi arsort , asort , krsort , krsort , sort et usort .

7.2.63 sizeof() : Alias de count

sizeof est un alias de count .

7.2.64 sort() : Trie un tableau

void **sort** (array array , int sort_flags)

sort trie le tableau ^{array} . Les éléments seront triés du plus petit au plus grand.

Exemple avec sort

```
<?php
$fruits = array("papaye","orange","banane","ananas");
sort ($fruits);
reset ($fruits);
while (list ($key, $val) = each ($fruits)) {
    echo "fruits[\".$key.\"] = ".$val."\n";
}
?>
```

Cet exemple 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
- SORT_NUMERIC : compare les éléments numériquement
- SORT_STRING : compare les éléments comme des chaînes de caractères

Note

Le second paramètre a été ajouté en ^{PHP} 4.Voir aussi [arsort](#) , [asort](#) , [ksort](#) , [natsort](#) , [natcasesort](#) , [rsort](#) , [usort](#) , [array_multisort](#) et [uksort](#) .

7.2.65 uasort() : Trie un tableau en utilisant une fonction de comparaison personnalisée

```
void 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.

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](#) .

7.2.66 uksort() : Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur

```
void uksort ( array array , callback cmp_function )
```

[uksort](#) trie les clés du tableau en utilisant une fonction définie par l'utilisateur. Si un tableau doit être trié avec un critère complexe, il est préférable d'utiliser [uksort](#) .

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");

while (list ($key, $value) = each ($a)) {
    echo "key: $value\n";
}
?>
```

Cet exemple affichera:

```
20: vingt
10: dix
4: quatre
3: trois
```

Voir aussi [usort](#) , [uasort](#) , [sort](#) , [asort](#) , [arsort](#) , [ksort](#) , [natsort](#) et [rsort](#) .

7.2.67 usort() : Trie un tableau en utilisant une fonction de comparaison personnalisée

```
void 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 <code>cmp_function</code> , 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.

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");

while (list ($key, $value) = each ($a)) {
    echo "$key: $value\n";
}
?>
```

Cet exemple va afficher :

```
0: 6
1: 5
2: 3
3: 2
4: 1
```

Note
Evidemment dans ce cas trivial, <code>rsort</code> 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, "cmp");

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.

Cet exemple 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)
{
    $al = strtolower($a->name);
    $bl = strtolower($b->name);
    if ($al == $bl) return 0;
    return ($al > $bl) ? +1 : -1;
}
}

$a[] = new TestObj("c");
$a[] = new TestObj("b");
$a[] = new TestObj("d");

uasort($a, array ("TestObj", "cmp_obj"));

foreach ($a as $item) {
    print $item->name."\n";
}
?>

```

Cet exemple va afficher :

```

b
c
d

```

Voir aussi [uasort](#) , [uksort](#) , [sort](#) , [asort](#) , [arsort](#) , [ksort](#) , [natsort](#) et [rsort](#) .

7.3 Aspell [Obs.]

7.3.1 Introduction

Les fonctions Aspell vous permettent de vérifier l'orthographe d'un mot, et d'offrir des suggestions de correction. Plusieurs langues sont disponibles, comme le français, l'anglais, l'allemand, le suédois ou le danois.

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](http://php.net/pspell). Elle utilise la librairie pspell et fonctionne avec les nouvelles versions de Aspell.

7.3.2 Pré-requis

Aspell fonctionne avec de très vieilles versions (jusqu'à la version .27.* ou presque) de la librairie Aspell. Ce module, et ces versions d'Aspell, sont disponibles sur <http://aspell.sourceforge.net/>.

7.3.3 Installation

En ^{PHP} 4, ces fonctions ne sont disponibles que si ^{PHP} a été compilé avec l'option `--enable-mailparse`.

7.3.4 Voir aussi

Voir aussi [pspell](#).

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Voir aussi](#)
- [aspell_check_raw](#)
- [aspell_check](#)
- [aspell_new](#)
- [aspell_suggest](#)

7.3.6 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, et ^{FALSE} sinon.

Exemple avec [aspell_check](#)

```
<?php
$aspell_link = aspell_new("english");
if (aspell_check($aspell_link,"testt")) {
    echo "L'orthographe est correcte.";
} else {
    echo "Désolé, l'orthographe est incorrecte.";
}
?>
```

7.3.7 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 dans les fonctions aspell. aspell_new retourne ^{FALSE} en cas d'erreur.

Exemple avec aspell_new

```
<?php
$aspell_link = aspell_new("english");
?>
```

7.3.8 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 formé.

Exemple avec aspell_suggest

```
<?php
$aspell_link = aspell_new("french");
if (!aspell_check($aspell_link,"testt")) {
    $suggestions=aspell_suggest($aspell_link,"testt");
    for($i=0; $i < count($suggestions); $i++) {
        echo "Orthographes envisageables : " . $suggestions[$i] . "<BR>";
    }
}
?>
```


7.4 Nombres de grande taille BCMath

7.4.1 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.

7.4.2 Pré-requis

Depuis PHP 4.0.4, libbcmath est distribuée avec ^{PHP}. Vous n'avez besoin d'aucune librairie externe pour cette extension.

7.4.3 Installation

En PHP 4, 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 librairie supplémentaire pour disposer de ces fonctions.

7.4.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>bcmath.scale</code>	0	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

```
bcmath.scale integer
```

Le nombre de décimales pour toutes les fonctions bcmath.

7.4.5 Types de ressources

Cette extension ne définit aucune ressource.

7.4.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)

- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [bcadd](#)
- [bccomp](#)
- [bcddiv](#)
- [bcmmod](#)
- [bcmul](#)
- [bcpow](#)
- [bcpowmod](#)
- [bcscale](#)
- [bcsqrt](#)
- [bcsub](#)

7.4.8 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.

7.4.9 bcddiv() : Divise deux nombres de grande taille

```
string bcddiv ( string left_operand , string right_operand , int scale )
```

[bcddiv](#) 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.

Voir aussi [bcmul](#).

7.4.10 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`.

Voir aussi [bcddiv](#).

7.4.11 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.

Voir aussi [bcdiv](#) .

7.4.12 bcpow() : Elève un nombre à la puissance n-ième.

```
string bcpow ( string x , int 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.

Voir aussi [bcpowmod](#) et [bcsqrt](#) .

7.4.13 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.

Les deux lignes suivantes sont fonctionnellement identiques. La version qui utilise [bcpowmod](#) est bien plus rapide, et accepte des paramètres plus grands.

Exemple avec [bcpowmod](#)

```
<?php
$a = bcpowmod($x,$y,$mod);
$b = bcmod(bcpow($x,$y),$mod);
/* $a and $b are equal to each other. */
?>
```

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.

Voir aussi [bcpow](#) et [bcmod](#) .

7.4.14 bcscale() : Détermine le nombre de décimales par défaut

```
string 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 $scale$.

7.4.15 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.

Voir aussi [bcpow](#) .

7.4.16 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.

Exemple avec [bcsub](#)

```
<?php
$a = 1.234;
$b = 5;

echo bcsub($a, $b);      // -3
echo bcsub($a, $b, 4);  // -3.7660

?>
```

Voir aussi [bcadd](#) .

7.5 Compression Bzip2

7.5.1 Introduction

Les fonctions bzip2 servent à lire et écrire dans les fichiers compressés avec la méthode bzip2 (.bz2). Il est possible de lire et d'écrire de tels fichiers.

7.5.2 Pré-requis

Ce module utilise les fonctions de la librairie bzip2, de Julian Seward pour écrire et lire des fichiers bzip2 (.bz2) de manière transparente.

7.5.3 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. Ce module requiert la librairie bzip2/libbzip2, version `>= 1.0.x`.

7.5.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.5.5 Types de ressources

Cette extension définit un type de ressource : un pointeur de fichier, identifiant un fichier compressé bzip2.

7.5.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.5.7 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
bzclose($bz);

// ouvre le fichier en lecture
$bz = bzopen($filename, "r");
```

```
// lit 10 caractères
print bzread($bz, 10);

// affiche tout le reste du fichier, puis le ferme
print bzread($bz);

bzclose($bz);
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [bzclose](#)
- [bzcompress](#)
- [bzdecompress](#)
- [bzerrno](#)
- [bzerror](#)
- [bzerrstr](#)
- [bzflush](#)
- [bzopen](#)
- [bzread](#)
- [bzwrite](#)

7.5.9 bzcompress() : Comprime une chaîne avec bzip2

string **bzcompress** (string source , int blocksize , int workfactor)

[bzcompress](#) compresse la chaîne `source` et retourne les données ainsi encodées.

Le paramètre optionnel `blocksize` spécifie la taille de bloc utilisée durant la compression et doit être un nombre de 1 à 9, sachant que 9 représente la meilleure compression, mais qu'elle utilise plus de ressources pour se faire. `blocksize` vaut par défaut 4.

Le paramètre optionnel `workfactor` contrôle le comportement de la compression dans les pires cas de données hautement répétitives. Cette valeur peut aller de 0 à 250 (0 est une valeur spéciale et 30 la valeur par défaut). En dehors de `workfactor`, le résultat sera le même.

Exemple avec [bzcompress](#)

```
<?php
$str = "données de test";
$bzstr = bzcompress($str, 9);
?>
```

Voir aussi [bzdecompress](#).

7.5.10 bzdecompress() : Décompresse une chaîne bzip2

string **bzdecompress** (string source , int small)

[bzdecompress](#) décompresse la chaîne `source`, en supposant qu'elle a été compressée avec bzip2, puis la

retourne. Si le paramètre optionnel `small` vaut `TRUE`, un autre algorithme de décompression sera utilisé : il consomme moins de mémoire (le maximum demandé tombe autour de 2300 ko), mais fonctionne globalement à la moitié de la vitesse. Reportez-vous à la [documentation bzip2](#) pour plus de détails sur cette fonctionnalité.

Exemple avec [bzdecompress](#)

```
<?php
$str = bzdecompress($bzstr);
?>
```

Voir aussi [bzcompress](#).

7.5.11 bzerrno() : Retourne le numéro d'erreur bzip2

int **bzerrno** (resource bz)

[bzerrno](#) retourne le numéro d'erreur du fichier bz2 représenté par le pointeur `bz`.

Voir aussi [bzerror](#) et [bzerrstr](#).

7.5.12 bzerror() : Retourne le numéro et le message d'erreur bzip2 dans un tableau

array **bzerror** (resource bz)

[bzerror](#) retourne le numéro et le message d'erreur du fichier bz2 représenté par le pointeur `bz`. [bzerror](#) retourne un tableau associatif.

Exemple avec [bzerror](#)

```
<?php
$error = bzerror($bz);
echo $error["errno"];
echo $error["errstr"];
?>
```

Voir aussi [bzerrno](#) et [bzerrstr](#).

7.5.13 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`.

Voir aussi [bzerrno](#) et [bzerror](#).

7.5.14 bzflush() : Force l'écriture de toutes les données compressées

int **bzflush** (resource bz)

[bzflush](#) vide les buffers d'écriture du fichier représenté par `bz`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [bzfread](#) et [bzfwrite](#).

7.5.15 bzopen() : Ouvre un fichier compressé avec bzip2

resource **bzopen** (string filename , string mode)

[bzopen](#) ouvre un fichier bzip2 (.bz2) en écriture ou en lecture. `filename` est le nom du fichier à ouvrir. `mode` est similaire au même paramètre de la fonction [fopen](#) ('r' pour lecture, 'w' pour écriture, etc.).

Si l'ouverture échoue, [bzopen](#) retourne `FALSE`, sinon, elle retourne un pointeur de fichier.

Exemple avec [bzopen](#)

```
<?php
$bz = bzopen("/tmp/foo.bz2", "r");
?>
```

Voir aussi [bzclose](#).

7.5.16 bzread() : Lecture binaire d'un fichier bzip2

string **bzread** (resource bz , int length)

[bzread](#) lit jusqu'à `length` octets depuis le fichier bzip2, référencé par le pointeur `bz`. La lecture s'arrête lorsque `length` octets (non compressés) ont été lus, qu'une erreur est rencontrée, ou bien que la fin du fichier a été atteinte : le premier des trois qui survient. Si le paramètre optionnel `length` est omis, [bzread](#) lit 1024 octets (non compressés) en même temps.

Exemple avec [bzread](#)

```
<?php
$bz = bzopen("/tmp/foo.bz2", "r");
$str = bzread($bz, 2048);
echo $str;
?>
```

Voir aussi [bzwrite](#) et [bzopen](#).

7.5.17 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`. Si le paramètre optionnel `length` est fourni, l'écriture sera arrêtée après l'écriture de `length` octets (non compressés), ou la fin de la chaîne (le premier qui survient).

Exemple [bzwrite](#)

```
<?php
$str = "données non compressées";
$bz = bzopen("/tmp/foo.bz2", "w");
bzwrite($bz, $str, strlen($str));
?>
```


Voir aussi [bzread](#) et [bzopen](#) .

7.6 Calendrier

7.6.1 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.

7.6.2 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 librairie supplémentaire pour disposer de ces fonctions.

7.6.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.6.4 Types de ressources

Cette extension ne définit aucune ressource.

7.6.5 Constantes prédefinies

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*)

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [cal_days_in_month](#)
- [cal_from_jd](#)
- [cal_info](#)
- [cal_to_jd](#)
- [easter_date](#)
- [easter_days](#)
- [frenchtojd](#)
- [gregoriantojd](#)
- [jddayofweek](#)
- [jdmonthname](#)
- [jdtofrench](#)
- [jdtogregorian](#)
- [jdtojewish](#)
- [jdtojulian](#)
- [jdtounix](#)
- [jewishtojd](#)
- [juliantojd](#)
- [unixtojd](#)

7.6.7 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.

Voir aussi [cal_to_jd](#).

7.6.8 cal_info() : Retourne des détails sur un calendrier

array **cal_info** (int calendar)

[cal_info](#) retourne des informations sur le calendrier ^{calendar} spécifié, ou sur tous les calendriers supportés si l'argument ^{calendar} est omis.

Les informations de calendriers sont retournés dans un tableau qui contient les éléments ^{calname}, ^{calsymbol}, ^{month}, ^{abbrevmonth} et ^{maxdaysinmonth}.

Si le paramètre `calendar` n'est pas spécifié, un tableau représentant tous les calendriers est retourné. Cette fonctionnalité sera disponible en PHP 5.

7.6.9 `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_to_id`.

7.6.10 `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. Si l'année n'est pas précisée, c'est l'année en cours qui est utilisée.

Depuis PHP 4.3.0, le paramètre `year` est optionnel, et prend la valeur de l'année en cours par défaut.

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).

Exemples avec `easter_date`

```
<?php
echo date( "M-d-Y", easter_date(1999) );      /* "04 avril 1999" */
echo date( "M-d-Y", easter_date(2000) );      /* "23 avril 2000" */
echo date( "M-d-Y", easter_date(2001) );      /* "15 avril 2001" */
?>
```

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 `easter_days` pour les calculs de date de Pâques avant 1970 et après 2037.

7.6.11 `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`.

7.6.12 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.

7.6.13 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 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

```
<?php
$jd = gregoriantojd(10,11,1970);
echo("$jd\n");
$gregorian = jdtogregorian($jd);
echo("$gregorian\n");
?>
```

7.6.14 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.

Mode	Signification
0	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).

7.6.15 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
0	Grégorien - abrégé
1	Grégorien
2	Julien - abrégé
3	Julien
4	Juif
5	Républicain français

7.6.16 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.

7.6.17 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".

7.6.18 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);
?>
```

7.6.19 jdtojulian() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien.

string **jdtojulian** (int julianday)

[jdtojulian](#) convertit le nombre de jours du calendrier Julien en une chaîne contenant la date du calendrier Julien, au format "mois/jour/année".

7.6.20 jdtounix() : Convertit un nombre de jours Julien en timestamp UNIX

int **jdtounix** (int jday)

[jdtounix](#) retourne un timestamp UNIX correspondant au nombre de 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).

Voir aussi [unixtojd](#) .

7.6.21 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.

7.6.22 `juliantojd()` : Convertit le nombre de jours du calendrier Julien en date 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.

7.6.23 `unixtojd()` : Convertit un timestamp UNIX en nombre de jours Julien

```
int unixtojd ( int timestamp )
```

`unixtojd` retourne le nombre de 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](#).

7.7 Paiement C CVS

7.7.1 Introduction

Ces fonctions font l'interface avec les API C CVS, vous permettant de travailler directement avec C CVS depuis vos scripts PHP. C CVS 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 C CVS pour PHP, vous pouvez effectuer des transactions avec les cartes de crédits, directement depuis vos scripts PHP via C CVS. La suite vous montrera comment procéder.

Note

C CVS 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.

7.7.2 Installation

Pour activer le support C CVS de PHP, commencez par vérifier votre installation C CVS. Vous devez configurer PHP avec l'option `--with-cvcs`. Si vous utilisez cette option sans spécifier le chemin de votre installation, PHP essaiera de la trouver à sa position par défaut (`/usr/local/cvcs`). Si C CVS est installé dans un autre dossier, lancez la configuration avec : `--with-cvcs=$ccvs_path`, où `$ccvs_path` est le chemin de votre installation C CVS. Notez bien que C CVS requiert que `$ccvs_path/lib` et `$ccvs_path/include` existent, et qu'ils contiennent respectivement `cv_api.h` et `libccvs.a` sous `include` et `lib`.

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 C CVS (i.e. si vous avez installé C CVS avec l'utilisateur 'ccvs', vos scripts PHP doivent tourner aussi en 'ccvs').

7.7.3 Voir aussi

RedHat a arrêté le support de C CVS. Cependant, un manuel légèrement en retard est disponible à <http://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](#)

7.7.5 ccvs_auth() : Perform credit authorization test on a transaction

```
string ccvs_auth ( string session , string invoice )
```

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.6 ccvs_command() : Performs a command which is peculiar to a single protocol, and thus is not available in the general CCVS API

```
string ccvs_command ( string session , string type , string argval )
```

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.7 ccvs_count() : Find out how many transactions of a given type are stored in the system

```
int ccvs_count ( string session , string type )
```

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.8 ccvs_delete() : Delete a transaction

```
string ccvs_delete ( string session , string invoice )
```

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.9 ccvs_done() : Terminate CCVS engine and do cleanup work

```
string ccvs_done ( string sess )
```

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.10 ccvs_init() : Initialize CCVS for use

```
string ccvs_init ( string name )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.11 ccvs_lookup() : Look up an item of a particular type in the database

```
string ccvs_lookup ( string session , string invoice , int inum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.12 ccvs_new() : Create a new, blank transaction

```
string ccvs_new ( string session , string invoice )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.13 ccvs_report() : Return the status of the background communication process

```
string ccvs_report ( string session , string type )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.14 ccvs_return() : Transfer funds from the merchant to the credit card holder

```
string ccvs_return ( string session , string invoice )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.15 ccvs_reverse() : Perform a full reversal on an already-processed authorization

```
string ccvs_reverse ( string session , string invoice )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.16 ccvs_sale() : Transfer funds from the credit card holder to the merchant

```
string ccvs_sale ( string session , string invoice )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.17 ccvs_status() : Check the status of an invoice

```
string ccvs_status ( string session , string invoice )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.18 ccvs_textvalue() : Get text return value for previous function call

```
string ccvs_textvalue ( string session )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.7.19 ccvs_void() : Perform a full reversal on a completed transaction

```
string ccvs_void ( string session , string invoice )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.8 Support COM pour Windows

7.8.1 Introduction

COM est une technologie qui permet la réutilisation de code écrit dans n'importe quel langage, en utilisant une convention d'appel standardisée, et un masquage des détails d'implémentations derrière des API (tels que le nom de la machine de stockage, ou le programme d'exécution). Elle peut être considérée comme un super mécanisme d'appel distant (Remote Procedure Call, RPC), avec des racines orientées objets. COM sépare l'implémentation de l'interface.

COM encourage le versionnage, la séparation de l'implémentation et de l'interface, et le masquage des détails d'implémentation, tels que le programme d'exécution ou le langage de programmation.

7.8.2 Pré-requis

Les fonctions COM sont uniquement disponibles sur les versions Windows de PHP. Elles ont été ajoutées à partir de PHP 4.0.

7.8.3 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 librairie supplémentaire pour disposer de ces fonctions.

7.8.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>com.allow_dcom</code>	"0"	PHP_INI_SYSTEM
<code>com.autoregister_typelib</code>	"0"	PHP_INI_SYSTEM
<code>com.autoregister_verbose</code>	"0"	PHP_INI_SYSTEM
<code>com.autoregister_casesensitive</code>	"1"	PHP_INI_SYSTEM
<code>com.typelib_file</code>	""	PHP_INI_SYSTEM

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

7.8.5 Constantes prédefinies

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*)

7.8.6 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 \(YACL\)](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes prédéfinies](#)
- [Voir aussi](#)
- [com_addrf](#)
- [com_get](#)
- [com_invoke](#)
- [com_isenum](#)
- [com_load_typelib](#)
- [com_load](#)
- [com_propget](#)
- [com_propput](#)
- [com_propset](#)
- [com_release](#)
- [com_set](#)

7.8.8 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.

7.8.9 com_invoke() : Appelle une méthode d'un composant COM

```
mixed com_invoke ( resource com_object , string function_name , mixed function_parameters )
```

com_invoke appelle la méthode `function_name` du composant COM `com_object`. com_invoke retourne `FALSE` en cas d'erreur, sinon retourne le résultat de la fonction `function_name` en cas de succès.

7.8.10 com_isenum() : Récupère un IEnumVariant

```
void com_isenum ( object com_module )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.8.11 com_load_typelib() : Charge une librairie de type (Typelib)

```
void com_load_typelib ( string typelib_name , int case_insensitive )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.8.12 com_load() : Crée une référence sur un composant COM

```
string com_load ( string module_name , string server_name , int codepage )
```

com_load crée une nouvelle référence au composant COM `module_name`, et retourne une référence dessus. com_load retourne `FALSE` en cas d'échec. Les valeurs possibles pour le paramètre `codepage` sont : `CP_ACP`, `CP_MACCP`, `CP_OEMCP`, `CP_SYMBOL`, `CP_THREAD_ACP`, `CP_UTF7` et `CP_UTF8`.

7.8.13 com_propget() : Lit la valeur d'une propriété d'un composant COM

```
mixed com_propget ( resource com_object , string property )
```

com_propget est un alias de com_get.

7.8.14 com_propput() : Modifie une propriété d'un composant COM

```
void com_propput ( resource com_object , string property , mixed value )
```

com_propput est un alias de com_set.

7.8.15 **com_propset()** : Modifie une propriété d'un composant COM

`void com_propset ( resource com_object , string property , mixed value )`

com_propset est un alias de com_propput .

7.8.16 **com_release()** : Décrémente le compteur de références

`void com_release ( void )`

com_release décrémente le compteur de références.

7.8.17 **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 `TRUE` en cas de succès, et `FALSE` sinon.

7.9 Classes et Objets

7.9.1 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).

7.9.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.9.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.9.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.9.5 Types de ressources

Cette extension ne définit aucune ressource.

7.9.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.9.7 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) {
    $arr = get_object_vars($obj);
    while (list($prop, $val) = each($arr))
        echo "\t$prop = $val\n";
}

function print_methods($obj) {
    $arr = get_class_methods(get_class($obj));
    foreach ($arr as $method)
        echo "\tfonction $method()\n";
}

function class_parentage($obj, $class) {
    global $$obj;
    if (is_subclass_of($$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";
    }
}

// instancie 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 prédefinies](#)
- [Exemples](#)
- [call_user_method_array](#)
- [call_user_method](#)
- [class_exists](#)
- [get_class_methods](#)
- [get_class_vars](#)
- [get_class](#)
- [get_declared_classes](#)
- [get_object_vars](#)
- [get_parent_class](#)
- [is_a](#)
- [is_subclass_of](#)
- [method_exists](#)

7.9.9 call_user_method() : Appelle une méthode utilisateur d'un objet

`mixed call_user_method ( string method_name , object obj , mixed parameter , mixed ... )`

`call_user_method` appelle la méthode `method_name` de l'objet `obj`. 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](#), [call_user_func](#) et [call_user_method_array](#).

7.9.10 `class_exists()` : Vérifie qu'une classe a été définie

`bool class_exists ( string class_name )`

`class_exists` retourne `TRUE` si la classe `class_name` a été définie, et `FALSE` sinon.

7.9.11 `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`.

Note

A partir de PHP 4.0.6, vous pouvez spécifier l'objet lui-même, au lieu de sa classe `class_name`. Par exemple :

```
<?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);
    }
}
$ma_classe = new maclasse();

$class_methods = get_class_methods(get_class($ma_classe));

foreach ($class_methods as $method_name) {
    echo "$method_name\n";
}
?>
```

Va afficher :

```
maclasse
myfunc1
mafunc2
```

Voir aussi [get_class_vars](#) et [get_object_vars](#)

7.9.12 `get_class_vars()` : Retourne les valeurs par défaut des attributs d'une classe.

`array get_class_vars ( string class_name )`

`get_class_vars` retourne un tableau contenant les valeurs par défaut des attributs 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;

    // 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";
}

?>
```

va afficher :

```
// Avant PHP 4.2.0
var2 : xyz
var3 : 100

// Depuis PHP 4.2.0
var1 :
var2 : xyz
var3 : 100
```

7.9.13 `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.

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();
?>
```

Le résultat sera :

```
Son nom est foo
Mon nom est foo
```

Voir aussi [gettype](#) , [get_parent_class](#) et [is_subclass_of](#)

Voir aussi [get_parent_class](#) , [gettype](#) et [is_subclass_of](#) .

7.9.14 get_declared_classes() : Liste toutes les classes définies

array **get_declared_classes** (void)

[get_declared_classes](#) retourne un tableau contenant la liste des fonctions 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`).

7.9.15 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));

?>
```

Le script 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](#)

7.9.16 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.

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();

?>
```

The output is:

```
je suis le fils de papa
Je suis aussi le papa de papa
```

Voir aussi [get_class](#) et [is_subclass_of](#) .

Voir aussi [get_class](#) et [is_subclass_of](#) .

7.9.17 `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.

Voir aussi `get_class`, `get_parent_class` et `is_subclass_of`.

7.9.18 `is_subclass_of()` : Détermine si un objet est une sous-classe

`bool is_subclass_of ( object object , string class_name )`

`is_subclass_of` retourne ^{TRUE} si l'objet ^{object} est une sous-classe de ^{class_name}, ^{FALSE} sinon.

Voir aussi `get_class` et `get_parent_class`

7.9.19 `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}.

7.10 ClibPDF

7.10.1 Introduction

L'extension ClibPDF vous permet de créer des documents PDF avec PHP. Elle est disponible en téléchargement sur le site [FastIO](#) , mais requiert l'achat d'une licence pour utilisation commerciale. 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 sont natives de ClibPDF et se retrouvent dans le module PHP, et tout comme pdflib, elles 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 pourrait 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 traduite, 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 librairie [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.

7.10.2 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 licence pour utilisation commerciale. PHP requiert que vous utilisiez ClibPDF >= 2.

7.10.3 Installation

Pour faire fonctionner 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, ajouter les options de configuration `--with-jpeg-dir[=DIR]` et `--with-tiff-dir[=DIR]` .

7.10.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.10.5 Constantes prédefinies

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)
```

7.10.6 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($cpdf, "Times Roman outlined", 50, 750);
cpdf_end_text($cpdf);
cpdf_moveto($cpdf, 50, 740);
cpdf_lineto($cpdf, 330, 740);
cpdf_stroke($cpdf);
cpdf_finalize($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 pdfclock de la distribution pdflib 2.0

```
<?php
$radius = 200;
$margin = 20;
$pagecount = 40;

$pdf = cpdf_open(0);
cpdf_set_creator($pdf, "pdf_clock.php3");
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);
?>

```

7.10.7 Voir aussi

Voir aussi la documentation de [PDFlib](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes prédefinies](#)
- [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](#)

7.10.9 cpdf_add_outline() : Ajoute un signet à la page courante

void **cpdf_add_outline** (int pdf_document , string text)

[cpdf_add_outline](#) ajoute un signet à la page courante, avec le texte `text` qui pointe sur la page courante.

Ajouter une mise en relief

```
<?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");
cpdf_output_buffer($cpdf);
cpdf_close($cpdf);
?>
```

7.10.10 cpdf_arc() : Dessine un arc de cercle

void **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` .

Le paramètre `mode` est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

Voir aussi [cpdf_circle](#) .

7.10.11 cpdf_begin_text() : Démarre une section de texte

void **cpdf_begin_text** (int pdf_document)

[cpdf_begin_text](#) démarre une section de texte. Elle 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](#) .

7.10.12 `cpdf_circle()` : Dessine un cercle

```
void 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` .

Le paramètre `mode` est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

Voir aussi [cpdf_arc](#) .

7.10.13 `cpdf_clip()` : Aligne les dessins sur le chemin courant

```
void cpdf_clip ( int pdf_document )
```

`cpdf_clip` aligne les dessins sur le chemin courant.

7.10.14 `cpdf_close()` : Ferme un fichier PDF

```
void cpdf_close ( int pdf_document )
```

`cpdf_close` ferme un fichier PDF. 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](#) .

7.10.15 `cpdf_closepath_fill_stroke()` : Remplit le chemin, dessine le bord et ferme le chemin

```
void cpdf_closepath_fill_stroke ( int pdf_document )
```

`cpdf_closepath_fill_stroke` remplit le chemin, dessine le bord et ferme le chemin.

Voir aussi [cpdf_closepath](#) , [cpdf_stroke](#) , [cpdf_fill](#) , [cpdf_setgray_fill](#) , [cpdf_setgray](#) , [cpdf_setrgbcolor_fill](#) et [cpdf_setrgbcolor](#) .

7.10.16 `cpdf_closepath_stroke()` : Ferme le fichier et dessine une ligne le long du chemin.

```
void cpdf_closepath_stroke ( int pdf_document )
```

[`cpdf\_closepath\_stroke`](#) est une combinaison de [`cpdf\_closepath`](#) et [`cpdf\_stroke`](#).

Voir aussi [`cpdf\_closepath`](#) et [`cpdf\_stroke`](#).

7.10.17 `cpdf_closepath()` : Ferme le chemin

```
void cpdf_closepath ( int pdf_document )
```

[`cpdf\_closepath`](#) ferme le chemin courant.

7.10.18 `cpdf_continue_text()` : Imprime le texte à la ligne suivante

```
void cpdf_continue_text ( int pdf_document , string text )
```

[`cpdf\_continue\_text`](#) imprime le texte `text` à la ligne suivante.

Voir aussi [`cpdf\_show\_xy`](#), [`cpdf\_text`](#), [`cpdf\_set\_leading`](#) et [`cpdf\_set\_text\_pos`](#).

7.10.19 `cpdf_curveto()` : Dessine une courbe

```
void 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 (x^3 , y^3), en utilisant les points de contrôle (x^1 , y^1) et (x^2 , y^2).

Le paramètre `mode` est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

Voir aussi [`cpdf\_moveto`](#), [`cpdf\_rmoveto`](#), [`cpdf\_rlineto`](#) et [`cpdf\_lineto`](#).

7.10.20 `cpdf_end_text()` : Termine une section de texte

```
void cpdf_end_text ( int pdf_document )
```

[`cpdf\_end\_text`](#) termine une section de texte, commencée avec [`cpdf\_begin\_text`](#).

Affichage de texte avec <code>cpdf_end_text</code> <pre><?php cpdf_begin_text(\$pdf); cpdf_set_font(\$pdf, 16, "Helvetica", "WinAnsiEncoding"); cpdf_text(\$pdf, 100, 100, "Du texte"); cpdf_end_text(\$pdf) ?></pre>

Voir aussi [`cpdf\_begin\_text`](#).

7.10.21 `cpdf_fill_stroke()` : Remplit le chemin, et dessine le bord

```
void cpdf_fill_stroke ( int pdf_document )
```

[cpdf_fill_stroke](#) remplit l'intérieur du chemin avec la couleur courante, et dessine le chemin.

Voir aussi [cpdf_closepath](#) , [cpdf_stroke](#) , [cpdf_fill](#) , [cpdf_setgray_fill](#) , [cpdf_setgray](#) , [cpdf_setrgbcolor_fill](#) et [cpdf_setrgbcolor](#) .

7.10.22 cpdf_fill() : Remplit le chemin courant

```
void cpdf_fill ( int pdf_document )
```

[cpdf_fill](#) remplit l'intérieur du chemin courant avec la couleur courante.

Voir aussi [cpdf_closepath](#) , [cpdf_stroke](#) , [cpdf_setgray_fill](#) , [cpdf_setgray](#) , [cpdf_setrgbcolor_fill](#) et [cpdf_setrgbcolor](#) .

7.10.23 cpdf_finalize_page() : Termine une page

```
void cpdf_finalize_page ( int pdf_document , int page_number )
```

[cpdf_finalize_page](#) termine la page de numéro `page_number` . [cpdf_finalize_page](#) ne fait qu'une sauvegarde mémoire. Les pages terminées prennent moins de place, mais ne peuvent plus être modifiées.

Voir aussi [cpdf_page_init](#) .

7.10.24 cpdf_finalize() : Termine un document

```
void cpdf_finalize ( int pdf_document )
```

[cpdf_finalize](#) termine un document. Vous devez toujours appeler [cpdf_close](#) après.

Voir aussi [cpdf_close](#) .

7.10.25 cpdf_global_set_document_limits() : Fixe les limites d'un document PDF

```
void 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. [cpdf_global_set_document_limits](#) doit être appelé avant [cpdf_open](#) pour être effective. Elle fixe les limites de tous les documents ouverts après.

Voir aussi [cpdf_open](#) .

7.10.26 cpdf_import_jpeg() : Ouvre une image JPEG

```
int 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 mode )
```

[cpdf_import_jpeg](#) ouvre une image JPG, enregistré 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.

Le paramètre ^{mode} est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

Voir aussi [cpdf_place_inline_image](#) .

7.10.27 cpdf_lineto() : Dessine une ligne

```
void 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}).

Le paramètre ^{mode} est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

Voir aussi [cpdf_moveto](#) , [cpdf_rmoveto](#) et [cpdf_curveto](#) .

7.10.28 cpdf_moveto() : Fixe le point courant

```
void 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}).

Le paramètre ^{mode} est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

7.10.29 cpdf_newpath() : Commence un nouveau chemin

```
void cpdf_newpath ( int pdf_document )
```

[cpdf_newpath](#) commence un nouveau chemin dans le document ^{pdf_document} .

7.10.30 cpdf_open() : Ouvre un nouveau document PDF

```
int cpdf_open ( int compression , string filename )
```

[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 librairie 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](#).

7.10.31 cpdf_output_buffer() : Affiche le document PDF qui est en mémoire

```
void 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.

Voir aussi [cpdf_open](#).

7.10.32 cpdf_page_init() : Commence une nouvelle page

```
void 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`. `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](#).

7.10.33 cpdf_place_inline_image() : Place une image dans une page

```
void cpdf_place_inline_image ( int pdf_document , int image , float x-coor , float y-coor , float angle , float width , float height , 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 sera aussi modifiée avec les paramètres `width` et `height`.

Le paramètre optionnel `mode` détermine l'unité de longueur. Si c'est 0 ou que le paramètre est omis, la valeur par défaut de l'unité de mesure de la page sera utilisée. Sinon, les coordonnées sont mesurées en points postscripts, sans tenir compte de l'unité courante.

Voir aussi [cpdf_import_image](#).

7.10.34 cpdf_rect() : Dessine un rectangle

```
void 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`.

Le paramètre optionnel ^{mode} détermine l'unité de longueur. Si c'est 0 ou que le paramètre est omis, la valeur par défaut de l'unité de mesure de la page sera utilisée. Sinon, les coordonnées sont mesurées en points postscripts, sans tenir compte de l'unité courante.

7.10.35 `cpdf_restore()` : Restaure un environnement

void **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.

Sauver/Restaurer l'environnement CPDF

```
<?php
cpdf_save($pdf);
// plein de transformations, translations, ...
cpdf_restore($pdf)
?>
```

Voir aussi `cpdf_save`.

7.10.36 `cpdf_rlineto()` : Dessine une ligne, relativement

void **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}).

Le paramètre ^{mode} est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

Voir aussi `cpdf_moveto`, `cpdf_rmoveto` et `cpdf_curveto`.

7.10.37 `cpdf_rmoveto()` : Fixe le point courant relativement

void **cpdf_rmoveto** (int pdf_document , float x-coor , float y-coor , int mode)

`cpdf_rmoveto` fixe le point courant aux coordonnées (^{x-coor} , ^{y-coor}), relativement.

Le paramètre ^{mode} est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

Voir aussi `cpdf_moveto`.

7.10.38 `cpdf_rotate_text()` : Configure l'inclinaison d'un texte

void **cpdf_rotate_text** (int pdfdoc , float angle)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.10.39 `cpdf_rotate()` : Effectue une rotation

```
void cpdf_rotate ( int pdf_document , float angle )
```

`cpdf_rotate` effectue une rotation, d'un angle de `angle` degrés.

7.10.40 `cpdf_save_to_file()` : Ecrit un document PDF dans un fichier

```
void 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. `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`.

7.10.41 `cpdf_save()` : Sauve l'environnement courant

```
void 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.

Voir aussi `cpdf_restore`.

7.10.42 `cpdf_scale()` : Modifie l'échelle

```
void cpdf_scale ( int pdf_document , float x-scale , float y-scale )
```

`cpdf_scale` modifie l'échelle dans les deux directions.

7.10.43 `cpdf_set_action_url()` : Paramètre un hyper lien

```
void 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.

7.10.44 `cpdf_set_char_spacing()` : Fixe l'espacement des caractères

```
void cpdf_set_char_spacing ( int pdf_document , float space )
```

`cpdf_set_char_spacing` fixe l'espacement des caractères à `space`.

Voir aussi `cpdf_set_word_spacing` et `cpdf_set_leading`.

7.10.45 `cpdf_set_creator()` : Fixe le créateur d'un document PDF

```
void cpdf_set_creator ( string creator )
```

`cpdf_set_creator` fixe le créateur d'un document PDF.

Voir aussi `cpdf_set_subject`, `cpdf_set_title` et `cpdf_set_keywords`.

7.10.46 `cpdf_set_current_page()` : Fixe la page courante

```
void cpdf_set_current_page ( int pdf_document , int page_number )
```

`cpdf_set_current_page` fixe la page courante, où toutes les prochaines opérations vont avoir lieu. On peut changer de page jusqu'à ce qu'une page soit terminée avec `cpdf_finalize_page`.

Voir aussi `cpdf_finalize_page`.

7.10.47 `cpdf_set_font_directories()` : Précise les dossiers à fouiller pour rechercher des polices externes

```
void 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.

7.10.48 `cpdf_set_font_map_file()` : Configure le fichier de remplacement de polices externes

```
void cpdf_set_font_map_file ( int pdfdoc , string filename )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.10.49 `cpdf_set_font()` : Sélectionne la police courante et sa taille

```
void cpdf_set_font ( int pdf_document , string font_name , float size , string encoding )
```

`cpdf_set_font` sélectionne la police courante, sa taille et l'encodage. Actuellement, seules les polices postscript sont supportées.

Le dernier paramètre `encoding` peut prendre les valeurs suivantes : "MacRomanEncoding", "MacExpertEncoding", "WinAnsiEncoding", et " NULL ". " NULL " signifie qu'il faut utiliser l'encodage par défaut.

Reportez vous à la doc de ClibPDF, pour plus d'informations, notamment sur les polices asiatiques.

7.10.50 `cpdf_set_horiz_scaling()` : Fixe l'échelle horizontale du texte

```
void cpdf_set_horiz_scaling ( int pdf_document , float scale )
```

`cpdf_set_horiz_scaling` fixe l'échelle horizontale du texte à `scale` %.

7.10.51 `cpdf_set_keywords()` : Fixe les mot clés d'un document PDF

```
void cpdf_set_keywords ( string keywords )
```

`cpdf_set_keywords` fixe les mots-clé d'un document PDF.

Voir aussi `cpdf_set_title` , `cpdf_set_creator` et `cpdf_set_subject` .

7.10.52 `cpdf_set_leading()` : Fixe la distance entre deux lignes

```
void 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` .

Voir aussi `cpdf_continue_text` .

7.10.53 `cpdf_set_page_animation()` : Fixe l'animation de la transition entre les pages

```
void cpdf_set_page_animation ( int pdf_document , int transition , float duration )
```

`cpdf_set_page_animation` fixe l'animation de la transition entre les pages.

La valeur du paramètre de transition `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.

7.10.54 `cpdf_set_subject()` : Fixe le sujet d'un document PDF

```
void cpdf_set_subject ( string subject )
```

`cpdf_set_subject` fixe le sujet d'un document PDF.

Voir aussi `cpdf_set_title` , `cpdf_set_creator` et `cpdf_set_keywords` .

7.10.55 `cpdf_set_text_matrix()` : Fixe la matrice du texte

```
void cpdf_set_text_matrix ( int pdf_document , array matrix )
```

`cpdf_set_text_matrix` fixe la matrice du texte, qui décrit la transformation appliquée à police.

7.10.56 `cpdf_set_text_pos()` : Fixe la position du texte

```
void 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` .

Le paramètre `mode` est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisée.

Voir aussi `cpdf_show` et `cpdf_text` .

7.10.57 `cpdf_set_text_rendering()` : Détermine le rendu du texte

```
void cpdf_set_text_rendering ( int pdf_document , int mode )
```

`cpdf_set_text_rendering` détermine le rendu du texte.

Les valeurs possibles pour `mode` 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=et ajouté au chemin.

7.10.58 `cpdf_set_text_rise()` : Fixe l'élévation du texte

```
void cpdf_set_text_rise ( int pdf_document , float value )
```

`cpdf_set_text_rise` fixe l'élévation du texte à `value` unités.

7.10.59 `cpdf_set_title()` : Fixe le titre d'un document PDF

```
void cpdf_set_title ( string title )
```

`cpdf_set_title` fixe le titre d'un document PDF.

Voir aussi `cpdf_set_subject` , `cpdf_set_creator` et `cpdf_set_keywords` .

7.10.60 `cpdf_set_viewer_preferences()` : Indique comment afficher le document à l'utilisateur

```
void cpdf_set_viewer_preferences ( int pdfdoc , array preferences )
```

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.10.61 `cpdf_set_word_spacing()` : Fixe l'espacement des mots

```
void cpdf_set_word_spacing ( int pdf_document , float space )
```

`cpdf_set_word_spacing` fixe l'espacement des caractères.

Voir aussi `cpdf_set_char_spacing` et `cpdf_set_leading` .

7.10.62 `cpdf_setdash()` : Fixe le motif de pointillé

```
void cpdf_setdash ( int pdf_document , float white , float black )
```

`cpdf_setdash` fixe le motif de pointillé à ^{white} unité de blanc et ^{black} unités de noir. Si les deux sont à 0, une ligne pleine est affichée.

7.10.63 `cpdf_setflat()` : Fixe la platitude (flatness)

```
void cpdf_setflat ( int pdf_document , float value )
```

`cpdf_setflat` fixe la platitude (flatness), entre 0 et 100.

7.10.64 `cpdf_setgray_fill()` : Modifie le niveau de gris comme couleur de remplissage.

```
void cpdf_setgray_fill ( int pdf_document , float value )
```

`cpdf_setgray_fill` remplace le niveau de gris, couleur de remplissage courante, par ^{value} .

Voir aussi `cpdf_setrgbcolor_fill` .

7.10.65 `cpdf_setgray_stroke()` : Choisit un niveau de gris comme couleur de dessin

```
void cpdf_setgray_stroke ( int pdf_document , float gray_value )
```

`cpdf_setgray_stroke` remplace le niveau de gris, couleur de dessin courante, par ^{value} .

Voir aussi `cpdf_setrgbcolor_stroke` .

7.10.66 `cpdf_setgray()` : Modifie un niveau de gris comme couleur de dessin et de remplissage.

```
void cpdf_setgray ( int pdf_document , float gray_value )
```

`cpdf_setgray` remplace le niveau de gris, couleur de dessin et de remplissage, par ^{value} .

Voir aussi `cpdf_setrgbcolor_stroke` et `cpdf_setrgbcolor_fill` .

7.10.67 `cpdf_setlinecap()` : Fixe le paramètre linecap

```
void cpdf_setlinecap ( int pdf_document , int value )
```

`cpdf_setlinecap` fixe le paramètre linecap à une valeur `value` entre 0 et 2. 0 = butt end, 1 = round, 2 = projecting square.

7.10.68 `cpdf_setlinejoin()` : Fixe le paramètre linejoin

```
void 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.

7.10.69 `cpdf_setlinewidth()` : Fixe la largeur de ligne

```
void cpdf_setlinewidth ( int pdf_document , float width )
```

`cpdf_setlinewidth` fixe la largeur de ligne à la valeur de `width` .

7.10.70 `cpdf_setmiterlimit()` : Fixe le paramètre miter limit

```
void cpdf_setmiterlimit ( int pdf_document , float value )
```

`cpdf_setmiterlimit` fixe le paramètre "miter limit" à une valeur supérieure ou égale à 1.

7.10.71 `cpdf_setrgbcolor_fill()` : Choisit une couleur rgb comme couleur de remplissage

```
void 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`).

Voir aussi `cpdf_setrgbcolor_stroke` et `cpdf_setrgbcolor` .

7.10.72 `cpdf_setrgbcolor_stroke()` : Choisit une couleur rgb comme couleur de dessin

```
void 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`).

Voir aussi `cpdf_setrgbcolor_fill` et `cpdf_setrgbcolor` .

7.10.73 `cpdf_setrgbcolor()` : Choisit une couleur rgb comme couleur de dessin et de remplissage.

```
void 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`).

Voir aussi `cpdf_setrgbcolor stroke` et `cpdf_setrgbcolor fill` .

7.10.74 `cpdf_show_xy()` : Affiche un texte à une position

```
void 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`). Le dernier paramètre optionnel est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisé.

Note
The function <code>cpdf_show_xy</code> est identique à <code>cpdf_text</code> sans les options.

Voir aussi `cpdf_text` .

7.10.75 `cpdf_show()` : Imprime un texte à la position courante

```
void cpdf_show ( int pdf_document , string text )
```

`cpdf_show` imprime la chaîne `text` , à la position courante.

Voir aussi `cpdf_text` , `cpdf_begin_text` et `cpdf_end_text` .

7.10.76 `cpdf_stringwidth()` : Retourne la taille de la chaîne

```
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` .

7.10.77 `cpdf_stroke()` : Dessine une ligne le long du chemin

```
void cpdf_stroke ( int pdf_document )
```

`cpdf_stroke` dessine une ligne le long du chemin.

Voir aussi `cpdf_closepath` et `cpdf_closepath stroke` .

7.10.78 `cpdf_text()` : Imprime un texte avec des options

```
void cpdf_text ( int pdf_document , string text , float x-coor , float y-coor , int mode , float orientation , int alignmode )
```

`cpdf_text` imprime le texte `text` à la position de coordonnées (`x-coor` , `y-coor`). Le paramètre `mode` est une unité de longueur. Si il prend la valeur de 0 (ou si il est omis), c'est la valeur par défaut (72) qui est utilisé.

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 doc de ClibPDF, pour les valeurs possibles.

Voir aussi [cpdf_show_xy](#) .

7.10.79 `cpdf_translate()` : Modifie l'origine du système de coordonnées.

```
void cpdf_translate ( int pdf_document , float x-coor , float y-coor , int mode )
```

`cpdf_translate` modifie l'origine du système de coordonnées en plaçant l'origine aux coordonnées (`x-coor` , `y-coor`).

Le paramètre `mode` est une unité de longueur. S'il prend la valeur de 0 (ou s'il est omis), c'est la valeur par défaut (72) qui est utilisé.

7.11 Crack

7.11.1 Introduction

Ces fonctions vous permettent d'utiliser la librairie CrackLib, pour tester la robustesse de vos mots de passe. Cette librairie 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.

7.11.2 Pré-requis

Plus d'informations concernant la librairie CrackLib sont disponibles à <http://www.crypticide.org/users/alecm/>.

7.11.3 Installation

Pour utiliser cette fonction, vous devez compiler PHP avec l'extension Crack, en utilisant l'option de compilation `--with-crack[=DIR]`.

7.11.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>crack.default_dictionary</code>	NULL	PHP_INI_SYSTEM

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

7.11.5 Types de ressources

Cette extension ne définit aucune ressource.

7.11.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.11.7 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');</pre>

```
// Lecture du diagnostic
$diag = crack_getlastmessage();
echo $diag; // 'strong password'

// Fermeture du dictionnaire
crack_closedict($dictionary);
?>
```

Note

Si `crack_check` retourne `TRUE`, `crack_getlastmessage` retournera le message 'strong password' (mot de passe robuste).

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [crack_check](#)
- [crack_closedict](#)
- [crack_getlastmessage](#)
- [crack_opendict](#)

7.11.9 crack_closedict() : Referme le dictionnaire CrackLib

`bool crack_closedict ( resource dictionary )`

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`crack_closedict` ferme le dictionnaire `dictionary`. Si `dictionary` n'est pas spécifié, le dictionnaire courant est fermé.

7.11.10 crack_getlastmessage() : Retourne le message de diagnostic

`string crack_getlastmessage ( 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.

`crack_getlastmessage` retourne le diagnostic obtenu lors de la dernière vérification.

7.11.11 crack_opendict() : Ouvre un nouveau dictionnaire CrackLib dictionary

resource **crack_opendict** (string dictionary)

crack_opendict retourne une ressource représentant un dictionnaire, en cas de succès, et ^{FALSE} sinon.

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

crack_opendict ouvre le dictionnaire CrackLib ^{dictionary} , afin de pouvoir l'utiliser avec crack_check .

Note

Un seul dictionnaire peut être utilisé en même temps.

Voir aussi crack_check et crack_closedict .

7.12 CURL

7.12.1 Introduction

PHP supporte libcurl, une librairie 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 chargement par formulaire HTTP, les proxies, les cookies et l'authentification par mot de passe et nom de compte.

Ces fonctions ont été ajoutée en PHP 4.0.2.

7.12.2 Pré-requis

Pour pouvoir utiliser les fonctions CURL, vous devez installer le package CURL. PHP requiert la version CURL 7.0.2-beta ou plus récente. PHP ne fonctionnera pas avec une version inférieure à la version 7.0.2-beta. A partir de PHP 4.2.3, vous aurez besoin de la version 7.9.0 ou plus récent.

7.12.3 Installation

Pour utiliser CURL depuis les scripts PHP, vous devez aussi compiler PHP 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 librairies `libeay32.dll` et `ssleay32.dll` depuis le dossier `DLL` de PHP/Win32 dans le dossier système `SYSTEM32` de votre machine (par exemple : `C:\WINNT\SYSTEM32` ou `C:\WINDOWS\SYSTEM`).

7.12.4 Constantes prédefinies

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_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_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_COOKIEJAR (*entier*)
CURLOPT_SSL_CIPHER_LIST (*entier*)
CURLOPT_BINARYTRANSFER (*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*)
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*)

7.12.5 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_exec](#) et de finir votre session avec [curl_close](#). Voici un exemple d'utilisation des fonctions CURL, qui récupère la page principale de PHP :

Utilisation de CURL et PHP pour récupérer une page

```

<?php
$ch = curl_init ("http://www.nexen.net/");
$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 prédefinies](#)
- [Exemples](#)
- [curl_close](#)
- [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](#)

7.12.7 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](#).

7.12.8 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](#).

7.12.9 curl_exec() : Exécute une session CURL

`bool 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 ^{ch}.

Array

7.12.10 curl_getinfo() : Lit les informations détaillant un transfert CURL

`string 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_encode`
- `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`

7.12.11 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 récupération d'une page web

```
<?php
$ch = curl_init();

curl_setopt ($ch, CURLOPT_URL, "http://www.exemple.com/");
curl_setopt ($ch, CURLOPT_HEADER, 0);

curl_exec ($ch);
```

```
curl_close ($ch);
?>
```

Voir aussi [curl_close](#) et [curl_setopt](#) .

7.12.12 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](#) .

7.12.13 curl_multi_close() : Termine un jeu de transferts 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](#) .

7.12.14 curl_multi_exec() : Exécute les sous-requêtes d'un cURL

```
int curl_multi_exec ( resource mh )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Voir aussi [curl_multi_init](#) et [curl_exec](#) .

7.12.15 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](#) .

7.12.16 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](#) .

7.12.17 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](#) .

7.12.18 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](#) .

7.12.19 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](#) .

7.12.20 curl_setopt() : Modifie une option de transfert CURL

bool **curl_setopt** (resource ch , string 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 de type "long" pour les options suivantes (spécifiée par `option`) :

- `curl_opt_infilesize` : Lorsque vous téléchargez un fichier sur un site distant, cette option sert à indiquer à ^{PHP} la taille maximale du fichier attendu.
- `curl_opt_verbose` : Choisissez une valeur non nulle pour que CURL vous affiche tous les événements.
- `curl_opt_header` : Choisissez une valeur non nulle pour que CURL inclut l'en-tête dans la valeur de retour.

- `curl_opt_noprogress` : Choisissez une valeur non nulle pour que ^{PHP} n'affiche pas l'état des transferts CURL.

Note

^{PHP} choisit automatiquement une valeur non nulle. Ne changez cette valeur que le temps du débogage.

- `curl_opt_nobody` : Choisissez une valeur non nulle pour que le corps du transfert ne soit pas inclus dans la valeur de retour.
- `curl_opt_failonerror` : Choisissez une valeur non nulle 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.
- `curl_opt_upload` : Choisissez une valeur non nulle pour que ^{PHP} prépare un chargement.
- `curl_opt_post` : Choisissez une valeur non nulle pour que ^{PHP} fasse un HTTP POST. Un POST est un encodage normal "application/x-www-form-urlencoded", utilisé couramment par les formulaires HTML.
- `curl_opt_ftplistonly` : Choisissez une valeur non nulle pour que ^{PHP} ne fasse que lister les noms d'un dossier FTP.
- `curl_opt_ftppappend` : Choisissez une valeur non nulle pour que ^{PHP} concatène le fichier distant, plutôt que de l'écraser.
- `curl_opt_netrc` : Choisissez une valeur non nulle 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.
- `curl_opt_followlocation` : Choisissez une valeur non nulle 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).
- `curl_opt_put` : Choisissez une valeur non nulle pour que pour chargement se fasse par HTTP PUT. Le fichier à charger doit être fixé avec les options `CURLOPT_INFILE` et `CURLOPT_INFILESIZE`.
- `curl_opt_mute` : Choisissez une valeur non nulle pour que ^{PHP} soit totalement silencieux concernant toutes les fonctions CURL.
- `curl_opt_timeout` : Passez un entier "long" comme paramètre qui représente le temps maximum d'exécution de la fonction CURL.
- `curl_opt_low_speed_limit` : Passez un entier long qui représente la vitesse minimale en octets par secondes en dessous de laquelle, et pendant `CURLOPT_LOW_SPEED` secondes, ^{PHP} considérera qu'elle est trop lente, et annulera le transfert.
- `curl_opt_low_speed_time` : Passez un entier "long" qui représente le temps en secondes, qui, si la vitesse de transfert reste en dessous de `CURLOPT_LOW_SPEED_LIMIT`, ^{PHP} considérera que la connexion est trop lente, et l'annulera.
- `curl_opt_resume_from` : Passez un entier "long", qui représente l'offset, en octets, à partir duquel vous voulez commencer le transfert.
- `curl_opt_sslversion` : Passez un entier "long" qui contient la version de SSL (2 ou 3) à utiliser. Par défaut, ^{PHP} essaiera de le déterminer par lui-même, bien que dans certains cas, il vous faudra le faire manuellement.
- `curl_opt_timecondition` : Passez un entier "long" qui définit comment `CURLOPT_TIMEVALUE` est utilisé. Vous pouvez choisir entre les valeurs `TIMECOND_IFMODSINCE` ou `TIMECOND_ISUNMODSINCE`. C'est une fonctionnalité HTTP.
- `curl_opt_timevalue` : Passez un entier "long" qui représente le temps en secondes depuis le 1er janvier 1970. Cette valeur sera utilisée comme spécifié dans l'option `CURLOPT_TIMEVALUE`. Par défaut, `TIMECOND_IFMODSINCE` sera utilisé.

`value` doit être une chaîne de caractères pour les valeurs suivantes de `option`

- `curl_opt_url` : L'URL que ^{PHP} va récupérer. Vous pouvez aussi choisir cette valeur lors de l'appel à `curl_init`.
- `curl_opt_userpwd` : Passez une chaîne de caractères au format [nom]:[mot de passe], pour que ^{PHP} l'utilise lors de la connexion.
- `curl_opt_proxyuserpwd` : Passez une chaîne de caractères au format [nom]:[mot de passe], pour que ^{PHP} l'utilise lors de la connexion à un proxy HTTP.

- `curl_opt_range` : Passez une chaîne de caractères qui représente la plage de valeur que vous désirez. Elle est au format "X-Y", où les valeurs de X ou Y peuvent être omises. Le transfert HTTP supporte aussi plusieurs intervalles, séparé par des virgules : X-Y,N-M.
- `curl_opt_postfields` : Passez une chaîne de caractères qui contient toutes les données à passer lors d'une opération de HTTP POST.
- `curl_opt_referer` : Passez une chaîne de caractères qui contient l'en-tête de "REFERER", utilisé lors d'une requête HTTP.
- `curl_opt_useragent` : Passez une chaîne de caractères qui contient l'en-tête "user-agent" utilisé dans une requête HTTP.
- `curl_opt_ftpport` : Passez une chaîne de caractères qui désignera l'adresse IP utilisée pour l'instruction FTP "PORT". L'instruction POST indique au serveur distant de se connecter 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.
- `curl_opt_cookie` : Passez une chaîne de caractères qui contiendra le contenu du cookie, à transmettre dans l'en-tête HTTP.
- `curl_opt_sslcert` : Passez une chaîne de caractères qui contiendra le nom de fichier du certificat, au format PEM.
- `curl_opt_sslcertpasswd` : Passez une chaîne de caractères qui contient le mot de passe nécessaire pour utiliser le certificat `CURLOPT_SSLCERT`.
- `curl_opt_cookiefile` : Passez une chaîne de caractères qui contiendra le nom du fichier contenant les données de cookie. Le fichier de cookie peut être au format Netscape, ou simplement des en-têtes HTTP écrites dans un fichier.
- `curl_opt_customrequest` : Passez une chaîne de caractères qui sera utilisé 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.

Note

N'utilisez pas cette commande sans vous assurer que le serveur l'accepte.

Les options suivantes requièrent un pointeur de fichier, qui est obtenu avec la fonction [fopen](#) :

- `curl_opt_file` : Le fichier de sortie de votre transfert. Par défaut, STDOUT.
- `curl_opt_infile` : Le fichier d'entrée de votre transfert.
- `curl_opt_writeheader` : Le fichier de destination de l'en-tête de la sortie du transfert.
- `curl_opt_stderr` : Le fichier d'erreurs.

7.12.21 `curl_version()` : Retourne la version courante de CURL

```
string curl_version ( void )
```

[curl_version](#) retourne une chaîne avec la version courante de la librairie CURL.

7.13 Paiement Cybercash

7.13.1 Installation

Ces fonctions ne sont disponibles que si PHP a été compilé avec l'option `--with-cybercash=[DIR]` . Ces fonctions ont été ajoutées dans PHP 4.

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

7.13.3 cybercash_base64_encode() :

```
string cybercash_base64_encode ( string inbuff )
```

7.13.4 cybercash_decr() :

```
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).

7.13.5 cybercash_encr() :

```
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).

7.14 Cyrus IMAP

7.14.1 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 plate-formes Windows.

7.14.2 Installation

Pour activer le support de Cyrus IMAP et utiliser ces fonctions, vous devez compiler PHP avec l'option `--with-cyrus`.

7.14.3 Constantes prédefinies

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 prédefinies](#)
- [cyrus_authenticate](#)
- [cyrus_bind](#)
- [cyrus_close](#)
- [cyrus_connect](#)
- [cyrus_query](#)
- [cyrus_unbind](#)

7.14.5 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.

7.14.6 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.

7.14.7 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.

7.14.8 cyrus_query() : Envoie une requête à un serveur Cyrus IMAP

```
bool cyrus_query ( resource connection , string query )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.14.9 cyrus_unbind() : Supprime une fonction de callback sur une connexion Cyrus IMAP

```
bool cyrus_unbind ( resource connection , string trigger_name )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.15 Caractères

7.15.1 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.

Appelées avec un argument de type entier, ces fonctions se comportent exactement comme le équivalent en langage C.

Appelées avec un argument de type chaîne, elles vérifieront chaque caractère de la chaîne, et ne retourneront ^{TRUE} que si chaque caractère de la chaîne satisfait les critères requis.

Tout autre type d'argument (autre que chaîne ou entier) génère une erreur, et retourne ^{FALSE} immédiatement.

7.15.2 Pré-requis

Aucun pré-requis pour utiliser cette extension, hormis la librairie C standard, qui est toujours disponible.

7.15.3 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 librairie supplémentaire pour disposer de ces fonctions.

7.15.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.15.5 Types de ressources

Cette extension ne définit aucune ressource.

7.15.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)

- [Types de ressources](#)
- [Constantes prédefinies](#)
- [ctype_alnum](#)
- [ctype_alpha](#)
- [ctype_cntrl](#)
- [ctype_digit](#)
- [ctype_graph](#)
- [ctype_lower](#)
- [ctype_print](#)
- [ctype_punct](#)
- [ctype_space](#)
- [ctype_upper](#)
- [ctype_xdigit](#)

7.15.8 ctype_alpha() : Vérifie qu'une chaîne est alphabétique

`bool ctype_alpha ( string text )`

[ctype_alpha](#) retourne `TRUE` 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))`, mais certaines langues ont des caractères qui ne sont ni majuscule, ni minuscule.

Voir aussi [ctype_upper](#), [ctype_lower](#) et [setlocale](#).

7.15.9 ctype_cntrl() : Vérifie qu'un caractère est un caractère de contrôle

`bool ctype_cntrl ( string text )`

[ctype_cntrl](#) retourne `TRUE` si tous les caractères de la chaîne `text` sont des caractères de contrôle spéciaux. Ces caractères sont les nouvelles lignes, les tabulations, etc...

7.15.10 ctype_digit() : Vérifie qu'une chaîne est un entier

`bool ctype_digit ( string text )`

[ctype_digit](#) retourne `TRUE` si tous les caractères de la chaîne `text` sont des chiffres, et `FALSE` sinon.

Voir aussi [ctype_alnum](#) et [ctype_xdigit](#).

7.15.11 ctype_graph() : Vérifie qu'une chaîne est imprimable

`bool ctype_graph ( string text )`

[ctype_graph](#) retourne `TRUE` 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).

Voir aussi [ctype_alnum](#), [ctype_print](#) et [ctype_punct](#).

7.15.12 ctype_lower() : Vérifie qu'une chaîne est en minuscules

```
bool ctype_lower ( string text )
```

ctype_lower retourne `TRUE` si tous les caractères de la chaîne `text` sont des minuscules.

Voir aussi ctype_alpha et ctype_upper .

7.15.13 ctype_print() : Vérifie qu'une chaîne est imprimable

```
bool ctype_print ( string text )
```

ctype_print retourne `TRUE` si tous les caractères de la chaîne `text` va être imprimable à l'écran (y compris les espaces blancs).

Voir aussi ctype_cntrl , ctype_graph et ctype_punct .

7.15.14 ctype_punct() : Vérifie qu'une chaîne contient de la ponctuation

```
bool ctype_punct ( string text )
```

ctype_punct retourne `TRUE` si tous les caractères de la chaîne `text` est imprimable, mais n'est ni une lettre, ni un chiffre, ni un espace.

Voir aussi ctype_cntrl , ctype_graph et ctype_punct .

7.15.15 ctype_space() : Vérifie qu'une chaîne est faite caractères blancs

```
bool ctype_space ( string text )
```

ctype_space retourne `TRUE` si tous les caractères de la chaîne `text` vont créer un espace blanc, et sinon retourne `FALSE` . En plus de l'espace, cette définition couvre les tabulations (verticales et horizontales), les nouvelles lignes, les retours chariots et les retours à la ligne.

7.15.16 ctype_upper() : Vérifie qu'une chaîne est en majuscules

```
bool ctype_upper ( string text )
```

ctype_upper retourne `TRUE` si tous les caractères de la chaîne `text` sont des majuscules.

Voir aussi ctype_alpha et ctype_lower .

7.15.17 ctype_xdigit() : Vérifie qu'un caractère représente un nombre hexadécimal

```
bool ctype_xdigit ( string text )
```

`ctype_xdigit` retourne `TRUE` si tous les caractères de la chaîne `text` sont des chiffres hexadécimaux, c'est à dire un chiffre ou une des lettres `[A-Fa-f]`, sinon, elle renvoie `FALSE`.

Voir aussi `ctype_digit`.

7.16 DBA

7.16.1 Introduction

Ces fonctions sont l'interface avec les bases de type Berkeley.

C'est une couche générale pour plusieurs bases de données sur fichiers. En tant que tel, les fonctionnalités sont limitées à une partie des fonctionnalités des bases de données modernes, comme Sleepycat Software's DB2 (A ne pas confondre avec IBM's DB2 software, qui fonctionne avec ODBC).

7.16.2 Pré-requis

The behaviour of various aspects depends on the implementation of the underlying database. Functions such as dba_optimize and dba_sync will do what they promise for one database and will do nothing for others. You have to download and install supported dba-Handlers.

Librairie	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 (elle est de ce fait obsolète).
gdbm	gdbm est la base dbm GNU.
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> .
cdb	cdb est "un package rapide, robuste, léger, pour créer et lire des bases de données constantes". C'est l'auteur de gmail 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 librairies internes de cdb.
cdb_make	Depuis PHP 4.3.0, la création de fichier cdb est supportée, lorsque la librairie 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 librairies dbm pour compiler PHP avec.

Lors de l'utilisation de la fonction dba_open ou de dba_popen , une des librairies suivantes doit être fournie comme argument. La liste complète des librairies supportées par votre configuration est disponible avec la fonction phpinfo .

7.16.3 Installation

En utilisant l'option de compilation `--enable-dba=shared` , vous pouvez compiler un module dynamique qui activera le support des bases de données DBM pour PHP. Vous devez aussi ajouter le support de l'un des gestionnaires suivants, en spécifiant l'option de configuration `--with-XXXX` .

Handler	Configure Switch
dbm	Pour activer le support de dbm, ajoutez <code>--with-dbm[=DIR]</code> .
ndbm	Pour activer le support de ndbm, ajoutez <code>--with-ndbm[=DIR]</code> .
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> .
db3	Pour activer le support de db3, ajoutez <code>--with-db3[=DIR]</code> .
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 librairie cdb fournie avec PHP, qui ajoute un gestionnaire <u>cdb_make</u> , qui permet la création de fichier cdb, et l'accès aux fichiers cbd via le réseau avec les flôts de PHP.
flatfile	Pour activer le support des fichiers, ajoutez <code>--with-flatfile</code> .
	Note

	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 librairies dbm pour compiler PHP avec.
inifile	Pour activer le support de inifile, ajoutez l'option de compilation <code>--with-inifile</code> .
	Note
	Cette option a été ajoutée en ^{PHP} 5.0.0 et permet de lire et écrire dans des fichiers d'initialisation de type microsoft (.ini), comme le <code>php.ini</code> par exemple).

Note
Jusqu'en ^{PHP} 4.3.0, il était possible d'utiliser simultanément les gestionnaires db2 et db3 mais n'en n'utiliser qu'un seul en interne. Cela signifie que vous ne pouvez pas avoir les deux formats de fichiers. Depuis ^{PHP} 5.0.0, il existe une option de configuration pour éviter ces problèmes.

7.16.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.16.5 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.

7.16.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.16.7 Exemples

Exemple DBA (1)
<pre> <?php \$id = dba_open ("/tmp/test.db", "n", "db2"); if (!\$id) { echo "dba_open a échoué\n"; exit; } dba_replace ("cle", "Ceci est un exemple!", \$id); if (dba_exists ("cle", \$id)) { echo dba_fetch ("cle", \$id); dba_delete ("cle", \$id); } dba_close (\$id); ?> </pre>

DBA gère les données binaires, et n'a pas de limites arbitraires. Elle hérite de toutes les limites de la base sous-jacentes.

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 4ème 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...

$key = dba_firstkey ($id);

while ($key != false) {
    if (...) {
        // conserver la clé pour faire d'autres opérations plus tard
        $handle_later[] = $key;
    }
    $key = dba_nextkey ($id);
}

for ($i = 0; $i < count($handle_later); $i++)
    dba_delete ($handle_later[$i], $id);

?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [dba_close](#)
- [dba_delete](#)
- [dba_exists](#)
- [dba_fetch](#)
- [dba_firstkey](#)
- [dba_handlers](#)
- [dba_insert](#)
- [dba_list](#)
- [dba_nextkey](#)
- [dba_open](#)
- [dba_optimize](#)
- [dba_popen](#)
- [dba_replace](#)
- [dba_sync](#)

7.16.9 dba_delete() : Efface une entrée

bool dba_delete (string key , resource handle)

[dba_delete](#) efface l'entrée spécifiée par la clé ^{key}, dans la base identifiée par ^{handle}.

^{key} est la clé de l'entrée à effacer.

^{handle} est un identifiant de lien, retourné par [dba_open](#).

[dba_delete](#) retourne ^{TRUE} ou ^{FALSE}, si l'entrée est effacée, ou pas effacée, respectivement.

Voir aussi [dba_exists](#), [dba_fetch](#), [dba_insert](#) et [dba_replace](#)

7.16.10 dba_exists() : Vérifie qu'une clé existe

bool **dba_exists** (string key , resource handle)

dba_exists vérifie si la clé ^{key} existe dans la base identifiée par ^{handle} .

^{key} est la clé qui doit être vérifiée.

^{handle} est un identifiant de base, retourné par dba_open .

dba_exists retourne ^{TRUE} ou ^{FALSE} , si la clé est trouvée, ou pas, respectivement.

Voir aussi dba_fetch , dba_delete , dba_insert et dba_replace .

7.16.11 dba_fetch() : Lit les données liées à une clé

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} .

^{key} est la clé dont on veut lire les données.

^{skip} est le nombre de paire clé-valeur que vous souhaitez ignorer lors de l'utilisation de bases cdb. Cette valeur sera ignorée pour toutes les autres bases de données qui ne supportent pas les clés multiples avec le même nom.

^{handle} est un identifiant de base, retourné par dba_open .

dba_fetch retourne la chaîne associée ou ^{FALSE} , si la paire clé/valeur n'a pas été trouvée.

Note
Le paramètre ^{skip} est disponible depuis ^{PHP} 4.3 pour supporter les fonctionnalités de cdb : le support des clé multiples avec le même nom.

Voir aussi dba_exists , dba_delete , dba_insert et dba_replace .

7.16.12 dba_firstkey() : Lit la première clé

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.

^{handle} est un identifiant de base, retourné par dba_open .

dba_firstkey retourne la clé, ou ^{FALSE} , suivant que la première clé existe ou pas.

Voir aussi dba_nextkey et l'exemple 2 de l'introduction DBA .

7.16.13 dba_handlers() : Liste les gestionnaires dba disponibles

array **dba_handlers** (void)

dba_handlers retourne un tableau avec la liste des gestionnaires supportés par cette extension.

Lorsque la librairie interne cdb est utilisée, vous verrez 'cdb' et 'cdb_make'.

7.16.14 dba_insert() : Insère une entrée

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}. Si une entrée avec la même clé ^{key} existe déjà, l'insertion échouera.

^{key} est la clé de la valeur à insérer.

^{value} est la valeur à insérer.

^{handle} est un identifiant de base, retourné par dba_open.

dba_insert retourne TRUE ou FALSE, suivant que l'insertion a réussi ou échoué.

Voir aussi dba_exists, dba_delete, dba_fetch et dba_replace.

7.16.15 dba_list() : Liste tous les fichiers de bases de données ouverts

array **dba_list** (void)

dba_list retourne un tableau associatif, avec tous les fichiers de bases de données ouvertes. Ce tableau est de la forme ^{ressourceid=>nom de fichier}.

7.16.16 dba_nextkey() : Lit la clé suivante

string **dba_nextkey** (resource handle)

dba_nextkey retourne la clé suivante, dans la base identifiée par ^{handle} et incrémente le pointeur de clé.

^{handle} est un identifiant de base, retourné par dba_open.

dba_nextkey retourne la clé, ou FALSE en cas d'échec.

Voir aussi dba_firstkey.

7.16.17 dba_open() : Ouvre une base de données

resource **dba_open** (string path , string mode , string handler , ...)

dba_open établit une connexion à la base repérée par ^{path} avec le mode ^{mode} et l'identifiant ^{handler}.

`path` est le chemin sur votre machine.

`mode` vaut "r" pour lecture seule, "w" pour lecture/écriture, "c" pour lecture/écriture, et création si la base n'existe pas, et "n" pour création, écrasement, et accès en lecture/écriture.

`handler` est le nom de l'identifiant qui sera utilisé pour accéder à `path`. Il est passé à `dba_popen`.

Voir aussi `dba_popen` et `dba_close`.

7.16.18 dba_optimize() : Optimise une base

`bool dba_optimize ( resource handle )`

`dba_optimize` optimise la base de données identifiée par `handle`.

`handle` est un identifiant de base retourné par `dba_open`.

`dba_optimize` retourne `TRUE` ou `FALSE`, suivant que l'optimisation a réussi ou échoué.

Voir aussi `dba_sync`.

7.16.19 dba_popen() : Ouvre une connexion persistante à une base de données dba

`resource dba_popen ( string path , string mode , string handler , ... )`

`dba_popen` établit une connexion persistante à la base repérée par `path` avec le mode `mode`, en utilisant l'identifiant `handler`.

`path` est le chemin sur votre machine.

`mode` vaut "r" pour lecture seule, "w" pour lecture/écriture, "c" pour lecture/écriture, et création si la base n'existe pas, et "n" pour création, écrasement, et accès en lecture/écriture. De plus, vous pouvez poser un verrou sur la base de données avec un caractère supplémentaire. Utilisez le "l" pour verrouiller la base avec un fichier .lck ou bien "d" pour verrouiller la base de données elle-même. Il est important que vos applications utilisent cela de manière cohérence. Si vous voulez tester l'accès et ne pas attendre pour le verrou, vous pouvez ajouter l'option "t" comme troisième caractère. Lorsque vous êtes absolument sûr que vous n'avez pas besoin du verrou de base, utilisez l'option "-" au lieu de "l" ou "d". Lorsqu'aucun de "d", "l" ou "-" est utilisé dba va verrouiller la base de données avec l'option "d".

`handler` est le nom de l'identifiant qui sera utilisé pour accéder à `path`. Il est passé à `dba_popen`.

`dba_popen` retourne un identifiant positif, ou `FALSE`, suivant que la base a été ouverte, ou que l'accès a échoué.

Note

Il ne peut y avoir qu'un seul accès en écriture sur un fichier de base de données DBA. Lorsque vous utilisez DBA sur un serveur web et que plusieurs requêtes demandent un verrou d'écriture, il ne peut y en avoir qu'un seul après les autres. De plus, durant l'écriture, la lecture n'est pas autorisée. Voyez la table suivante :

already open	mode = "rl"	mode = "rlt"	mode = "wl"	mode = "wlt"	mode = "rd"	mode = "rdt"	mode = "wd"	mode = "wdt"
not open	ok	ok	ok	ok	ok	ok	ok	ok
mode = "rl"	ok	ok	wait	false	illegal	illegal	illegal	illegal
mode = "wl"	wait	false	wait	false	illegal	illegal	illegal	illegal
mode = "rd"	illegal	illegal	illegal	illegal	ok	ok	wait	false
mode = "wd"	illegal	illegal	illegal	illegal	wait	false	wait	false

- ok : le second appel réussira.
- wait : le second appel attend que [dba_close](#) soit appelé pour le premier.
- false : le second appel retourne FALSE .
- illégal : vous ne pouvez pas mêler les options "l" et "d" pour l'option mode .

Note

Depuis ^{PHP} 4.3.0, il est possible d'ouvrir un fichier de base de données à travers un réseau. Toutes fois, dans le cas des connexion sockets (comme par exemple via ftp ou http), la connexion sera verrouillée, en lieu et place du fichier. C'est une nuance importante car le verrouillage est alors ignoré par la ressource, et il faut trouver une alternative.

Note

Le verrouillage et les options de ^{mode} "l", "d", "-" et "t" ont été ajoutés en PHP 4.3.0. Dans les versions de ^{PHP} antérieures à ^{PHP} 4.3.0, vous devez utiliser des sémaphore pour vous protéger contre les accès simultanés à un gestionnaire de base de données, à l'exception de GDBM. Voyez les [sémaphores System V](#) .

Voir aussi [dba_open](#) et [dba_close](#) .

7.16.20 dba_replace() : Remplace ou insère une entrée

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} .

^{key} est la clé qui va être insérée.

^{value} est la valeur qui va être insérée.

^{handle} est un identifiant de base retourné par [dba_open](#) .

[dba_replace](#) retourne ^{TRUE} ou ^{FALSE} , suivant que l'opération réussit ou échoue.

Voir aussi [dba_exists](#) , [dba_delete](#) , [dba_fetch](#) et [dba_insert](#) .

7.16.21 dba_sync() : Synchronise une base de données

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.

^{handle} est un identifiant de base retourné par [dba_open](#) .

[dba_sync](#) retourne ^{TRUE} ou ^{FALSE} , si la synchronisation réussit, ou échoue, respectivement.

Voir aussi [dba_optimize](#) .

7.17 Dates et heures

7.17.1 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, ainsi qu'aux années bissextiles.

7.17.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.17.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.17.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.17.5 Types de ressources

Cette extension ne définit aucune ressource.

7.17.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [checkdate](#)
- [date](#)
- [getdate](#)
- [gettimeofday](#)
- [gmdate](#)
- [gmmktime](#)
- [gmstrftime](#)
- [localtime](#)
- [microtime](#)

- [mktime](#)
- [strftime](#)
- [strtotime](#)
- [time](#)

7.17.8 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.

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).
Pour générer un timestamp à partir d'une représentation de date, pour pouvez utiliser la fonction strtotime . De plus, certaines bases de données disposent de fonctions pour convertir leurs propres formats de date en timestamps (comme par exemple, MySQL et sa fonction UNIX_TIMESTAMP()).

Les caractères suivants sont utilisés pour spécifier le format :

Caractère	Description	Exemple de valeur
a	Ante meridiem et Post meridiem en minuscules	am ou pm
A	Uppercase Ante meridiem and Post meridiem	AM ou PM
B	Heure Internet Swatch	000 à 999
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
F	Mois, textuel, version longue; en anglais, comme January ou December	January à December
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
I (i majuscule)	L'heure d'été est activée ou as	1 si oui, 0 sinon.
j	Jour du mois sans les zéros initiaux	1 à 31
l (lowercase 'L')	Jour de la semaine, textuel, version longue, en anglais	Sunday à Saturday
L	Est ce que l'année est bissextile	1 si bissextile, 0 sinon.
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
O	Différence d'heures avec l'heure de Greenwich (GMT), exprimée en heures	Exemple : +0200
r	Format de date RFC 822	Exemple : Thu, 21 Dec 2000 16:01:07 +0200
s	Secondes, avec zéros initiaux	00 à 59
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
t	Nombre de jours dans le mois	28 à 31
T	Fuseau horaire du serveur	Exemples : EST , MDT ...
U	Secondes depuis l'époque Unix (1er Janvier 1970, 0h00 00s GMT)	Voir aussi time
w	Jour de la semaine au format numérique	0 (pour dimanche) à 6 (pour samedi)
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)
Y	Année, 4 chiffres	Exemples : 1999 et 2003
y	Année, 2 chiffres	Exemples : 99 et 03
z	Jour de l'année	0 à 366
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

Les caractères non reconnus seront imprimés tels quel. "Z" retournera toujours "0" lorsqu'il est utilisé avec gmdate .

Exemple avec date

```
<?php
echo (date("l dS of F Y h:i:s A"));
echo ("Le 14 Juillet 2002 tombe un " . date("l", mktime(0,0,0,7,14,2002)));
?>
```

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.

Protection des caractères avec date

```
<?php
echo date("l \\t\\h\\e js");
// affiche 'Saturday the 8th'
// \\t représente une tabulation
?>
```

Il est possible d'utiliser date et mktime ensemble pour générer des dates dans le futur ou dans le passé.

Exemples avec date et mktime

```
<?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);
?>
```

Note

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'hivers 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 .

Voir aussi getlastmod , time , strftime , gmdate et mktime .

7.17.9 `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), avec les champs suivants :

- "seconds" – secondes
- "minutes" – minutes
- "hours" – heures
- "mday" – jour du mois
- "wday" – jour de la semaine, numérique. 0: dimanche jusqu'à 6: samedi
- "mon" – mois, numérique
- "year" – année, numérique
- "yday" – jour de l'année, numérique; i.e. "299"
- "weekday" – jour de la semaine, texte complet (en anglais); i.e. "Friday"
- "month" – mois, texte complet (en anglais); i.e. "January"

Exemple avec getdate

```
<?php
$aujourd'hui = getdate();
$mois = $aujourd'hui['month'];
$mjour = $aujourd'hui['mday'];
$annee = $aujourd'hui['year'];
echo "$mjour/$mois/$annee";
?>
```

Voir aussi date , time et setlocale .

7.17.10 `gettimeofday()` : Retourne l'heure actuelle

array **gettimeofday** (void)

gettimeofday est une interface vers `gettimeofday(2)`. Elle retourne un tableau associatif qui contient les informations retournées par le système :

- "sec" – secondes
- "usec" – microsecondes
- "minuteswest" – minutes de décalage par rapport à Greenwich, vers l'Ouest.
- "dsttime" – type de correction dst

7.17.11 `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

Sous Microsoft Windows, les bibliothèques qui implémentent cette fonction sont buggées, et `gmdate` ne supporte pas les valeurs négatives pour le paramètre `timestamp`. Pour plus de détails, voyez ces rapports de bugs : [#22620](#), [#22457](#), et [#14391](#).

Ce problème n'est pas présent sur les systèmes d'exploitation Unix et Linux, où les bibliothèques se comportent comme prévu.

PHP ne peut pas réparer ces bibliothèques systèmes. Contactez l'auteur de votre système d'exploitation pour corriger ce problème.

Voir aussi `date`, `strftime`, `mktime` et `gmmktime`.

7.17.12 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.

Comme `mktime`, les arguments restants peuvent être ignorés. Ils prendront alors leur valeur GMT actuelle.

7.17.13 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`.

7.17.14 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 0 ou ignoré, force `localtime` à retourner un tableau à index numérique. S'il est mis à 1, `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
- "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?

7.17.15 microtime() : Retourne le timestamp UNIX actuel avec microsecondes

string **microtime** (void)

microtime 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().

Les deux parties de la chaîne sont exprimées en secondes.

Exemple avec microtime

```
<?php
function getmicrotime(){
    list($usec, $sec) = explode(" ",microtime());
    return ((float)$usec + (float)$sec);
}

$time_start = getmicrotime();

for ($i=0; $i < 1000; $i++){
    //do nothing, 1000 times
}

$time_end = getmicrotime();
$time = $time_end - $time_start;

echo "Ne fait rien pendant $time secondes";
?>
```

Voir aussi time .

7.17.16 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)

ATTENTION : l'ordre des arguments est différent de celui de la commande UNIX habituelle mktime(), et fournit des résultats aléatoires si on oublie cet ordre. C'est une erreur très commune que de se tromper de sens.

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) 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.

`is_dst` peut être mis à 1 si l'heure d'hiver est appliquée, 0 si elle ne l'est pas, et -1 (par défaut) si on ne sait pas.

Note

<code>is_dst</code> a été ajouté à partir de la version 3.0.10.

`mktime` est pratique pour faire des calculs de dates et des validations, car elle va automatiquement corriger les valeurs invalides. Par exemple, toutes les lignes suivantes vont retourner la même date : "Jan-01-1998".

Exemple <code>mktime</code>

<pre><?php echo date("M-d-Y", mktime (0,0,0,12,32,1997)); echo date("M-d-Y", mktime (0,0,0,13,1,1997)); echo date("M-d-Y", mktime (0,0,0,1,1,1998)); echo date("M-d-Y", mktime (0,0,0,1,1,98)); ?></pre>

`year` peut prendre deux ou quatre chiffres, avec les valeurs entre 0–69 qui correspondent à 2000–2069 et 70–99 à 1970–1999 (sur les systèmes où `time_t` sont sur des entiers 32bit signés, comme cela se fait le plus souvent de nos jours, `year` est valide dans l'intervalle 1902 et 2037).

Le dernier jour d'un mois peut être décrit comme le jour "0" du mois suivant, et non pas le jour -1. Les deux exemples suivants vont donner : "Le dernier jour de Février 2000 est: 29".

Dernier jour du mois

<pre><?php \$lastday = mktime (0,0,0,3,0,2000); echo strftime ("Le dernier jour de Fevrier 2000 est: %d", \$lastday); \$lastday = mktime (0,0,0,4,-31,2000); echo strftime ("Le dernier jour de Fevrier 2000 est: %d", \$lastday); ?></pre>

Voir aussi `date` et `time` .

7.17.17 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')

- %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)
- %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.
- %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 – fuseau horaire, ou nom ou abréviation
- %% – un caractère '%' littéral

Note

Tous les caractères suivants ne sont pas toujours supportés par toutes les librairies 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");
print(strftime("%A en Finlandais est "));
setlocale (LC_TIME, "fi");
print(strftime("%A, en Français "));
setlocale (LC_TIME, "fr");
print(strftime("%A et en Allemand "));
setlocale (LC_TIME, "de");
print(strftime("%A.\n"));
?>
```

Cet exemple ne fonctionnera que si vous avez les configurations respectives installées sur votre système.

Note

%G et %V, qui sont basées sur la semaine ISO 8601:1988, peut conduire à des résultat 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
print "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
print "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
print "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
print "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
print "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
print "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
print "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
print "1/3/2005 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/3/2005")) . "\n";

?>

```

Voir aussi [strftime](#) , [setlocale](#) , [mktime](#) et le [groupe de spécifications de strftime\(\)](#) .

7.17.18 strtotime() : Transforme un texte anglais en timestamp

```
int strtotime ( string time , int now )
```

[strtotime](#) essaye de lire une date au format anglais dans la chaîne ^{time}, et de la transformer en timestamp UNIX, relativement au timestamp ^{now}, ou à la date courante si ce dernier est omis. En cas d'erreur, ⁻¹ est retourné.

Comme [strtotime](#) suit la syntaxe de date GNU, vous pouvez lire le manuel GNU intitulé [Date Input Formats](#) . Ci-dessous, vous trouverez une syntaxe valide pour le paramètre ^{time} .

Exemple avec [strtotime](#)

```

<?php
// l'exemple n'est pas traduit, car cela ne fonctionne qu'en anglais
echo strtotime ("now"), "\n";
echo strtotime ("10 September 2000"), "\n";
echo strtotime ("+1 day"), "\n";
echo strtotime ("+1 week"), "\n";
echo strtotime ("+1 week 2 days 4 hours 2 seconds"), "\n";
echo strtotime ("next Thursday"), "\n";
echo strtotime ("last Monday"), "\n";
?>

```

Vérification des erreurs avec [strtotime](#)

```
$str = 'Pas une bonne date';
if (($timestamp = strtotime($str)) === -1) {
    echo "La chaîne ($str) est bogus";
} else {
    echo "$str == ". date('l dS of F Y h:i:s A', $timestamp);
}
```

Note

L'intervalle de validité d'un timestamp va du Vendredi 13 Décembre 1901 20:45:54 GMT au Mardi 19 Jan 2038 03:14:07 GMT : cela correspond aux dates maximales et minimales pour un entier de 32 bits signé.

7.17.19 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).

Voir aussi date .

7.18 dBase

7.18.1 Introduction

Ces fonctions vous permettront d'accéder aux enregistrements d'une base au format dBase (.dbf).

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.

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](#).

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 PostgreSQL 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 des données issues du web, car ce format de fichier est communément accepté par les feuilles et organisateurs Windows. L'import et l'export de données sont les deux uniques fonctionnalités qui recommandent l'utilisation de dBase.

7.18.2 Installation

Pour activer la librairie dbase fournie avec PHP, vous devez compiler PHP avec l'option `--enable-dbase`.

7.18.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.18.4 Types de ressources

Cette extension ne définit aucune ressource.

7.18.5 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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](#)

7.18.7 dbase_close() : Ferme une base dBase

`bool dbase_close ( int dbase_identifiant )`

[dbase_close](#) ferme la base associée à `dbase_identifiant`.

7.18.8 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`.

`fields` est un tableau de tableaux. Chaque tableau décrit le format d'un fichier de la base. Chaque champs est constitué d'un nom, d'un caractère de type de champs, d'une longueur et d'une précision.

Les types de champs disponibles sont :

<i>L</i>	Boolean (booléen). Pas de longueur ou de précision pour ces valeurs.
<i>M</i>	Memo. (Note importante : les Memos ne sont pas supportés par PHP.) Elles n'ont pas de longueur ou de précision.
<i>D</i>	Date (enregistrée au format 'YYYYMMDD'). Elles n'ont pas de longueur ou de précision.
<i>N</i>	Number (nombre). Possède une longueur et un précision (le nombre de chiffres après la virgule).
<i>C</i>	String (chaîne).

Si la base de données a été créée, un identifiant de base `dbase_identifiant` est retourné, sinon, `FALSE` est retourné.

```
Création d'une base dBase

<?php
// "database" name
$dbname = "/tmp/test.dbf";
// database "definition"
$def =
    array(
        array("date",      "D"),
        array("name",      "C", 50),
        array("age",       "N",  3, 0),
        array("email",     "C", 128),
        array("ismember",  "L")
    );
// création
if (!dbase_create($dbname, $def))
    print "<strong>Erreur!</strong>";
```

`?>`

7.18.9 `dbase_delete_record()` : Efface un enregistrement dans une base dBase

```
bool dbase_delete_record ( int dbase_identifiant , int record )
```

`dbase_delete_record` marque l'enregistrement `record` pour l'effacement, dans la base `dbase_identifiant`. Pour effacer réellement l'enregistrement, il faut utiliser aussi `dbase_pack`.

7.18.10 `dbase_get_header_info()` : Lit les informations d'entête d'une base dBase

```
array dbase_get_header_info ( int dbase_identifiant )
```

`dbase_get_header_info` retourne les informations de la colonne structure, pour la base de données référencée par `dbase_identifiant`. Pour chaque colonne de la base, il y a une entrée qui est un tableau indexé numériquement. L'index du tableau commence à 0. Chaque élément contient un tableau associatif, avec les informations de la colonne. Si les entêtes de la base ne peuvent pas être lus, la fonction retourne `FALSE`.

Les éléments du tableau sont :

<i>name</i>	Le nom de la colonne.
<i>type</i>	Le nom lisible humainement du type dBase de la colonne (i.e. date, booléen, etc...).
<i>length</i>	Le nombre d'octets que cette colonne peut contenir.
<i>precision</i>	Le nombre de décimales pour cette colonne.
<i>format</i>	Une suggestion de format pour <code>printf</code> .
<i>offset</i>	L'octet d'offset de la colonne depuis le début de la ligne.

Afficher les informations d'entête d'une base dBase

```
<?php
// Chemin jusqu'à la base dBase
$db_path = "/tmp/test.dbf";

// Ouvre la base
$dbh = dbase_open($dbname)
    or die("Impossible d'ouvrir le fichier de base de données! '$db_path'.");

// Lecture des informations
$column_info = dbase_get_header_info($dbh);

// Affichage des informations.
print_r($column_info);
?>
```

7.18.11 `dbase_get_record_with_names()` : Lit un enregistrement dans une base, sous la forme d'un tableau associatif

array **dbase_get_record_with_names** (int dbase_identifiant , int record)

`dbase_identifiant` retourne les données de l'enregistrement `record` dans un tableau associatif. Le tableau inclus un membre nommé 'deleted' (effacé), qui sera mis à 1 si l'enregistrement a été marqué pour l'effacement (voir [dbase_delete_record](#)).

Chaque champs est converti au format approprié PHP, hormi :

- Les dates, qui sont laissées au format chaîne
- Les entiers qui causeraient des dépassement de capacité (> 32 bits) sont retournés comme des chaînes.

7.18.12 `dbase_get_record()` : Lit un enregistrement dans une base dBase

array **dbase_get_record** (int dbase_identifiant , int record)

[dbase_get_record](#) retourne les données de l'enregistrement `record` dans un tableau. Le tableau est indexé à partir de 0, et inclus un membre nommé 'deleted' (effacé), qui sera mis à 1 si l'enregistrement a été marqué pour l'effacement (voir [dbase_delete_record](#)).

Chaque champs est converti au format approprié PHP, hormi :

- Les dates, qui sont laissées au format chaîne
- Les entiers qui causeraient des dépassement de capacité (> 32 bits) sont retournés comme des chaînes.

7.18.13 `dbase_numfields()` : Compte le nombre de champs d'une base dBase

int **dbase_numfields** (int dbase_identifiant)

[dbase_numfields](#) retourne le nombre de champs (colonnes) de la base de données `dbase_identifiant`. Les numéros de champs sont numérotés de 0 à `dbase_numfields($db)-1`, tandis que les numéros d'enregistrements sont numérotés de 1 à `dbase_numrecords($db)`.

Utiliser [dbase_numfields](#)

```
<?php
$rec = dbase_get_record($db, $recno);
$nf = dbase_numfields($db);
for ($i=0; $i < $nf; $i++) {
    print $rec[$i]."<BR>\n";
}
?>
```

7.18.14 `dbase_numrecords()` : Compte le nombre d'enregistrements dans une base dBase

```
int dbase_numrecords ( int dbase_identifiant )
```

dbase_numrecords retourne le nombre d'enregistrements (lignes) dans la base `dbase_identifiant`. Les numéros de champs sont numérotés de 0 à `dbase_numfields($db)-1`, tandis que les numéros d'enregistrements sont numérotés de 1 à `dbase_numrecords($db)`.

7.18.15 dbase_open() : Ouvre une base dBase

```
int dbase_open ( string filename , int flags )
```

dbase_open retourne un identifiant de base de données, ou `FALSE` si la base n'a pas pu être sélectionnée.

`flags` est un flag, comme pour la fonction `open()`. (Typiquement; 0 signifie lecture seule, 1 signifie écriture seule, et 2 écriture/lecture).

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.

7.18.16 dbase_pack() : Compacte une base dBase

```
bool dbase_pack ( int dbase_identifiant )
```

dbase_pack compacte la base de données `dbase_identifiant` (effacement définitif de tous les enregistrements marqués pour l'effacement, avec la fonction dbase_delete_record).

7.18.17 dbase_replace_record() : Remplace un enregistrement dans une base dBase

```
bool dbase_replace_record ( int dbase_identifiant , array record , int dbase_record_number )
```

dbase_replace_record remplace les données associées à l'enregistrement `record_number` par les données enregistrées dans `record`. Si le nombre de colonnes fourni n'est pas celui du nombre de champs dans la base, l'opération échouera, et `FALSE` sera retourné.

`dbase_record_number` est un entier qui peut aller de 1 jusqu'au nombre maximal d'enregistrement de la base (retourné par dbase_numrecords).

7.19 DBM (Obsolètes)

7.19.1 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 librairies systèmes, ou certaines librairies du système d'exploitation) enregistre les paires clés/valeurs, (contrairement aux enregistrements par ligne, utilisés par les autres bases de données relationnelles).

Note

Toutes fois, le support dbm est obsolète, et il est fortement recommandé d'utiliser la [couche d'abstraction de base dba](#).

7.19.2 Pré-requis

Pour utiliser ces fonctions, vous devez compiler PHP avec le support d'un gestionnaire. Reportez vous à la [liste](#) de gestionnaires supportés.

7.19.3 Installation

In order to use these functions, you must compile PHP with dbm support by using the `--with-db` option. In addition you must ensure [support](#) for an underlying database or you can use some sytem libraries.

7.19.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.19.5 Types de ressources

La fonction [dbmopen](#) retourne une ressource de base de données, utilisée par toutes les autres fonctions dbm.

7.19.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.19.7 Exemples

Présentation de dbm

```
$dbm = dbmopen ("lastseen", "w");
if (dbmexists ($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 prédefinies](#)
- [Exemples](#)
- [dblist](#)
- [dbmclose](#)
- [dbmdelete](#)
- [dbmexists](#)
- [dbmfetch](#)
- [dbmfirstkey](#)
- [dbminsert](#)
- [dbmnextkey](#)
- [dbmopen](#)
- [dbmreplace](#)

7.19.9 dbmclose() : Ferme une base de données dbm

```
bool dbmclose ( resource dbm_identifieur )
```

[dbmclose](#) déverrouille et ferme la base de données `dbm_identifieur`.

7.19.10 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.

7.19.11 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`.

7.19.12 dbmfetch() : Lit une valeur

```
string dbmfetch ( resource dbm_identifieur , string key )
```

[dbmfetch](#) retourne la valeur associée à la clé `key`.

7.19.13 dbmfirstkey() : Lit la première clé

string **dbmfirstkey** (resource dbm_identifiant)

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.

7.19.14 dbminsert() : Insère une valeur

int **dbminsert** (resource dbm_identifiant , string key , string value)

dbminsert ajoute la valeur `value` 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.)

7.19.15 dbmnextkey() : Lit la clé suivante

string **dbmnextkey** (resource dbm_identifiant , 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
<pre><?php \$cle = dbmfirstkey(\$dbm_id); while (\$cle){ echo "\$cle = " . dbmfetch(\$dbm_id, \$cle) . "\n"; \$cle = dbmnextkey(\$dbm_id, \$cle); } ?></pre>

7.19.16 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'ajoute à celui éventuellement utilisé par les bibliothèques. PHP n'efface jamais les fichiers `.lock` 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.

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.

7.19.17 dbmreplace() : Remplace une valeur

int **dbmreplace** (resource dbm_identifiant , string key , string value)

dbmreplace remplace la valeur courante par la valeur ^{value} pour la clé ^{key} , dans une base ^{dbm_identifiant} .

dbmreplace crée la clé, si elle n'existe pas dans la base.

7.20 dbx

7.20.1 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'interfacent pas directement avec les bases de données, mais s'interfacent avec les modules utilisées pour supporter ces bases.

7.20.2 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) (CVS uniquement).

La documentation nécessaire pour ajouter le support d'autres bases de données est disponible à l'adresse <http://www.guidance.nl/php/dbx/doc/> .

7.20.3 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` .

7.20.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<code>dbx.colnames_case</code>	"unchanged"	PHP_INI_SYSTEM

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#) .

Note
Ces options de configurations sont uniquement disponibles à partir du ^{PHP} issu du serveur CVS.

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 "majuscule" ("uppercase") ou encore "minuscules" ("lowercase"). Cette directive peut être reconfigurée par une option dans [dbx_query](#).

7.20.5 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.

7.20.6 Constantes prédefinies

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*) CVS only
 DBX_SYBASECT (*entier*)
 DBX_PERSISTENT (*entier*)
 DBX_RESULT_INFO (*entier*)
 DBX_RESULT_INDEX (*entier*)
 DBX_RESULT_ASSOC (*entier*)
 DBX_COLNAMES_UNCHANGED (*entier*) CVS only
 DBX_COLNAMES_UPPERCASE (*entier*) CVS only
 DBX_COLNAMES_LOWERCASE (*entier*) CVS only
 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 prédefinies](#)
- [dbx_close](#)
- [dbx_compare](#)
- [dbx_connect](#)
- [dbx_error](#)
- [dbx_escape_string](#)
- [dbx_fetch_row](#)
- [dbx_query](#)
- [dbx_sort](#)

7.20.8 dbx_compare() : Compare deux lignes 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 ^{flag} est configurée à DBX_CMP_DESC . dbx_compare est une fonction d'aide pour dbx_sort , afin d'aider aux tris. sorting function.

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 .

7.20.9 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)

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 persistante.

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" (CVS uniquement)

Le support de pgsql était au stade expérimental jusqu'en avril 2001, et vous devez compiler vous-même le module pgsql après avoir modifié un des fichiers sources. Sinon, vous aurez une alerte affichée à chaque requête.

La ressource dbx_link_object a trois membres : 'handle', 'module' et 'database'. Le membre 'database' contient le nom de la base de données actuellement sélectionnée. Le membre 'module' est à usage interne à dbx, et contient le numéro de module sus-cité. Le membre 'handle' est une ressource valide de connexion à la base de données, et peut être utilisé en tant que tel dans les autres fonctions spécifiques à cette base de données.

Le message d'erreur pour Microsoft SQL server est actuellement le résultat direct de la fonction `mssql_get_last_message`.

Exemple avec `dbx_connect`

```
<?php
$link = dbx_connect (DBX_MYSQL, "localhost", "db", "username", "password");
mysql_close ($link->handle);
// dbx_close($link) est beaucoup plus adapté ici
?>
```

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.

Exemple avec `dbx_connect`

```
<?php
$link = dbx_connect (DBX_ODBC, "", "db", "username", "password", DBX_PERSISTENT)
    or die ("Impossible de se connecter");

print ("Connexion réussie");
dbx_close ($link);
?>
```

Note

Reportez-vous aussi à la documentation de la base de données que vous utilisez.

Voir aussi `dbx_close`.

7.20.10 dbx_error() : Rapporte le message d'erreur du dernier appel de fonction

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 retourné (le module est alors représenté par `link_identifiant`). Notez que le module ODBC ne supporte pas encore cette fonction.

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).

7.20.11 dbx_escape_string() : Protège une chaîne de caractères pour l'utiliser dans une requête

string **dbx_escape_string** (object link_identifiant , 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`.

7.20.12 dbx_fetch_row() : Lit une ligne dans un résultat DBX ayant l'option `DBX_RESULT_UNBUFFERED` activée

object **dbx_fetch_row** (object result_identifiant)

`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`

```
$result = dbx_query ($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_identifiant` est le résultat de la fonction `dbx_query`.

Le tableau retournée 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`. were set that way.

Voir aussi `dbx_query`.

7.20.13 dbx_query() : Envoie une requête et lit tous les résultats

object **dbx_query** (object link_identifiant , string sql_statement , long flags)

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.

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 quelquechose d'utile ici
    // first, print out field names and types
    // then, draw a table filled with the returned field values
}
else if ( $result == 1 ) {
    echo("Requête exécutée correctement, mais pas de résultat.");
}
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`.

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 si 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_COLNAMES_UNCHANGED (disponible depuis PHP 4.3.0)

La casse du nom des colonnes retournée 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, suivants 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 colonne et de ligne.

```
$result = dbx_query ($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`

```
$result = dbx_query ($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`

```
$result = dbx_query ($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";
```

Note

Reportez-vous aussi à la documentation de la base de données que vous utilisez.

Voir aussi [dbx_escape_string](#) et [dbx_connect](#).

7.20.14 dbx_sort() : Tri un résultat avec une fonction utilisateur

bool **dbx_sort** (object result , string user_compare_function)

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec [dbx_sort](#)

```
<?php
function user_re_order ($a, $b) {
    $rv = dbx_cmp_desc($a, $b, "parentid");
    if (!$rv) $rv = dbx_cmp_asc($a, $b, "id");
    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");
echo "Les données sont maintenant triées par id<BR>";
dbx_sort($result, "user_re_order");
echo "Les données sont maintenant triées par parentid décroissant, puis par id<BR>";
dbx_close($link);
?>
```

Voir aussi [dbx_compare](#) .

7.21 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.

7.21.1 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.

7.21.2 Pré-requis

Cette extension dépend de librairie 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).

7.21.3 Installation

Pour compiler cette extension par vous même, vous devez avoir les librairies 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 librairies 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` .

7.21.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.21.5 Types de ressources

7.21.5.1 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, générée par une requête.

7.21.6 Constantes prédefinies

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.

7.21.6.1 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_PGSIZE (entier)	ERR_PGSIZE	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'utilisateur
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 (entier)	ERR_DBPARSE	Erreur dans l'analyseur
DBPLUS_ERR_DBRUNERR (entier)	ERR_DBRUNERR	Erreur durant l'exécution
DBPLUS_ERR_DBPREEXIT (entier)	ERR_DBPREEXIT	La condition d'exit a été causée par la procédure prexit()
DBPLUS_ERR_WAIT (entier)	ERR_WAIT	Attend un peu (Simple seulement)
DBPLUS_ERR_CORRUPT_TUPLE (entier)	ERR_CORRUPT_TUPLE	Un client a envoyé une ligne corrompue
DBPLUS_ERR_WARNING0 (entier)	ERR_WARNING0	Les routines Simple ont rencontrées une erreur non fatale, qui a été corrigée.
DBPLUS_ERR_PANIC (entier)	ERR_PANIC	Le serveur ne devrait pas s'arrêter, mais après un désastre, envoyer ERR_PANIC à tout ses clients
DBPLUS_ERR_FIFO (entier)	ERR_FIFO	Impossible de créer une pile FIFO
DBPLUS_ERR_PERM (entier)	ERR_PERM	Permission refusée
DBPLUS_ERR_TCL (entier)	ERR_TCL	TCL_error
DBPLUS_ERR_RESTRICTED (entier)	ERR_RESTRICTED	Seulement deux utilisateurs
DBPLUS_ERR_USER (entier)	ERR_USER	Une erreur est survenue dans l'utilisation de la librairie par l'application.
DBPLUS_ERR_UNKNOWN (entier)	ERR_UNKNOWN	Erreur inconnue

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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_rctxexact](#)
- [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](#)

7.21.8 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... Query Language sont disponibles dans le manuel DB++.

7.21.9 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 , vous pouvez lire le nom du dossier courant sans le modifier.

7.21.10 dbplus_close() : Ferme une relation

```
int 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.

dbplus_close ferme la relation `relation` , ouverte par dbplus_open .

7.21.11 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 .

7.21.12 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.

7.21.13 dbplus_erro() : Lit le code d'erreur généré par la dernière opération

int **dbplus_erro** (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_erro retourne le code d'erreur généré par la dernière opération DB++.

Voir aussi dbplus_errcode .

7.21.14 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 .

7.21.15 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_curr , dbplus_prev , dbplus_next et dbplus_last .

7.21.16 dbplus_flush() : Ecrit toutes les modifications apportées à une relation

```
int dbplus_flush ( 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_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.

7.21.17 dbplus_freealllocks() : Libère tous les verrous posés par le client

```
int dbplus_freealllocks ( 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_freealllocks libère tous les verrous posés par le client.

Voir aussi dbplus_getlock , dbplus_freelock et dbplus_freerlocks .

7.21.18 dbplus_freelock() : Lève un verrou en écriture sur une ligne

```
int dbplus_freelock ( 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_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](#) .

7.21.19 dbplus_freerlocks() : Lève tous les verrous sur les lignes d'une relation

```
int dbplus_freerlocks ( 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_freerlocks](#) lève tous les verrous sur les lignes de la relation ^{relation} .

Voir aussi [dbplus_getlock](#) , [dbplus_freelock](#) et [dbplus_freealllocks](#) .

7.21.20 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} . La 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_freerlocks](#) et [dbplus_freealllocks](#) .

7.21.21 dbplus_getunique() : Lit 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.

7.21.22 dbplus_info() : Encore non implémentée

```
int dbplus_info ( resource relation , string key , array )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Encore non implémentée.

7.21.23 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.

7.21.24 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.

7.21.25 dbplus_next() : Lit la ligne suivante dans la relation

```
int dbplus_next ( resource relation , array )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 table `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` .

7.21.26 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 de nom `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` .

7.21.27 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 précédente 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'erreurs db++.

Voir aussi [dbplus_first](#) , [dbplus_curr](#) , [dbplus_next](#) et [dbplus_last](#) .

7.21.28 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.

7.21.29 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 significations.

7.21.30 dbplus_rcrtextact() : Crée une copie exacte mais vide d'une relation

```
resource dbplus_rcrtextact ( 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_rcrtextact](#) 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.

7.21.31 dbplus_rcrtlike() : Crée une copie exacte d'une relation, sans les indexes

```
resource dbplus_rcrtlike ( string name , resource relation , int 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.

`dbplus_rcrtexact` crée une copie exacte de la relation `relation` , sous le nouveau nom de `name` , et 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.

7.21.32 dbplus_resolve() : Résoud les informations d'hôte pour une relation

```
int 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 de 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` .

7.21.33 dbplus_restorepos() : Pas encore implémenté

```
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.

7.21.34 dbplus_rkeys() : Spécifie la nouvelle clé primaire d'une relation

```
resource 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.

`dbplus_rkeys` 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.

7.21.35 `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.

`dbplus_ropen` 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 `dbplus_current` et `dbplus_next` peuvent être utilisées avec la relation retournée.

7.21.36 `dbplus_rquery()` : Performe une requête AQL locale

`int 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` performe une requête AQL brute, en utilisant l'interpréteur du client DB++. `dbplus_rquery` est plus rapide que `dbplus_aql` mais ne fonctionne que sur des données locales.

7.21.37 `dbplus_rename()` : Renomme une relation

`int dbplus_rename ( 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_rename` va changer le nom de la relation `relation` en `name` .

7.21.38 `dbplus_rsecindex()` : Crée un nouvel index secondaire pour une relation

`resource 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.

7.21.39 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`.

7.21.40 dbplus_rzap() : Supprime toutes les lignes d'une relation

```
int dbplus_rzap ( 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_rzap` supprime toutes les lignes de la relation `relation`.

7.21.41 dbplus_savepos() : Pas encore implémentée

```
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.

Note

Cette fonction n'est pas encore implémentée.

7.21.42 dbplus_setindex() : Pas encore implémentée

```
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.

Note

Cette fonction n'est pas encore implémentée.

7.21.43 dbplus_setindexbynumber() : Pas encore implémentée

```
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.

Note

Cette fonction n'est pas encore implémentée.

7.21.44 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.

Note

Cette fonction n'est pas encore implémentée.

7.21.45 dbplus_tcl() : Exécute du code TCL coté serveur

```
int 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](#) .

7.21.46 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 `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.

7.21.47 dbplus_undo() : Pas encore implémentée

```
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.

Note

Cette fonction n'est pas encore implémentée.

7.21.48 dbplus_undoprepere() : Pas encore implémentée

```
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.

Note

Cette fonction n'est pas encore implémentée.

7.21.49 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` .

7.21.50 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 `relation` .

7.21.51 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` .

7.21.52 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 `relation` , qui empêchera les autres clients d'accéder à la relation, et même de la lire.

Voir aussi `dbplus_xunlockrel` .

7.21.53 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 `relation` , grâce à la fonction `dbplus_xlockrel` .

7.22 Direct IO

7.22.1 Introduction

PHP supporte les entrées/sorties directes, comme décrites dans le Standard Posixd (Section 6). Cela permet de réaliser des accès en lecture écriture à un niveau inférieur aux fonctions proposées par le langage C (fopen, fread...).

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.22.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.22.3 Installation

Pour faire fonctionner ces fonctions, vous devez configurer PHP avec l'option `--enable-dio`.

7.22.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.22.5 Types de ressources

Cette extension définit un type de ressource : un descripteur de fichier, retourné par [`dio\_open`](#).

7.22.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [dio_close](#)
- [dio_fcntl](#)
- [dio_open](#)
- [dio_read](#)
- [dio_seek](#)
- [dio_stat](#)
- [dio_tcsetattr](#)
- [dio_truncate](#)
- [dio_write](#)

7.22.8 dio_fcntl() : Exécute une fonction fichier de langage C

`mixed dio_fcntl ( resource fd , int cmd , mixed arg )`

`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`.

`arg` est un tableau associatif lorsque `cmd` vaut `F_SETLK` ou `F_SETLLW`, avec les entrées suivantes :

- "start" – offset de début de verrou
- "length" – taille de la surface verrouillée. zéro indique la fin du fichier.
- "wenth" – `l_start` est relatif à : can be `SEEK_SET`, `SEEK_END` et `SEEK_CUR`
- "type" – type de verrou : peut être `F_RDLCK` (verrou en lecture), `F_WRLCK` (verrou en écriture) ou `F_UNLCK` (déverrouillage)

`cmd` peut être l'une des opérations suivantes :

- `F_SETLK` – Le verrou est posé ou levé. Si le verrou appartient à un autre utilisateur, `dio_fcntl` retourne `-1`.
- `F_SETLKW` – identique à `F_SETLK`, mais si le verrou appartient à quelqu'un d'autre, `dio_fcntl` attend la levée du verrou.
- `F_GETLK` – `dio_fcntl` retourne un tableau associatif (comme décrit ci-dessus) si un autre utilisateur empêche la pose d'un verrou. S'il n'y a aucune empêchement, l'entrée "type" prendra la valeur `F_UNLCK`.
- `F_DUPFD` – trouve les numéros de ressource de fichier les plus petits, disponibles, inférieurs ou égales à `arg` et les retourne.

7.22.9 dio_open() : Ouvre un nouveau fichier

`resource dio_open ( string filename , int flags , int mode )`

`dio_open` ouvre un accès à un fichier, et retourne une ressource de fichier, ou bien `-1` si une erreur survient. Si `flags` vaut `O_CREAT`, le troisième paramètre optionnel `mode` indiquera le mode du fichier (les permissions). Le paramètre `flags` peut prendre l'une des valeurs suivantes :

- `O_RDONLY` – ouvre un fichier en lecture
- `O_WRONLY` – ouvre un fichier en écriture
- `O_RDWR` – ouvre un fichier en lecture et écriture

Le paramètre `flags` peut aussi inclure une combinaison des options suivantes :

- `O_CREAT` – crée le fichier, s'il n'existe pas
- `O_EXCL` – si `O_CREAT` et `O_EXCL` sont utilisés, `dio_open` échouera si le fichier existe déjà.
- `O_TRUNC` – si le fichier existe, et qu'il est ouvert en écriture, le fichier sera réduit à la taille nulle.
- `O_APPEND` – les écritures se font à partir de la fin du fichier.
- `O_NONBLOCK` – active le mode non bloquant

7.22.10 dio_read() : Lit des octets dans un fichier

`string dio_read ( resource fd , int n )`

dio_read lit et retourne ⁿ octets dans le fichier représenté par la ressource ^{fd}. Si ⁿ est omis, dio_read lira 1ko de données, et les retournera.

7.22.11 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 ^{fd}. Le paramètre ^{whence} spécifie comment la position ^{pos} doit être interprétée :

- SEEK_SET – ^{pos} est spécifiée en byte à partir du début du fichier.
- SEEK_CUR – ^{pos} est un nombre d'octets à partir de la position courante. Ce nombre peut être positif ou négatif.
- SEEK_END – ^{pos} est un nombre de caractères à partir de la fin du fichier. Un nombre négatif indique une position à l'intérieur du fichier courant; un nombre positif indique une extension du fichier. Si vous utilisez une position au delà de la fin du fichier, et que vous écrivez réellement des données, vous allez aggrandir le fichier de ^{pos} caractères.

7.22.12 dio_stat() : Lit des informations sur le fichier

```
array dio_stat ( resource fd )
```

dio_stat retourne les informations sur le fichier représenté par la ressource ^{fd}. dio_stat retourne un tableau associatif, avec les entrées suivantes :

- "device" – device
- "inode" – inode
- "mode" – mode
- "nlink" – nombre de liens
- "uid" – user id
- "gid" – group id
- "device_type" – type de device (si inode device)
- "size" – taille en octets
- "blocksize" – taille de block
- "blocks" – nombre de block allouée
- "atime" – date de dernier accès
- "mtime" – date de dernière modification
- "ctime" – date de dernier changement

En cas d'erreur, dio_stat retourne ^{NULL}.

7.22.13 dio_tcsetattr() : Modifie les attributs du terminal et le taux de baud du port série

```
dio_tcsetattr ( resource fd , array options )
```

dio_tcsetattr modifie les attributs du terminal et le taux de baud du port série, en les remplaçant par ^{options}. ^{fd} est une ressource déjà ouverte. Les options actuelles sont :

- 'baud' – taux de baud du port : peut être 38400,19200,9600,4800,

2400,1800,1200,600,300,200,150,134,110,75 ou 50. Par défaut, c'est 9600.

- 'bits' – bits de données – peut être 8,7,6 ou 5. Par défaut, c'est 8.
- 'stop' – bits de stop – peut être 1 ou 2. Par défaut, c'est 1.
- 'parity' – peut être 0, 1 ou 2. Par défaut, c'est 0.

Modifier le taux de baud du port série

```
<?php

$fd = dio_open('/dev/ttyS0', O_RDWR | O_NOCTTY | O_NONBLOCK);

dio_fcntl($fd,F_SETFL, O_SYNC );

dio_tcsetattr($fd, array(
    'baud' => 9600,
    'bits' => 8,
    'stop' =>1,
    'parity' => 0
));

while (1) {

    $data = dio_read($fd,256);

    if ($data) {
        echo $data;
    }
}

?>
```

Note

Cette fonction a été introduite en PHP 4.3.0.

7.22.14 dio_truncate() : Tronque un fichier

bool **dio_truncate** (resource *fd* , int *offset*)

dio_truncate tronque le fichier représenté par la ressource ^{fd} à la taille maximale de ^{offset} octets. Si le fichier était plus grand, les données supplémentaires sont perdues. Si le fichier était plus petit, le comportement est non spécifié : le document peut être inchangé, ou bien agrandi. Dans ce dernier cas, l'extension se fait avec des octets nuls. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.22.15 dio_write() : Ecrit des données dans le fichier

int **dio_write** (resource *fd* , string *data* , int *len*)

dio_write écrit jusqu'à ^{len} octets issues de la variable ^{data}, dans le fichier représenté par la ressource ^{fd}. Si ^{len} n'est pas spécifié, dio_write utilise la totalité de la variable ^{data}. dio_write retourne le nombre d'octets écrits dans le fichier ^{fd}.

7.23 Accès aux dossiers

7.23.1 Introduction

7.23.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.23.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.23.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.23.5 Types de ressources

7.23.6 Constantes prédefinies

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 )
```

7.23.7 Voir aussi

Pour des fonctions connexes telles que [dirname](#) , [is_dir](#) , [mkdir](#) et [rmdir](#) , voyez donc les fonctions [Système de fichiers](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Voir aussi](#)
- [chdir](#)
- [chroot](#)
- [closedir](#)
- [dir](#)
- [getcwd](#)
- [opendir](#)
- [readdir](#)

- [rewinddir](#)
- [scandir](#)

7.23.9 chroot() : Change le dossier racine

bool **chroot** (string directory)

[chroot](#) change la racine du script en cours, et la remplace par `directory`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec..

Note

Il n'est pas conseillé d'utiliser cette fonction sur un site web, car il n'est pas possible de restaurer la racine à sa valeur initiale à la fin de la requête. Cette fonction n'est viable que lorsque PHP est utilisé en CGI.

Note

Cette fonction n'est pas implémentée sous Windows.

7.23.10 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](#).

7.23.11 dir() : Classe dossier

new **dir** (string directory)

Un mécanisme pseudo-objet permet la lecture d'un dossier. L'argument `directory` doit être 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 permettent de lire, remettre à zéro et fermer le dossier.

Exemple avec [dir](#)

```
<?php
$d = dir("/etc");
echo "Pointeur: " . $d->handle."<BR>\n";
echo "Chemin: " . $d->path."<BR>\n";
while($entry=$d->read()) {
    echo $entry."<BR>\n";
}
$d->close();
?>
```

7.23.12 getcwd() : Retourne le dossier de travail

string **getcwd** (void)

[getcwd](#) retourne le nom du dossier courant.

Voir aussi [chdir](#).

7.23.13 opendir() : Ouvre un dossier, et récupère un pointeur dessus.

resource **opendir** (string path)

opendir retourne un pointeur sur un dossier pour être utilisé avec les fonctions closedir, readdir et rewinddir.

Si le paramètre `path` n'est pas un dossier valide, ou si le dossier ne peut être accédé pour des raisons de permissions ou des erreurs liées au système de fichiers, opendir retourne `FALSE` et génère une erreur PHP. Vous pouvez supprimer cette erreur en ajoutant `@` avant le nom de la fonction.

Exemple avec opendir

```
<?php
$dir = "/tmp/";

// Ouvre un dossier bien connu, et liste tous les fichiers
if (is_dir($dir)) {
    if ($dh = opendir($dir)) {
        while (($file = readdir($dh)) !== false) {
            print "fichier : $file : type: " . filetype($dir . $file) . "\n";
        }
        closedir($dh);
    }
}
?>
```

Voir aussi is_dir, readdir et dir

7.23.14 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 n'importe quel ordre.

Liste tous les fichiers du dossier courant

```
<?php
$handle=opendir('.');
echo "Pointeur de dossier : $handle\n";
echo "Fichiers :\n";
while ($file = readdir($handle)) {
    echo "$file\n";
}
closedir($handle);
?>
```

Notez que readdir retournera aussi les dossiers "." et "..". Si vous ne les voulez pas, supprimez les simplement :

Liste tous les fichiers du dossier courant, sauf "." et ".."

```
<?php
$handle=opendir('.');
while ($file = readdir($handle)) {
    if ($file != "." && $file != "..") {
        echo "$file\n";
    }
}
closedir($handle);
?>
```

Voir aussi is_dir et glob.

7.23.15 rewinddir() : Retourne à la première entrée du dossier

void **rewinddir** (resource dir_handle)

rewinddir retourne à la première entrée du dossier identifié par `dir_handle` : le prochain fichier lu sera le premier.

7.23.16 scandir() : Liste les fichiers et dossier dans un dossier

array **scandir** (string directory , int sorting_order)

scandir retourne un `array` de fichier et dossier, issus de `directory`. Si `directory` n'est pas un dossier, le booléen `FALSE` est retourné, et une erreur de niveau `E_WARNING` est générée.

Par défaut, les fichiers sont donnés par ordre alphabétique. Si le paramètre optionnel `sorting_order` est utilisé avec la valeur 1, le tri sera fait en ordre alphabétique inverse.

Listage simple de dossier avec **scandir**

```
<?php
$dir = '/tmp';
$files1 = scandir($dir);
$files2 = scandir($dir, 1);

print_r($files1);
print_r($files2);

/* Affichera ceci :
Array
(
    [0] => .
    [1] => ..
    [2] => bar.php
    [3] => foo.txt
    [4] => somedir
)
Array
(
    [0] => somedir
    [1] => foo.txt
    [2] => bar.php
    [3] => ..
    [4] => .
)
*/
?>
```

Alternatives à **scandir**

```
<?php
$dir = "/tmp";
$dh = opendir($dir);
while (false !== ($filename = readdir($dh))) {
    $files[] = $filename;
}

sort($files);

print_r($files);

rsort($files);

print_r($files);

/* Affichera ceci :
Array
(
    [0] => .
    [1] => ..
    [2] => bar.php
    [3] => foo.txt
    [4] => somedir
)
Array
(
```

```
[0] => somedir
[1] => foo.txt
[2] => bar.php
[3] => ..
[4] => .
)
*/
?>
```

Voir aussi [opendir](#) , [readdir](#) , [glob](#) , [is_dir](#) et [sort](#) .

7.24 DOM XML

7.24.1 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.

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 `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 `domxml_dumpmem` 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 `domxml_add_node` , `domxml_set_attribute` , etc... et finalement `domxml_dumpmem` .

7.24.2 Pré-requis

Cette extension utilise la librairie GNOME XML . Téléchargez la, puis installez-la. Vous aurez besoin de la version libxml-2.4.14 ou plus récent. Pour utiliser les fonctionnalités DOM XSLT vous pouvez utiliser la librairie libxslt library et les progrès des EXSLT (<http://www.exslt.org/>). Téléchargez et installez ces librairies si vous envisagez d'exploiter ces fonctionnalités. Vous aurez besoin d'au moins libxslt-1.0.18.

7.24.3 Installation

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'installatoir de libxslt. Ajoutez `--with-dom-exslt=[DIR]` pour inclure le support DOM EXSLT. DIR est le dossier d'installatoir de libxsl.

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier les librairies **libxml2.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).

7.24.4 Deprecated functions

Il y a un groupe de fonction 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
<code>xmldoc</code>	<code>domxml_open_mem</code>
<code>xmldocfile</code>	<code>domxml_open_file</code>
<code>domxml_new_xmldoc</code>	<code>domxml_new_doc</code>
<code>domxml_dump_mem</code>	<code>domdocument_dump_mem</code>
<code>domxml_dump_mem_file</code>	<code>domdocument_dump_file</code>
<code>DomDocument_dump_mem_file</code>	<code>domdocument_dump_file</code>
<code>DomDocument_add_root</code>	<code>domdocument_create_element</code> suivie de <code>domnode_append_child</code>
<code>DomDocument_dtd</code>	<code>domdocument_doctype</code>
<code>DomDocument_root</code>	<code>domdocument_document_element</code>
<code>DomDocument_children</code>	<code>domnode_child_nodes</code>
<code>DomDocument_imported_node</code>	No replacement.
<code>DomNode_add_child</code>	Create a new node with e.g. <code>domdocument_create_element</code> puis ajoutez-le avec <code>domnode_append_child</code> .
<code>DomNode_children</code>	<code>domnode_child_nodes</code>
<code>DomNode_parent</code>	<code>domnode_parent_node</code>
<code>DomNode_new_child</code>	Create a new node with e.g. <code>domdocument_create_element</code> puis ajoutez-le avec <code>domnode_append_child</code> .
<code>DomNode_set_content</code>	Create a new node with e.g. <code>domdocument_create_text_node</code> puis ajoutez-le avec <code>domnode_append_child</code> .
<code>DomNode_get_content</code>	Le contenu est juste un noeud de texte, et est accessible via <code>domnode_child_nodes</code> .
<code>DomNode_set_content</code>	Le contenu est juste un noeud de texte, et est accessible via <code>domnode_append_child</code> .

7.24.5 Constantes prédefinies

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
<code>XML_ELEMENT_NODE (integer)</code>	1	Node is an element
<code>XML_ATTRIBUTE_NODE (integer)</code>	2	Node is an attribute
<code>XML_TEXT_NODE (integer)</code>	3	Node is a piece of text
<code>XML_CDATA_SECTION_NODE (integer)</code>	4	
<code>XML_ENTITY_REF_NODE (integer)</code>	5	
<code>XML_ENTITY_NODE (integer)</code>	6	Node is an entity like
<code>XML_PI_NODE (integer)</code>	7	Node is a processing instruction
<code>XML_COMMENT_NODE (integer)</code>	8	Node is a comment
<code>XML_DOCUMENT_NODE (integer)</code>	9	Node is a document
<code>XML_DOCUMENT_TYPE_NODE (integer)</code>	10	
<code>XML_DOCUMENT_FRAG_NODE (integer)</code>	11	
<code>XML_NOTATION_NODE (integer)</code>	12	
<code>XML_GLOBAL_NAMESPACE (integer)</code>	1	
<code>XML_LOCAL_NAMESPACE (integer)</code>	2	
<code>XML_HTML_DOCUMENT_NODE (integer)</code>		
<code>XML_DTD_NODE (integer)</code>		
<code>XML_ELEMENT_DECL_NODE (integer)</code>		
<code>XML_ATTRIBUTE_DECL_NODE (integer)</code>		
<code>XML_ENTITY_DECL_NODE (integer)</code>		
<code>XML_NAMESPACE_DECL_NODE (integer)</code>		
<code>XML_ATTRIBUTE_CDATA (integer)</code>		
<code>XML_ATTRIBUTE_ID (integer)</code>		
<code>XML_ATTRIBUTE_IDREF (integer)</code>		
<code>XML_ATTRIBUTE_IDREFS (integer)</code>		
<code>XML_ATTRIBUTE_ENTITY (integer)</code>		
<code>XML_ATTRIBUTE_NMTOKEN (integer)</code>		
<code>XML_ATTRIBUTE_NMTOKENS (integer)</code>		
<code>XML_ATTRIBUTE_ENUMERATION (integer)</code>		
<code>XML_ATTRIBUTE_NOTATION (integer)</code>		
<code>XP_PATH_UNDEFINED (integer)</code>		
<code>XP_PATH_NODESET (integer)</code>		
<code>XP_PATH_BOOLEAN (integer)</code>		
<code>XP_PATH_NUMBER (integer)</code>		
<code>XP_PATH_STRING (integer)</code>		
<code>XP_PATH_POINT (integer)</code>		
<code>XP_PATH_RANGE (integer)</code>		
<code>XP_PATH_LOCATIONSET (integer)</code>		
<code>XP_PATH_USERS (integer)</code>		
<code>XP_PATH_NUMBER (integer)</code>		

7.24.6 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 qu 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	Classes parents
<code>DomAttribute</code>	<code>DomNode</code>
<code>DomCData</code>	<code>DomNode</code>
<code>DomComment</code>	<code>DomCData</code> : <code>DomNode</code>
<code>DomDocument</code>	<code>DomNode</code>
<code>DomDocumentType</code>	<code>DomNode</code>
<code>DomElement</code>	<code>DomNode</code>
<code>DomEntity</code>	<code>DomNode</code>
<code>DomEntityReference</code>	<code>DomNode</code>
<code>DomProcessingInstruction</code>	<code>DomNode</code>
<code>DomText</code>	<code>DomCData</code> : <code>DomNode</code>
<code>Parser</code>	Actuellement, toujours appelée <code>DomParser</code>
<code>XPathContext</code>	

Nom de la méthode	Nom de la fonction	Remarques
<code>doctype</code>	<code>domdocument doctype</code>	
<code>document_element</code>	<code>domdocument document element</code>	
<code>create_element</code>	<code>domdocument create element</code>	
<code>create_text_node</code>	<code>domdocument create text node</code>	
<code>create_comment</code>	<code>domdocument create comment</code>	
<code>create_cdata_section</code>	<code>domdocument create cdata section</code>	
<code>create_processing_instruction</code>	<code>domdocument create processing instruction</code>	
<code>create_attribute</code>	<code>domdocument create attribute</code>	
<code>create_entity_reference</code>	<code>domdocument create entity reference</code>	
<code>get_elements_by_tagname</code>	<code>domdocument get elements by tagname</code>	
<code>get_element_by_id</code>	<code>domdocument get element by id</code>	
<code>dump_mem</code>	<code>domdocument dump mem</code>	not DOM standard
<code>dump_file</code>	<code>domdocument dump file</code>	not DOM standard
<code>html_dump_mem</code>	<code>domdocument html dump mem</code>	Non standard DOM
<code>xpath_init</code>	<code>xpath_init</code>	Non standard DOM
<code>xpath_new_context</code>	<code>xpath_new_context</code>	Non standard DOM
<code>xptr_new_context</code>	<code>xptr_new_context</code>	Non standard DOM

Nom de la méthode	Nom de la fonction	Remarques
<code>tagname</code>	<code>domelement tagname</code>	
<code>get_attribute</code>	<code>domelement get attribute</code>	
<code>set_attribute</code>	<code>domelement set attribute</code>	
<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>	

domnode_first_child	
domnode_child_nodes	
domnode_previous_sibling	
domnode_next_sibling	
domnode_parent_node	
domnode_owner_document	
domnode_insert_before	
domnode_append_child	
domnode_append_sibling	Non standard DOM. Cette fonction émule le comportement précédent de domnode_append_child .
domnode_remove_child	
domnode_has_child_nodes	
domnode_has_attributes	
domnode_clone_node	
domnode_attributes	
domnode_unlink_node	Non standard DOM
domnode_replace_node	Non standard DOM
domnode_set_content	Non standard DOM, deprecated
domnode_get_content	Non standard DOM, deprecated
domnode_dump_node	Non standard DOM
domnode_is_blank_node	Non standard DOM

Nom de la méthode		Remarques
name	domattribute_name	
value	domattribute_value	
specified	domattribute_specified	

Nom de la méthode	Nom de la fonction	Remarques
target	domprocessinginstruction_target	
data	domprocessinginstruction_data	

Nom de la méthode	Nom de la fonction	Remarques
add_chunk	parser_add_chunk	
end	parser_end	

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.

7.24.7 Exemples

De nombreux exemples de cette partie requiert 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 mis dans la section suivante. Alternativement, vous voulez créer un document XML et le lire avec la fonction [domdocument_open_file](#) .

Fichier d'inclusion example.inc pour la chaîne XML d'exemple
<pre> <?php \$xmlstr = " <chapter language='en'><title language='en'>Title</title> <para language='ge'> &sp; <informatable ID='findme' language='&sp;'> <tgroup cols='3'> </pre>

```

<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>";
?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Deprecated functions](#)
- [Constantes prédefinies](#)
- [Classes](#)
- [Exemples](#)
- [domattribute->name](#)
- [domattribute->specified](#)
- [domattribute->value](#)
- [domdocument->add_root \[obsolète\]](#)
- [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_attributess](#)
- [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](#)
- [domxsltstylesheet->process](#)
- [domxsltstylesheet->result_dump_file](#)
- [domxsltstylesheet->result_dump_mem](#)
- [domxml_new_doc](#)
- [domxml_open_file](#)
- [domxml_open_mem](#)
- [domxml_version](#)
- [domxml_xmltree](#)
- [domxml_xslt_stylesheet_doc](#)
- [domxml_xslt_stylesheet_file](#)
- [domxml_xslt_stylesheet](#)
- [xpath_eval_expression](#)
- [xpath_eval](#)
- [xpath_new_context](#)
- [xptr_eval](#)
- [xptr_new_context](#)

7.24.9 domattribute->specified() : Vérifie si un attribut est spécifié

`bool domattribute->specified ( void )`

[domattribute->specified](#) vérifie le standard DOM pour une explication détaillée.

7.24.10 domattribute->value() : Retourne la valeur d'un attribut

`bool domattribute->value ( void )`

domattribute->value retourne la valeur de l'attribut.

Voir aussi [domattribute_name](#).

7.24.11 domdocument->add_root [obsolète]() : Ajoute un noeud

resource **domdocument->add_root** (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.

domdocument->add_root ajoute un noeud racine au document dom, et retourne le nouveau noeud. Le nom de l'élément créé est Adds a root element node to a dom document and returns the new node. The element name est fourni par le paramètre `name`.

Création d'un entête de document HTML

```
<?php
$doc = domxml_new_doc("1.0");
$root = $doc->add_root("HTML");
$head = $root->new_child("HEAD", "");
$head->new_child("TITLE", "Hier der Titel");
echo htmlentities($doc->dump_mem());
?>
```

7.24.12 domdocument->create_attribute() : Create new attribute

object **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](#).

7.24.13 domdocument->create_cdata_section() : Crée un nouveau noeud CDATA

string **domdocument->create_cdata_section** (string content)

[domdocument->create_cdata_section](#) retourne une nouvelle instance de la classe `DomCData`. Le contenu du CDATA sera `content`. 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_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](#), [domnode_insert_before](#).

7.24.14 domdocument->create_comment() : Crée un nouveau noeud de commentaire

object **domdocument->create_comment** (string content)

domdocument->create_comment retourne une nouvelle instance de la classe `DomComment`. Le contenu du commentaire sera `content`. 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_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.

7.24.15 domdocument->create_element_ns() : Crée un nouvel élément avec un espace de noms associé

object **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`. Si 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. 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_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, domnode insert before.

7.24.16 domdocument->create_element() : Crée un nouveau noeud élément

object **domdocument->create_element** (string name)

domdocument->create_element retourne une nouvelle instance de la classe `DomElement`. Le nom de balise l'élément sera `content`. 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_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, domnode insert before.

7.24.17 domdocument->create_entity_reference() : Crée un objet DomEntityReference

object **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 comme paramètre `content` de la fonction. Ce noeud ne sera pas inséré dans le document tant qu'il n'y sera pas ajouté par la fonction 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.

7.24.18 domdocument->create_processing_instruction() : Crée un nouveau noeud de traitement d'insctructions (PI)

string **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`. 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_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.

7.24.19 domdocument->create_text_node() : Crée un nouveau noeud de texte

object **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, a 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.

7.24.20 domdocument->doctype() : Retourne le type de document

object **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`.

7.24.21 domdocument->document_element() : Retourne le noeud racine d'un document DOM XML

object **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);
?>
```

7.24.22 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 entête 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.

7.24.23 domdocument->dump_mem() : Convertit l'arbre XML interne en une chaîne de caractères

string **domdocument->dump_mem** (bool format , 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.

`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 entête 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`.

7.24.24 domdocument->get_element_by_id() : Recherche un élément avec son identifiant

object **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` si 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`.

7.24.25 domdocument->get_elements_by_tagname() : get_elements_by_tagname()

array **domdocument->get_elements_by_tagname** (string name)

Voir aussi `domdocument_add_root`.

7.24.26 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 entête de document HTML avec `domdocument->html_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->html_dump_mem());
echo "</PRE>";
?>
```

Voir aussi `domdocument_dump_file` et `domdocument_html_dump_mem`.

7.24.27 domdocument->xinclude() : Substitute XIncludes dans un objet domdocument

int `domdocument->xinclude ( 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.

7.24.28 domdocumenttype->entities() : Retourne la liste des entités

array `domdocumenttype->entities ( void )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.29 domdocumenttype->internal_subset() : Retourne le sous ensemble interne (internal subset)

bool `domdocumenttype->internal_subset ( void )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.30 domdocumenttype->name() : Retourne le nom du type de document

string `domdocumenttype->name ( void )`

`domdocumenttype->name` retourne le nom du type de document.

7.24.31 domdocumenttype->notations() : Retourne la liste des notations

array `domdocumenttype->notations ( void )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.32 domdocumenttype->public_id() : Retourne l'identifiant public du type de document

string `domdocumenttype->public_id ( void )`

`domdocumenttype->public_id` retourne l'identifiant public du type de document.

L'exemple ci-dessous n'affichera rien.

Lire l'identifiant public avec DOM XML

```
<?php
include("exemple.inc");

if(!$dom = domxml_open_mem($xmlstr)) {
    echo "Erreur lors de l'analyse du document\n";
    exit;
}

$doctype = $dom->doctype();
echo $doctype->public_id();
?>
```

7.24.33 domdocumenttype->system_id() : Retourne l'identifiant système du type de document

string `domdocumenttype->system_id ( void )`

`domdocumenttype->system_id` retourne l'identifiant système du type de document.

L'exemple ci-dessous affichera '/share/sgml/Norman_Walsh/db3xml10/db3xml10.dtd'.

Lire un identifiant système DOM XML

```
<?php
include("exemple.inc");

if(!$dom = domxml_open_mem($xmlstr)) {
    echo "Erreur durant l'analyse du document\n";
    exit;
}

$doctype = $dom->doctype();
echo $doctype->system_id();
?>
```

7.24.34 domelement->get_attribute_node() : Retourne la valeur d'un attribut

`object domelement->get_attribute_node ( object attr )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.35 domelement->get_attribute() : Retourne la valeur d'un attribut

`object domelement->get_attribute ( string name )`

`domelement->get_attribute` retourne la valeur de l'attribut de nom `name` .

(PHP >= 4.3 uniquement) Si aucun attribut de nom `name` n'est trouvé, une chaîne vide est retournée.

Voir aussi [domelement_set_attribute](#) .

7.24.36 domelement->get_elements_by_tagname() : Lit les éléments par nom de balise

`bool domelement->get_elements_by_tagname ( string name )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.37 domelement->has_attribute() : Vérifie si un attribut existe

`bool domelement->has_attribute ( string name )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.38 domelement->remove_attribute() : Supprime un attribut

`bool domelement->remove_attribute ( string name )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.39 domelement->set_attribute() : Ajoute un nouvel attribut

`bool domelement->set_attribute ( string name , string value )`

[domelement->set_attribute](#) ajouter l'attribut de nom `name` , avec la valeur `value` . Si cet attribut n'existe pas, il sera créé.

Créer un attribut avec DOM XML

```
<?php
$doc = domxml_new_doc("1.0");
$node = $doc->create_element("para");
$newnode = $doc->append_child($node);
$newnode->set_attribute("align", "left");
?>
```

Voir aussi [domelement_get_attribute](#) .

7.24.40 domelement->tagname() : Retourne le nom d'un élément

string **domelement->tagname** (void)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.41 domnode->add_namespace() : Ajoute une déclaration d'espace de nom à un noeud

bool **domnode->add_namespace** (string uri , string prefix)

Voir aussi [domdocument_create_element_ns](#) et [domnode_set_namespace](#)

7.24.42 domnode->append_child() : Ajoute un nouveau fils new child at the end of the children

object **domnode->append_child** (object newnode)

[domnode->append_child](#) ajoute le noeud `newnode` à une liste de fils existante, ou bien crée une nouvelle liste de fils. Un noeud fils peut être créé avec les fonctions [domdocument_create_element](#) , [domdocument_create_text](#) , etc... ou bien en utilisant simplement 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.

(PHP >= 4.3) Le nouveau fils `newnode` est d'abord déconnecté de son contexte, si il existait dans le document. Par conséquent, le noeud est déplacé et ce n'est plus une copie. C'est le comportement compatible avec les spécifications du consortium W3C. Si vous voulez dupliquer de grandes parties d'un document XML, utiliser la fonction [domnode->clone_node](#) avant de l'ajouter.

l'exemple suivant ajoute un nouveau noeud dans un document, et modifie son attribut "align" à "left".

ajouter un noeud fils avec dom xml

```
<?php
$doc = domxml_new_doc("1.0");
$node = $doc->create_element("para");
$newnode = $doc->append_child($node);
$newnode->set_attribute("align", "left");
```

```
?>
```

l'exemple ci-dessus peut être aussi écrit comme ceci :

ajouter un noeud fils avec dom xml (méthode 2)

```
<?php
$doc = domxml_new_doc("1.0");
$node = $doc->create_element("para");
$node->set_attribute("align", "left");
$newnode = $doc->append_child($node);
?>
```

un exemple plus complexe est celui ci-dessous. il commence par rechercher un élément, le duplique avec ses fils, et l'ajoute comme frère. finalement, un nouvel attribut est ajouté, et le document complet est publié.

Ajouter un noeud fils avec DOM XML (exemple 3)

```
<?php
include("exemple.inc");

if(!$dom = domxml_open_mem($xmlstr)) {
    echo "erreur lors de l'analyse du document\n";
    exit;
}

$elements = $dom->get_elements_by_tagname("informaltable");
print_r($elements);
$element = $elements[0];

$parent = $element->parent_node();
$newnode = $parent->append_child($element);
$children = $newnode->children();
$attr = $children[1]->set_attribute("align", "left");

echo "<pre>";
$xmlfile = $dom->dump_mem();
echo htmlentities($xmlfile);
echo "</pre>";
?>
```

l'exemple ci-dessus peut aussi être réalisé avec la fonction [domnode_insert_before](#) au lieu de [domnode_append_child](#).

Voir aussi [domnode_insert_before](#) et [domnode_clone_node](#).

7.24.43 domnode->append_sibling() : Ajoute un frère à un noeud

object **domnode->append_sibling** (object 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](#).

7.24.44 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.

7.24.45 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.

7.24.46 domnode->clone_node() : Clone un noeud

object **domnode->clone_node** (void)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.47 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.

7.24.48 domnode->first_child() : Retourne le premier fils du noeud

bool **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.

7.24.49 domnode->get_content() : Lit le contenu du noeud

string **domnode->get_content** (void)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.50 domnode->has_attributess() : 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 .

7.24.51 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 .

7.24.52 domnode->insert_before() : Insère un nouveau noeud fils

object **domnode->insert_before** (object newnode , object 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 .

7.24.53 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.

7.24.54 domnode->last_child() : Retourne le dernier fils du noeud

object **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 .

7.24.55 domnode->next_sibling() : Retourne le noeud frère suivant

object **domnode->next_sibling** (void)

domnode->next_sibling retourne le prochain noeud fils du noeud courant. Si 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

```
<?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();
}
?>
```

Voir aussi domnode_previous_sibling .

7.24.56 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 signification 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

7.24.57 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.

7.24.58 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

7.24.59 domnode->owner_document() : Retourne le document auquel appartient ce noeud DOM XML

object **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 <pre> <?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); ?> </pre>

Voir aussi [domnode_insert_before](#) .

7.24.60 domnode->parent_node() : Retourne le père d'un noeud DOM XML

object **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 <pre> <?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); ?> </pre>

7.24.61 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.

7.24.62 domnode->previous_sibling() : Retourne le frère précédent d'un noeud

object **domnode->previous_sibling** (void)

[domnode->previous_sibling](#) retourne le frère précédent d'un noeud. Si 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](#) .

7.24.63 domnode->remove_child() : Supprime un fils de la liste des noeuds fils

object **domnode->remove_child** (object 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](#) .

7.24.64 domnode->replace_child() : Remplace un noeud fils

object **domnode->replace_child** (object oldnode , object 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](#) .

7.24.65 domnode->replace_node() : Remplace un noeud DomXML

object **domnode->replace_node** (object newnode)

(PHP 4.2) [domnode->replace_node](#) remplace un noeud par `newnode` . Avant le remplacement, `newnode` est copié si 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](#) .

7.24.66 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.

7.24.67 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 .

7.24.68 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} . Si 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 prendra le préfixe fourni dans le paramètre optionnel ^{prefix} ou bien en générera un aléatoirement.

Voir aussi domdocument_create_element_ns et domnode_add_namespace

7.24.69 domnode->unlink_node() : Efface un noeud

object domnode->unlink_node (void)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.70 domprocessinginstruction->data() : Retourne les données du script

string domprocessinginstruction->data (void)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.71 domprocessinginstruction->target() : Retourne la cible du script

string domprocessinginstruction->target (void)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.72 domxsltstylesheet->process() : Applique une transformation XSLT à un objet domdocument

object **domxsltstylesheet->process** (object domdocument , array xslt_parameters , bool param_is_xpath)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

Voir aussi [domxml_xslt_stylesheet](#) , [domxml_xslt_stylesheet_file](#) et [domxml_xslt_stylesheet_doc](#) .

7.24.73 domxsltstylesheet->result_dump_file() : Ecrit le résultat d'une transformation XSLT dans un fichier

string **domxsltstylesheet->result_dump_file** (object domdocument , 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.

Cette fonction est uniquement disponible depuis PHP 4.3.

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.

[domxsltstylesheet->result_dump_file](#) , au contraire, utilise <xsl:output method="html|text"> et les autres directives de contrôle des sorties. Voyez l'exemple pour plus de détails.

Sauver le résultat d'une transformation XSLT dans un fichier

```
<?php
$filename = "stylesheet.xml";
$xml_doc = domxml_open_file("data.xml");
$xslt_doc = domxml_xslt_stylesheet_file($filename);
$result = $xslt_doc->process($xml_doc);
print $xslt_doc->result_dump_file($result, "filename");
?>
```

Voir aussi [domxml_xslt_result_dump_mem](#) et [domxml_xslt_process](#) .

7.24.74 domxsltstylesheet->result_dump_mem() : Ecrit le résultat d'une transformation XSLT dans une chaîne

string **domxsltstylesheet->result_dump_mem** (object domdocument)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Cette fonction est uniquement disponible depuis PHP 4.3.

Comme `domstylesheet->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. `domstylesheet->result_dump_mem`, au contraire, utilise `<xsl:output method="html|text">` et les autres directives de contrôle des sorties. Voyez l'exemple pour plus de détails.

Sauver le résultat d'une transformation XSLT dans une chaîne

```
<?php
$filename = "stylesheet.xsl";
$xml_doc = domxml_open_file("data.xml");
$xsl_doc = domxml_xslt_stylesheet_file($filename);
$result = $xsl_doc->process($xml_doc);
print $xsl_doc->result_dump_mem($result);
?>
```

Voir aussi [domxml_xslt_result_dump_file](#) et [domxml_xslt_process](#).

7.24.75 domxml_new_doc() : Crée un document XML vide

object **domxml_new_doc** (string version)

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[domxml_new_doc](#) crée un document XML et le retourne.

Voir aussi [domdocument_add_root](#).

7.24.76 domxml_open_file() : Crée un objet DOM à partir d'un fichier XML

object **domxml_open_file** (string filename)

domxml_open_file analyse le document XML contenu dans le fichier `filename` et retourne un objet de classe "Dom document", avec les propriétés telles que listées ci-dessus. Le fichier est accédé en lecture seule.

Ouvrir un document XML à partir d'un fichier

```
<?php
if(!$dom = domxml_open_file("exemple.xml")) {
    echo "Erreur lors de l'analyse du document\n";
    exit;
}

$root = $dom->document_element();
?>
```

Voir aussi [domxml_open_mem](#) et [domxml_new_doc](#).

7.24.77 domxml_open_mem() : Crée un objet DOM pour un document XML

object **domxml_open_mem** (string str)

domxml_open_mem analyse le document XML contenu dans la chaîne `str` et retourne un objet de classe "Dom document", avec les propriétés listées ci-dessus. La fonction domxml_open_file ou domxml_new_doc doit être appelée avant tout autre appel de fonction.

Ouvrir un document XML à partir d'une chaîne

```
<?php
include("exemple.inc");

if(!$dom = domxml_open_mem($xmlstr)) {
    echo "Erreur lors de l'analyse du document\n";
    exit;
}

$root = $dom->document_element();
?>
```

Voir aussi domxml_open_file et domxml_new_doc .

7.24.78 domxml_version() : Lit le numéro de version de la librairie XML

string **domxml_version** (void)

domxml_version lit le numéro de version de la librairie XML actuellement utilisée.

7.24.79 domxml_xmlltree() : Crée un arbre d'objets PHP à partir d'un document XML

object **domxml_xmlltree** (string str)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

domxml_xmlltree 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 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 en chaîne. Cependant, cette fonction est pratique si vous vouez lire un fichier XML et en étudier le contenu.

7.24.80 domxml_xslt_stylesheet_doc() : Crée un objet domssltstylesheet à partir d'un objet domdocument

object **domxml_xslt_stylesheet_doc** (object docdocument 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.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Voir aussi [domxsltstylesheet->process](#) , [domxml_xslt_stylesheet](#) et [domxml_xslt_stylesheet_file](#) .

7.24.81 domxml_xslt_stylesheet_file() : Crée un objet domxsltstylesheet à partir d'un document xsl dans un fichier

object **domxml_xslt_stylesheet_file** (string xsl file)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

Voir aussi [domxsltstylesheet->process](#) , [domxml_xslt_stylesheet](#) et [domxml_xslt_stylesheet_doc](#) .

7.24.82 domxml_xslt_stylesheet() : Crée un objet domxsltstylesheet à partir d'un document xml

object **domxml_xslt_stylesheet** (string xsl document)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

Voir aussi [domxsltstylesheet->process](#) , [domxml_xslt_stylesheet_file](#) et [domxml_xslt_stylesheet_doc](#) .

7.24.83 xpath_eval_expression() : Calcule un chemin XPath à partir d'une chaîne

array **xpath_eval_expression** (object xpath_context)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 [xpath_eval](#) .

7.24.84 xpath_eval() : Calcule un chemin XPath à partir d'une chaîne

```
array xpath_eval ( object xpath context , string xpath expression , object contextnode )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 paramètre optionnel `contextnode` sert à spécifier des requêtes XPath relatives.

Voir aussi [xpath_new_context](#) .

7.24.85 xpath_new_context() : Crée un nouveau contexte xpath

```
object xpath_new_context ( object dom document )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 [xpath_eval](#) .

7.24.86 xptr_eval() : Calcul un chemin XPtr à partir d'une chaîne

```
int xptr_eval ( object xpath_context , string eval_str )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.24.87 xptr_new_context() : Crée un nouveau contexte XPath

```
string xptr_new_context ( object doc_handle )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.25 .NET

7.25.1 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](#)

7.26 Gestion des erreurs

7.26.1 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'au erreurs personnalisées retournées par les fonctions.

7.26.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.26.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.26.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>error_reporting</code>	<code>E_ALL & ~E_NOTICE</code>	PHP_INI_ALL
<code>display_errors</code>	<code>"1"</code>	PHP_INI_ALL
<code>display_startup_errors</code>	<code>"0"</code>	PHP_INI_ALL
<code>log_errors</code>	<code>"0"</code>	PHP_INI_ALL
<code>log_errors_max_len</code>	<code>"1024"</code>	PHP_INI_ALL
<code>ignore_repeated_errors</code>	<code>"0"</code>	PHP_INI_ALL
<code>ignore_repeated_source</code>	<code>"0"</code>	PHP_INI_ALL
<code>report_memleaks</code>	<code>"1"</code>	PHP_INI_SYSTEM
<code>track_errors</code>	<code>"0"</code>	PHP_INI_ALL
<code>html_errors</code>	<code>"1"</code>	PHP_INI_ALL
<code>docref_root</code>	<code>"http://www.php.net/"</code>	PHP_INI_ALL
<code>docref_ext</code>	<code>".html"</code>	PHP_INI_ALL
<code>error_prepend_string</code>	<code>NULL</code>	PHP_INI_ALL
<code>error_append_string</code>	<code>NULL</code>	PHP_INI_ALL
<code>error_log</code>	<code>NULL</code>	PHP_INI_ALL
<code>warn_plus_overloading</code>	<code>NULL</code>	PHP_INI??

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`error_reporting` integer

Fixe le niveau d'erreur. Ce paramètre est un entier, représentant un champs de bits. Ajoutez les valeurs suivantes pour choisir le niveau que vous désirez, telles que décrite dans la section [Constantes prédéfinies](#), 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, la configuration par défaut montre toutes les erreurs, sauf les `E_NOTICE` . Il est recommandée de les afficher durant le développement.

Note Activer le rapport d'erreur de niveau <code>E_NOTICE</code> durant le développement a des avantages. En terme de débogage, les message d'alertes vous signalent des bugs potentiels dans votre code. Par exemple, l'utilisation de valeurs non initialisées est signalé. Il est aussi plus pratique pour trouver des coquilles, et ainsi, gagner du temps. Les messages NOTICE vous signaleront aussi les mauvaises pratiques de codages. Par exemple <code>\$arr[item]</code> doit toujours être écrit <code>\$arr['item']</code> car PHP va considérer "item" comme une constante, au premier abord. Si cette constante n'est pas définie, alors il va l'utilise comme une chaîne.

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 au format HTML ou non.

`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, et c'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` integer

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.

`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 par les erreurs les erreurs répétées provenant de fichier 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 cliquable, 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 le manuel dans votre langue, et configurer cette option pour qu'elle pointe sur votre manuel. 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%2Fwww.php.net%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.

`docref_ext` string

Voir aussi `docref_root`.

Note
La valeur de <code>docref_ext</code> doit commencer avec un point <code>.</code> .

`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. 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 (`.`).

7.26.5 Constantes prédefinies

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 <code>PHP</code> , 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 (integer)</code>	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 (integer)</code>	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 (integer)</code>	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 (integer)</code>	Les notes ne sont pas affichées par défaut, et indique 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 (integer)</code>	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.	PHP 4 seulement
32	<code>E_CORE_WARNING (integer)</code>	Elles sont similaires à <code>E_WARNING</code> , mais elles sont générées par le code de PHP. Les fonctions ne doivent pas générer ce genre d'erreur.	PHP 4 seulement

64	E_COMPILE_ERROR (integer)	Elles sont similaires à E_ERROR, mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur.	PHP 4 seulement
128	E_COMPILE_WARNING (integer)	Elles sont similaires à E_WARNING, mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur.	PHP 4 seulement
256	E_USER_ERROR (integer)	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.	PHP 4 seulement
512	E_USER_WARNING (integer)	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.	PHP 4 seulement
1024	E_USER_NOTICE (integer)	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.	PHP 4 seulement
2047	E_ALL (integer)	Toutes les erreurs et alertes supportées.	

Les valeurs ci-dessus (numérique ou symbolique) sont utilisées pour constituer des champs de bits, qui spécifie 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`.

7.26.6 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

```
<?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 2,8,256,512 et 1024
    $errortype = array (
        1 => "Erreur",
        2 => "Alerte",
        4 => "Error d'analyse",
        8 => "Note",
        16 => "Core Error",
        32 => "Core Warning",
        64 => "Compile Error",
        128 => "Compile Warning",
        256 => "Erreur spécifique",
        512 => "Alerte spécifique",
        1024=> "Note spécifique"
    );

    // 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";
    $err .= "\t<errmsg>".$errmsg."</errmsg>\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";

    // test
    // echo $err;

    // 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";

?>

```

7.26.7 Voir aussi

Voir aussi [syslog](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [Voir aussi](#)
- [debug_backtrace](#)
- [debug_print_backtrace](#)
- [error_log](#)
- [error_reporting](#)
- [restore_error_handler](#)
- [set_error_handler](#)
- [trigger_error](#)
- [user_error](#)

7.26.9 debug_print_backtrace() : Affiche une backtrace

`void debug_print_backtrace ( void )`

`debug_print_backtrace` affiche la backtrace de ^{PHP}.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Voir aussi `debug_backtrace`.

7.26.10 error_log() : Stocke un message d'erreur

```
int error_log ( string message , int message_type , string destination , string extra_headers )
```

`error_log` envoie un message d'erreur à l'historique du serveur web, à un port ^{TCP} ou un fichier. ^{message} est le message d'erreur qui doit être enregistré. ^{message_type} indique où le message doit être envoyé :

0	message est envoyé à l'historique PHP, qui est basé sur l'historique système ou un fichier, en fonction de la configuration de <code>error_log</code> .
1	message est envoyé par email à l'adresse destination . C'est le seul type qui utilise le quatrième paramètre <code>extra_headers</code> . Ce message utilise la même fonction interne que <code>mail</code> .
2	message est envoyé par la connexion de débogage PHP. Cette option n'est disponible que si l'option <code>remote debugging</code> a été désactivée. Dans ce cas, le paramètre destination spécifie l'hôte ou l'adresse IP, et optionnellement le numéro de port, de la socket qui recevra les informations de débogage.
3	message est ajouté au fichier destination .

Attention

Le débogage à distance via TCP/IP est une fonctionnalité PHP 3 qui **n'est pas** disponible en PHP 4.

Exemples avec `error_log`

```
<?php
// Envoi une notification par l'historique du serveur, si la connexion à la base
// de données est impossible.
if (!Ora_Logon ($username, $password)) {
    error_log ("Base Oracle indisponible!", 0);
}
// Indiquer à l'administrateur, par email, qu'il n'y a plus de F00
if (!($foo = allocate_new_foo())) {
    error_log ("Aya! Il ne reste plus de F00 disponibles!", 1,
        "operateur@mondomaine.com");
}
// D'autres manières d'appeler error_log():
error_log ("Grosse bourde!", 2, "127.0.0.1:7000");
error_log ("Grosse bourde!", 2, "loghost");
error_log ("Grosse bourde!", 3, "/var/tmp/my-errors.log");
?>
```

7.26.11 error_reporting() : Fixe le niveau de rapport d'erreurs PHP

```
int error_reporting ( int level )
```

`error_reporting` fixe le niveau de rapport d'erreur PHP et retourne l'ancienne valeur. Le niveau d'erreur peut être un champ de bits, ou une constante. L'utilisation des constantes est vivement recommandée, pour assurer une compatibilité maximale avec les futures versions. Au fur et à mesure que de nouveaux niveaux d'erreurs sont créés, l'intervalle de validité des niveaux évolue, et les anciennes valeurs n'ont plus les mêmes significations.

Exemple de modification de niveau d'erreur

```
<?php
error_reporting (55);
// En PHP 3, équivalent à E_ALL ^ E_NOTICE
// ...en PHP 4, '55' signifie (E_ERROR | E_WARNING | E_PARSE | E_CORE_ERROR | E_CORE_WARNING)
error_reporting (2039);
// PHP 4 équivalent à E_ALL ^ E_NOTICE
```

```
error_reporting (E_ALL ^ E_NOTICE);
// La même signification en PHP 3 et 4
```

Suivez les liens de chaque valeur interne pour connaître leur signification :

constante	valeur
1	E_ERROR
2	E_WARNING
4	E_PARSE
8	E_NOTICE
16	E_CORE_ERROR
32	E_CORE_WARNING
64	E_COMPILE_ERROR
128	E_COMPILE_WARNING
256	E_USER_ERROR
512	E_USER_WARNING
1024	E_USER_NOTICE
2047	E_ALL

Voir aussi la directive [display_errors](#) et [ini_set](#).

7.26.12 `restore_error_handler()` : Réactive l'ancienne fonction de gestion des erreurs

```
void 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).

Voir aussi [error_reporting](#), [set_error_handler](#), [trigger_error](#) et [user_error](#).

7.26.13 `set_error_handler()` : Choisi une fonction utilisateur comme gestionnaire d'erreurs

```
string set_error_handler ( callback error_handler )
```

[set_error_handler](#) choisit la fonction utilisateur `error_handler` pour gérer les erreurs dans un script. Retourne un pointeur sur l'ancienne fonction de gestion des erreurs (si il y en avait une), ou `FALSE`, en cas d'erreur. [set_error_handler](#) sert à définir votre propre gestionnaire d'erreurs, qui prendra en charge leur traitement durant l'exécution d'un script. Cela peut être utile lorsque vous devez repérer des erreurs critiques lors d'un nettoyage de bases, ou bien si vous souhaitez générer une erreur dans certaines conditions (avec [trigger_error](#)).

La fonction utilisateur doit accepter deux arguments : le code de l'erreur, et une chaîne décrivant l'erreur. L'exemple ci dessous montre le traitement d'exceptions en déclenchant des erreurs, et en les gérant avec une fonction utilisateur.

Note

Au lieu d'un nom de fonction, vous pouvez utiliser un tableau contenant deux éléments, une référence sur un objet et un nom de méthode. Cette fonctionnalité a été ajoutée en PHP 4.3.0.

Note

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` et `E_COMPILE_WARNING`.

L'exemple ci-dessous illustre l'interception d'erreurs internes avec génération d'erreur, et exploitation dans une fonction utilisateur :

Traitement des erreurs avec `set_error_handler` et `trigger_error`

```
<?php

// redéfini les constantes utilisateur (PHP 4 Uniquement)
define ("FATAL",E_USER_ERROR);
define ("ERROR",E_USER_WARNING);
define ("WARNING",E_USER_NOTICE);

// configure le niveau de rapport d'erreur pour ce script
error_reporting (FATAL | ERROR | WARNING);

// gestionnaire d'erreurs
function myErrorHandler ($errno, $errstr, $errfile, $errline) {
    switch ($errno) {
        case FATAL:
            echo "<b>FATAL</b> [$errno] $errstr<br>\n";
            echo " Fatal error in line ".$errline." of file ".$errfile;
            echo ", PHP ".PHP_VERSION." (" .PHP_OS.")<br>\n";
            echo "Aborting...<br>\n";
            exit(1);
            break;
        case ERROR:
            echo "<b>ERROR</b> [$errno] $errstr<br>\n";
            break;
        case WARNING:
            echo "<b>WARNING</b> [$errno] $errstr<br>\n";
            break;
        default:
            echo "Unkown error type: [$errno] $errstr<br>\n";
            break;
    }
}

// fonction pour tester la gestion d'erreur
function scale_by_log ($vect, $scale) {
    if ( !is_numeric($scale) || $scale <= 0 )
        trigger_error("log(x) for x <= 0 is undefined, you used: scale = $scale",
            FATAL);
    if (!is_array($vect)) {
        trigger_error("Incorrect input vector, array of values expected", ERROR);
        return null;
    }
    for ($i=0; $i<count($vect); $i++) {
        if (!is_numeric($vect[$i]))
            trigger_error("Value at position $i is not a number, using 0 (zero)",
                WARNING);
        $stemp[$i] = log($scale) * $vect[$i];
    }
    return $stemp;
}

// Configuration du gestionnaire d'erreurs
$old_error_handler = set_error_handler("myErrorHandler");

// génération de quelques erreurs. Commençons par créer un tableau
echo "vecteur a \n";
$a = array(2,3,"foo",5.5,43.3,21.11);
print_r($a);

// générons maintenant un autre tableau, avec des alertes
echo "----\nvecteur b - une alerte (b = log(PI) * a)\n";
$b = scale_by_log($a, M_PI);
print_r($b);

// ceci est un problème, nous avons utilisé une chaîne au lieu d'un tableau
echo "----\nvecteur c - une erreur\n";
$c = scale_by_log("not array",2.3);
var_dump($c);

// Ceci est une erreur critique : le logarithme de zéro ou d'un nombre négatif est indéfini
echo "----\nvecteur d - erreur fatale\n";
$d = scale_by_log($a, -2.5);

?>
```

L'exécution du script devrait donner ceci :

Résultat de `set_error_handler` et `trigger_error`

```
vecteur a
Array
(
```

```

    [0] => 2
    [1] => 3
    [2] => foo
    [3] => 5.5
    [4] => 43.3
    [5] => 21.11
)
-----
vecteur b - une alerte (b = log(PI) * a)
<b>WARNING</b> [1024] Value at position 2 is not a number, using 0 (zero)<br>
Array
(
    [0] => 2.2894597716988
    [1] => 3.4341896575482
    [2] => 0
    [3] => 6.2960143721717
    [4] => 49.566804057279
    [5] => 24.165247890281
)
-----
vecteur c - une erreur
<b>ERROR</b> [512] Incorrect input vector, array of values expected<br>
NULL
-----
vecteur d - erreur fatale
<b>FATAL</b> [256] log(x) for x <= 0 is undefined, you used: scale = -2.5<br>
Fatal error in line 37 of file /Users/imac/~/, PHP 5.0.0-dev (Darwin)<br>
Aborting...<br>

```

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 [@](#) (0 sera retourné).

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.

Note

Si une erreur survient avant que le script ne soit exécuté (par exemple un téléchargement de fichier), le gestionnaire d'erreurs personnalisés ne pourra pas être appelé, car il n'est pas encore enregistré.

Voir aussi [error_reporting](#) , [restore_error_handler](#) , [trigger_error](#) et [user_error](#)

7.26.14 trigger_error() : Déclenche une erreur utilisateur

```
void 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'erreur interne, ou un gestionnaire d'erreurs utilisateur qui a été choisi comme gestionnaire d'erreur avec [set_error_handler](#) .

[trigger_error](#) est pratique lorsque vous devez générer une réponse particulière lors de l'exécution. Par exemple :

Exemple avec trigger_error

```

<?php
if (assert ($divisor == 0))
    trigger_error ("Impossible de diviser par zéro", E_USER_ERROR);
?>

```

Note

Voyez [set_error_handler](#) pour une illustration.

Note

`error_msg` est limité à 1024 caractères en longueur. Tous les caractères supplémentaires seront ignorés.

Voir aussi [error_reporting](#) , [set_error_handler](#) , [restore_error_handler](#) et [user_error](#) .

7.26.15 `user_error()` : Alias de [trigger_error](#)

[user_error](#) est un alias de la fonction [trigger_error](#) .

7.27 File alteration monitor functions

7.27.1 Introduction

FAM monitors files and directories, notifying interested applications of changes.

A PHP script may specify a list of files for FAM to monitor using the functions provided by this extension.

The FAM process is started when the first connection from any application to it is opened. It exits after all connections to it have been closed.

7.27.2 Pré-requis

This extension requires ... version ... as available on ...

7.27.3 Installation

7.27.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.27.5 Types de ressources

7.27.5.1 FAM resource

7.27.6 Constantes prédefinies

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	meaning
FAMChanged (integer)	The status of
FAMDeleted (integer)	
FAMStartExecuting (integer)	
FAMStopExecuting (integer)	
FAMCreated (integer)	
FAMMoved (integer)	
FAMAcknowledge (integer)	
FAMExists (integer)	
FAMEndExist (integer)	

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)

- [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](#)

7.27.8 [fam_close\(\)](#) : Close FAM connection

void **fam_close** (resource fam)

[fam_close](#) closes a connection to the FAM service previously opened using [fam_open](#).

7.27.9 [fam_monitor_collection\(\)](#) : Monitor a collection of files in a directory for changes

resource **fam_monitor_collection** (resource fam , string dirname , int depth , string mask)

[fam_monitor_file](#) requests monitoring for a collection of files within a directory. The actual files to be monitored are specified by a directory path in ^{dirname}, the maximum search ^{depth} starting from this directory and a shell pattern restricting the file names to look for.

See also [fam_monitor_file](#), [fam_monitor_directory](#), [fam_cancel_monitor](#), [fam_suspend_monitor](#), and [fam_resume_monitor](#).

7.27.10 [fam_monitor_directory\(\)](#) : Monitor a directory for changes

resource **fam_monitor_directory** (resource fam , string dirname)

[fam_monitor_file](#) requests monitoring for a directory and all contained files. A FAM event will be generated whenever the status of the directory (i.e. the result of function [stat](#) on that directory) or its content (i.e. the results of [readdir](#)) change.

See also [fam_monitor_file](#), [fam_monitor_collection](#), [fam_cancel_monitor](#), [fam_suspend_monitor](#), and [fam_resume_monitor](#).

7.27.11 [fam_monitor_file\(\)](#) : Monitor a regular file for changes

resource **fam_monitor_file** (resource fam , string filename)

[fam_monitor_file](#) requests monitoring for a single file. A FAM event will be generated whenever the file status (i.e. the result of function [stat](#) on that file) changes.

See also [fam_monitor_directory](#), [fam_monitor_collection](#), [fam_cancel_monitor](#), [fam_suspend_monitor](#), and [fam_resume_monitor](#).

7.27.12 fam_next_event() : ...

array **fam_next_event** (resource fam)

fam_next_event returns the next pending FAM event. The function will block until an event is available which can be checked for using fam_pending .

fam_ext_event will return an array that contains a FAM event code in element 'code', the path of the file this event applies to in element 'filename' and optionally a hostname in element 'hostname' .

The possible event codes are described in detail in the introduction part of this section.

See also fam_pending .

7.27.13 fam_open() : Open connection to FAM daemon

resource **fam_open** (string appname)

fam_open opens a connection to the FAM service daemon. The optional parameter `appname` should be set to a string identifying the application for logging reasons.

See also fam_close .

7.27.14 fam_pending() : Check for pending FAM events

bool **fam_pending** (resource fam)

fam_pending returns `TRUE` if events are available to be fetched using fam_next_event .

See also fam_next_event .

7.27.15 fam_resume_monitor() : Resume suspended monitoring

bool **fam_resume_monitor** (resource fam , resource fam_monitor)

fam_resume_monitor resumes monitoring of a resource previously suspend using fam_suspend_monitor .

See also fam_suspend_monitor .

7.27.16 fam_suspend_monitor() : Temporarily suspend monitoring

bool **fam_suspend_monitor** (resource fam , resource fam_monitor)

fam_suspend_monitor temporarily suspend monitoring of a resource previously requested using one of the fam_monitor ... functions. Monitoring can later be continued using fam_resume_monitor without the need of requesting a complete new monitor.

See also fam_resume_monitor , and fam_cancel_monitor .

7.28 FrontBase

7.28.1 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=Documentat> .

Frontbase est supporté depuis ^{PHP} 4.0.6.

7.28.2 Pré-requis

Vous devez installer un serveur de bases de données FrontBase ou, tout au moins, les librairies clientes fbsql, pour accéder à ces fonctions. Vous pouvez accéder au site de FrontBase à l'adresse <http://www.frontbase.com/> .

7.28.3 Installation

Pour pouvoir les utiliser, vous devez compiler PHP avec le support fbsql en utilisant l'option `--with-fbsql` . Si vous utilisez cette option sans spécifier le chemin jusqu'à l'installation fbsql, PHP recherchera les librairies 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 librairies de FrontBase, et éviter les conflits.

7.28.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<code>fbsql.allow_persistent</code>	"1"	PHP_INI_SYSTEM
<code>fbsql.generate_warnings</code>	"0"	PHP_INI_SYSTEM
<code>fbsql.autocommit</code>	"1"	PHP_INI_SYSTEM
<code>fbsql.max_persistent</code>	"-1"	PHP_INI_SYSTEM
<code>fbsql.max_links</code>	"128"	PHP_INI_SYSTEM
<code>fbsql.max_connections</code>	"128"	PHP_INI_SYSTEM
<code>fbsql.max_results</code>	"128"	PHP_INI_SYSTEM
<code>fbsql.batchSize</code>	"1000"	PHP_INI_SYSTEM
<code>fbsql.default_host</code>	NULL	PHP_INI_SYSTEM
<code>fbsql.default_user</code>	"_SYSTEM"	PHP_INI_SYSTEM
<code>fbsql.default_password</code>	" "	PHP_INI_SYSTEM
<code>fbsql.default_database</code>	" "	PHP_INI_SYSTEM
<code>fbsql.default_database_password</code>	" "	PHP_INI_SYSTEM

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#) .

7.28.5 Types de ressources

7.28.6 Constantes prédefinies

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 prédefinies](#)
- [fbsql_affected_rows](#)
- [fbsql_autocommit](#)
- [fbsql_change_user](#)
- [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_transaction](#)
- [fbsql_start_db](#)
- [fbsql_stop_db](#)
- [fbsql_tablename](#)
- [fbsql_username](#)
- [fbsql_warnings](#)

7.28.8 [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](#)

7.28.9 [fbsql_change_user\(\)](#) : Change le nom d'utilisateur de la session active

resource **fbsql_change_user** (string user , string password , string database , resource link_identifiant)

fbsql_change_user change le nom de l'utilisateur courant sur la session active courante, ou sur `link_identifiant` . 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'authentification échoue, la session courante restera ouverte.

7.28.10 fbsql_close() : Ferme la connexion FrontBase

bool **fbsql_close** (resource link_identifiant)

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_identifiant` . Si `link_identifiant` 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 <u>fbsql_close</u> <pre><?php \$link = fbsql_connect("localhost", "_SYSTEM", "secret") or die("Impossible de se connecter"); print("Connecté!"); fbsql_close(\$link); ?></pre>

Voir aussi fbsql_connect et fbsql_pconnect .

7.28.11 fbsql_commit() : Valide une transaction

bool **fbsql_commit** (resource link_identifiant)

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

7.28.12 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
$link = fbsql_connect("localhost", "_SYSTEM", "secret")
    or die("Impossible de se connecter");
print("Connexion effectuée");
fbsql_close($link);
?>
```

Voir aussi [fbsql_pconnect](#) et [fbsql_close](#).

7.28.13 fbsql_create_blob() : Crée un BLOB

string **fbsql_create_blob** (string blob_data , resource link_identifieur)

[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";
$fhp = fopen($filename, "rb");
$blobdata = fread($fhp, filesize($filename));
fclose($fhp);
$blobHandle = fbsql_create_blob($blobdata, $link);
$sql = "INSERT INTO BLOB_TABLE (BLOB_COLUMN) VALUES ($blobHandle)";
$rs = fbsql_query($sql, $link);
?>
```

Voir aussi [fbsql_create_clob](#), [fbsql_read_blob](#), [fbsql_read_clob](#) et [fbsql_set_lob_mode](#).

7.28.14 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";
$fhp = fopen($filename, "rb");
$clobdata = fread($fhp, filesize($filename));
fclose($fhp);
```

```

$clobHandle = fbsql_create_clob($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](#) .

7.28.15 fbsql_create_db() : Crée une base de données

```
bool fbsql_create_db ( string database_name , resource link_identifieur )
```

[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")) {
    print("Base de données créée!\n");
} else {
    printf("Erreur de création de la base de données : %s\n", fbsql_error());
}
?>

```

Voir aussi [fbsql_drop_db](#) .

7.28.16 fbsql_data_seek() : Déplace le pointeur interne de résultat FBSQL

```
bool fbsql_data_seek ( resource result_identifieur , int row_number )
```

[fbsql_data_seek](#) déplace le pointeur interne de ligne dans le résultat de requête `result_identifieur` 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;
    printf ("%s %s<BR>\n", $row->last_name, $row->first_name);
}
fbsql_free_result($result);
?>

```

7.28.17 fbsql_database_password() : Modifie/lit le mot de passe dans une base FBSQL

string **fbsql_database_password** (resource link_identifiant , string database_password)

fbsql_database_password retourne le mot de passe du serveur représenté par link_identifiant .

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_identifiant 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 .

7.28.18 fbsql_database() : Lit ou définit le nom de la base de données utilisée avec une connexion

string **fbsql_database** (resource link_identifiant , string database)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.28.19 fbsql_db_query() : Envoie une requête à la base FrontBase

resource **fbsql_db_query** (string database , string query , resource link_identifiant)

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_identifiant 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 arguments.

Voir aussi fbsql_connect .

7.28.20 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.

7.28.21 fbsql_drop_db() : Supprime une base de données FrontBase

```
bool fbsql_drop_db ( string database_name , resource link_identifieur )
```

fbsql_drop_db essaie de supprimer la base de données `database_name`, sur la connexion représentée par `link_identifieur`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.28.22 fbsql_errno() : Retourne le code d'erreur FrontBase

```
int fbsql_errno ( resource link_identifieur )
```

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](#)

7.28.23 fbsql_error() : Retourne le message d'erreur FrontBase

string **fbsql_error** (resource link_identifiant)

[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](#)

7.28.24 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 lignes.

[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 clefs 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);
?>
```

7.28.25 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);
?>
```

7.28.26 fbsql_fetch_field() : Lit des informations sur une colonne dans un résultat, et retourne un objet

object **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](#) .

7.28.27 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](#) .

7.28.28 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](#) .

7.28.29 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 courant dans le résultat `result` , ou bien `FALSE` s'il n'y a plus de lignes 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 lignes à lire.

Voir aussi [fbsql_fetch_array](#) , [fbsql_fetch_object](#) , [fbsql_data_seek](#) , [fbsql_fetch_lengths](#) et [fbsql_result](#) .

7.28.30 fbsql_field_flags() : Lit les options associé 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 façons à faciliter la manipulation avec [explode](#) .

7.28.31 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` .

7.28.32 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
```

7.28.33 `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`.

7.28.34 `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 issue le champ d'indice `field_offset`. Les numéros de colonnes commencent à 0.

7.28.35 `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();
?>

```

7.28.36 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 toutes manières libérées à la fin du script.

7.28.37 fbsql_get_autostart_info() : Pas de documentation

array **fbsql_get_autostart_info** (resource link_identifiant)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.28.38 fbsql_hostname() : Lit ou définit le nom de l'hôte à utiliser avec une connexion

string **fbsql_hostname** (resource link_identifiant , string host_name)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.28.39 fbsql_insert_id() : Lit le dernier identifiant généré par une requête INSERT

int **fbsql_insert_id** (resource link_identifiant)

fbsql_insert_id retourne l'identifiant généré par la colonne de type DEFAULT UNIQUE, lors de la dernière requête INSERT, avec la connexion `link_identifiant`. Si `link_identifiant` 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 sauvegarder 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.

7.28.40 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 ceci :

```
database1
database2
database3
...
```

Note

L'exemple ci-dessus peut aussi bien fonctionner avec la fonction fbsql_fetch_row ou toute autre similaire.

7.28.41 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);
$num_columns = fbsql_num_fields($fields);
for ($i = 0; $i < $num_columns; $i++) {
    echo fbsql_field_name($fields, $i) . "\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
field1
field2
field3
...
```

7.28.42 fbsql_list_tables() : Liste les tables dans une base de données FrontBase

resource **fbsql_list_tables** (string database , resource link_identifieur)

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.

7.28.43 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 multiple 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 résultat 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);
?>
```

7.28.44 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 .

7.28.45 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](#).

7.28.46 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.

7.28.47 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 le même 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 fbssl 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 dite 'persistante'.

7.28.48 fbsql_query() : Exécute une requête sur un serveur FrontBase

```
resource fbsql_query ( string query , resource link_identifier )
```

[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.

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 utilisée 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 bien `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`. Lors 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`.

7.28.49 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 lectures 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 indentifiant vers les données du
// blob de la première ligne
$clob_data = fbsql_read_blob($row_data[0]);
fbsql_free_result($rs);
?>

```

Voir aussi [fbsql_create_blob](#) , [fbsql_read_blob](#) , [fbsql_read_clob](#) et [fbsql_set_lob_mode](#) .

7.28.50 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 lectures 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 indentifiant 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](#) , [fbsql_read_clob](#) et [fbsql_set_lob_mode](#) .

7.28.51 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 doivent pas être mélangé avec d'autres fonctions qui utilisent aussi le résultat `result` .

Alternative vivement recommandées : [fbsql_fetch_row](#) , [fbsql_fetch_array](#) et [fbsql_fetch_object](#) .

Voir aussi [fbsql_create_blob](#) , [fbsql_create_clob](#) , [fbsql_read_blob](#) et [fbsql_read_clob](#) .

7.28.55 fbsql_set_transaction() : Configure le verrouillage et l'isolation de transactions

```
void fbsql_set_transaction ( resource link_identifiant , int locking , int isolation )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.28.56 fbsql_start_db() : Démarre une base de données FBSQL

```
bool fbsql_start_db ( string database_name , resource link_identifiant )
```

[fbsql_start_db](#) démarre la base de données `database_name` sur la connexion `link_identifiant` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [fbsql_db_status](#) et [fbsql_stop_db](#) .

7.28.57 fbsql_stop_db() : Stoppe une base de données FBSQL

```
bool fbsql_stop_db ( string database_name , resource link_identifiant )
```

[fbsql_stop_db](#) stoppe la base de données `database_name` sur la connexion `link_identifiant` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [fbsql_db_status](#) et [fbsql_start_db](#) .

7.28.58 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:3306");
$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++;
}
?>
```

7.28.59 fbsql_username() : Lit ou définit le nom de login à utiliser avec une connexion

string **fbsql_username** (resource link_identifiant , string username)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.28.60 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} .

7.29 FilePro

7.29.3 filepro_fieldcount() : Retourne le nombre de champs dans une base filePro

```
int filepro_fieldcount ( void )
```

filepro_fieldcount retourne le nombre de champs (ou colonnes) d'une base filePro.

Voir aussi filepro .

- [Introduction](#)
- [Installation](#)
- [filepro_fieldcount](#)
- [filepro_fieldname](#)
- [filepro_fielddtype](#)
- [filepro_fieldwidth](#)
- [filepro_retrieve](#)
- [filepro_rowcount](#)
- [filepro](#)

7.29.5 filepro_fielddtype() : Retourne le type d'un champ Filepro

```
string filepro_fielddtype ( int field_number )
```

filepro_fielddtype retourne le type du champ d'index `field_number` .

7.29.6 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` .

7.29.7 filepro_retrieve() : Retourne la valeur d'un champ Filepro

```
string filepro_retrieve ( int row_number , int field_number )
```

filepro_retrieve retourne la valeur du champ d'index `row_number` , et à la ligne `field_number` .

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.

7.29.8 filepro_rowcount() : Retourne le nombre de champs dans une base filePro.

```
int filepro_rowcount ( void )
```

`filepro_rowcount` retourne le nombre de lignes dans une base filePro.

Voir aussi `filepro` .

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.

7.29.9 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 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.

7.30 Système de fichiers

7.30.1 Introduction

7.30.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.30.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.30.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>allow_url_fopen</code>	<code>"1"</code>	PHP_INI_ALL
<code>user_agent</code>	NULL	PHP_INI_ALL
<code>default_socket_timeout</code>	<code>"60"</code>	PHP_INI_ALL
<code>from</code>	NULL	??
<code>auto_detect_line_endings</code>	<code>"Off"</code>	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

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 apparaissent comme [zlib](#), qui ajoutent d'autres fonctionnalités.

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 <code>--disable-url-fopen-wrapper</code> .
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 <code>imagecreatefromXXX</code> 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="joe@example.com"` 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 d'interopérer avec des systèmes Macintosh, mais par défaut, cette option est désactivée, car cette détection impose une 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.

7.30.5 Types de ressources

7.30.6 Constantes prédefinies

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 ( integer )
GLOB_ONLYDIR ( integer )
GLOB_MARK ( integer )
GLOB_NOSORT ( integer )
GLOB_NOCHECK ( integer )
GLOB_NOESCAPE ( integer )
PATHINFO_DIRNAME ( integer )
PATHINFO_BASENAME ( integer )
PATHINFO_EXTENSION ( integer )

```

7.30.7 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 prédefinies](#)
- [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](#)
- [fpasssthru](#)
- [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](#)

7.30.9 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.

Voir aussi [chown](#) et [chmod](#).

7.30.10 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+rw,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, écriture pour les autres
chmod ("/un_dossier/un_fichier", 0644);
```



```
// Tous pour le propriétaire, lecture et écriture 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.

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.

Voir aussi [chown](#) et [chgrp](#).

7.30.11 chown() : Change le groupe propriétaire du fichier

```
bool chown ( string filename , mixed user )
```

[chown](#) change le groupe propriétaire du fichier. 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 fichiers distants, car le fichier utilisé doit être accessible sur le système de fichiers local.

Voir aussi [chmod](#).

7.30.12 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.

La valeur du cache n'est valable que pour la durée d'une requête.

Les fonctions affectées sont : [stat](#), [lstat](#), [file_exists](#), [is_writable](#), [is_readable](#), [is_executable](#), [is_file](#), [is_dir](#), [is_link](#), [filectime](#), [fileatime](#), [filemtime](#), [fileinode](#), [filegroup](#), [fileowner](#), [filesize](#), [filetype](#), et [fileperms](#).

7.30.13 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 `copy`

```
<?php
if (!copy($file, $file.'.bak')) {
    print("La copie du fichier $file n'a pas réussi...<BR>\n");
}
?>
```

Note

Depuis ^{PHP} 4.3.0, les deux paramètres `source` et `dest` peuvent être des URL si les gestionnaires d'URL ont été activés ("fopen wrappers"). 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.

Voir aussi `move_uploaded_file` et `rename`.

7.30.14 delete() : Effacer

void **delete** (string file)

Ceci est une fausse entrée du manuel pour ceux qui recherchent en fait la fonction `unlink` ou `unset`.

Voir aussi `unlink` et `unset`.

7.30.15 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 si 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.

Voir aussi `basename`, `pathinfo` et `realpath`.

7.30.16 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.

7.30.17 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 libres
// dans le dossier "/"
?>
```

Note

Cette fonction ne fonctionne pas avec les fichiers distants, car le fichier utilisé doit être accessible sur le système de fichiers local.

7.30.18 diskfreespace() : Alias de disk_free_space

float **diskfreespace** (string directory)

diskfreespace est l'ancien nom de la fonction disk_free_space. Elle est obsolète, et il est vivement recommandé d'utiliser disk_free_space.

7.30.19 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);
?>
```

7.30.20 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`.

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par `fopen`, `popen`, ou `fsockopen`.

7.30.21 fflush() : Envoi 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`.

7.30.22 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 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](#) .

7.30.23 fgetcsv() : Renvoie la ligne courante et cherche les champs CSV

array **fgetcsv** (resource handle , int length , string delimiter , string enclosure)

[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).

[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;
$fp = fopen ("test.csv","r");
while ($data = fgetcsv ($fp, 1000, ",")) {
    $num = count ($data);
    print "<p> $num champs dans la ligne $row: <br>\n";
    $row++;
    for ($c=0; $c < $num; $c++) {
        print $data[$c] . "<br>\n";
    }
}
fclose ($fp);
?>
```

Voir aussi [explode](#) , [file](#) et [pack](#) .

7.30.24 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, ou bien la fin du fichier, ou encore un retour chariot (le premier des trois qui sera rencontré).

Si une erreur survient, [fgets](#) retourne ^{FALSE} .

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.

`handle` doit être valide, et avoir été correctement ouvert par `fopen`, `popen`, ou `fsockopen`.

Un exemple simple :

Lecture d'un fichier ligne par ligne

```
<?php
$fd = fopen ("/tmp/inputfile.txt", "r");
while (!feof ($fd)) {
    $buffer = fgets($fd, 4096);
    echo $buffer;
}
fclose ($fd);
?>
```

Note

Le paramètre `length` est devenu optionnel depuis ^{PHP} 4.2.0. Si il est omis, il prend la valeur par défaut de 1024. Depuis ^{PHP} 4.3.0, omettre le paramètre `length` fait que la fonction va continuer à lire le flôt 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

`fgets` 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 `auto_detect_line_endings`.

Voir aussi `fread`, `fgetc`, `stream_get_line`, `fopen`, `popen`, `fsockopen` et `socket_set_timeout`.

7.30.25 fgetss() : Renvoie la ligne courante sur laquelle se trouve le pointeur 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.

Vous pouvez aussi préciser les balises qui seront ignorées dans le troisième paramètre optionnel.

Note

`allowable_tags` a été ajouté en ^{PHP} 3.0.13 et ^{PHP} 4.0.0.

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`.

7.30.26 file_exists() : Vérifie si un fichier existe

bool **file_exists** (string filename)

file_exists retourne ^{TRUE} si le fichier ^{filename} existe, et ^{FALSE} sinon.

Note

Utilisation des dossiers partagés Windows

Sous Windows, utilisez le format de chemin `//computername/share/filename` ou `\\\\computername\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 plus de détails.

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 is_readable, is_writable, is_file et file.

7.30.27 file_get_contents() : Lit tout un fichier dans une chaîne

string **file_get_contents** (string filename , int use_include_path , resource context)

file_get_contents est identique à la fonction readfile, hormis le fait que file_get_contents retourne le fichier ^{filename} dans une chaîne.

Note

Cette fonction a été introduite en ^{PHP} 4.3.0.

Note

Cette fonction gère les chaînes binaires

Array

Note

Le support de l'argument ^{context} sera ajouté en ^{PHP} 5.0.

Voir aussi fgets, file, fread, include et readfile.

7.30.28 file_put_contents() : Ecrit une chaîne dans un fichier

int **file_put_contents** (string filename , string data , int flags , resource context)

`file_put_contents` revient à appeler successivement `fopen`, `fwrite`, et `fclose`.

Le paramètre ^{flags} peut valoir `FILE_USE_INCLUDE_PATH` et/ou `FILE_APPEND`, mais `FILE_USE_INCLUDE_PATH` doit être utilisé avec prudence.

Note

Cette fonction gère les chaînes binaires

Array

Voir aussi `fopen`, `fwrite`, `fclose` et `file_get_contents`.

7.30.29 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 `readfile`, hormis le fait que `file` retourne le fichier 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.

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`.

Vous pouvez utiliser l'option `use_include_path` : en la mettant à "1", vous rechercherez aussi dans le dossier `include_path`.

Exemple avec `file`

```
<?php
// Lit une page web dans un tableau.
$lines = file ('http://www.exemple.com/');

// Affiche toutes les lignes du tableau comme code HTML, avec les numéros de ligne
foreach ($lines as $line_num => $line) {
    echo "Ligne No <b>$line_num</b> : " . 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.exemple.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.

Array

Voir aussi `readfile`, `fopen`, `fsockopen`, `popen`, `file_get_contents` et `include`.

7.30.30 filemtime() : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois.

```
int filemtime ( string filename )
```

filemtime renvoie la date à laquelle le fichier a été accédé pour la dernière fois, ou `FALSE` en cas d'erreur.

Note
Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

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.

7.30.31 filectime() : Renvoie l'heure à laquelle l'inode a été accédé pour la dernière fois.

```
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 filemtime .

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 plus de détails.

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.

7.30.32 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.

Note
Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

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.

7.30.33 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](#) plus de détails.

Note

Cette fonction ne fonctionne pas avec les [fichiers distants](#) , car le fichier utilisé doit être accessible sur le système de fichiers local.

7.30.34 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](#) plus de détails.

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 [filectime](#) , [stat](#) , [touch](#) et [getlastmod](#) .

7.30.35 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](#) plus de détails.

Note

Cette fonction ne fonctionne pas avec les [fichiers distants](#) , car le fichier utilisé doit être accessible sur le système de fichiers local.

7.30.36 fileperms() : Renvoie les permissions affectées au 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](#) plus de détails.

Note

Cette fonction ne fonctionne pas avec les [fichiers distants](#), car le fichier utilisé doit être accessible sur le système de fichiers local.

7.30.37 filesize() : Renvoie la taille du fichier

```
int filesize ( string filename )
```

`filesize` renvoie la taille du fichier `filename`, ou `FALSE` en cas d'erreur.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) plus de détails.

Note

Cette fonction ne fonctionne pas avec les [fichiers distants](#), car le fichier utilisé doit être accessible sur le système de fichiers local.

7.30.38 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` ne fonctionne pas sur les [fichiers distants](#). Les fichiers doivent être accessibles par le système de fichier du serveur.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) plus de détails.

Note

Cette fonction ne fonctionne pas avec les [fichiers distants](#), car le fichier utilisé doit être accessible sur le système de fichiers local.

7.30.39 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.

`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 vaut par défaut `TRUE`, si le verrou doit bloquer le script (condition d'erreur EWOULDBLOCK).

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

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 thread 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).

7.30.40 fnmatch() : Repère un fichier à partir d'un masque

```
array fnmatch ( string pattern , string string , int flags )
```

`fnmatch` vérifie si la chaîne `string` va passer le masque shell `pattern`.

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, leur expression 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 ...";
}
?>
```

Voir aussi [glob](#) , [ereg](#) et [preg_match](#) .

7.30.41 fopen() : Ouverture d'un fichier ou d'une URL

```
resource fopen ( string filename , string mode , int use_include_path , resource zcontext )
```

`fopen` crée une ressource nommée, spécifiée par le paramètre `filename` , sous la forme d'un flôt. 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 flôt 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'ouverte va échouer.

Note

La liste des protocoles supportés est disponible [Liste des protocoles supportés](#) . Certains protocoles (appelé aussi `wrappers` ou gestionnaires) supportent des `contexte` 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. (i.e. l'option de `php.ini user_agent` est utilisée par le gestionnaire `http`) Pour une description des `contexts` et du paramètre `zcontext` , voyez [Flôts](#) .

`mode` peut prendre les valeurs suivantes :

- '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 file. 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.

De plus, `mode` peut contenir la lettre 'b'. Cette option n'est utile que sur les systèmes qui font la différence entre les fichiers binaires et les fichiers textes (en bref, c'est une fonctionnalité Windows, totalement inutile sous Unix). S'il n'est pas nécessaire, il sera ignoré.

Vous pouvez utiliser le troisième paramètre optionnel pour explorer le dossier include_path , en le mettant à 1.

Le paramètre optionnel `zcontext` est utilisé pour spécifier des paramètres d'optimisation et de callback.

Si l'ouverture échoue, `fopen` retournera `FALSE` .

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 échapper 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");
?>
```

Voir aussi [Liste des protocoles supportés](#), [fclose](#), [fgets](#), [fsockopen](#), [file](#), [file_exists](#), [is_readable](#), [socket_set_timeout](#) et [open](#).

7.30.42 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 passé à la sortie standard.

Le pointeur de fichier doit être valide, et doit avoir été correctement ouvert par [fopen](#), [open](#), ou [fsockopen](#). Après lecture, `fpassthru` va fermer le fichier (le pointeur `handle` sera alors invalide).

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 encouragés 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.

Voir aussi [readfile](#), [fopen](#), [open](#) et [fsockopen](#).

7.30.43 fputs() : Ecrit dans un fichier

```
int fputs ( resource handle , string str , int length )
```

`fputs` est un alias de [fwrite](#), et lui est identique en tous points. Notez que `length` est un paramètre optionnel, et s'il n'est pas spécifié, toute la chaîne est écrite.

7.30.44 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 qu'une erreur survient (le premier des trois).

Exemple avec **fread**

```
<?php
// Lit un fichier, et le place dans une chaîne
$filename = "/usr/local/something.txt";
$handle = fopen ($filename, "r");
$contents = fread ($handle, filesize ($filename));
fclose ($handle);
?>
```

Note

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**.

Note

Utilisation des fichiers binaires avec **fread**

```
<?php
$filename = "c:\\fichiers\\uneimage.gif";
$handle = fopen ($filename, "rb");
$contents = fread ($handle, filesize ($filename));
fclose ($handle);
?>
```

Voir aussi **fwrite**, **fopen**, **fsockopen**, **popen**, **foets**, **foetss**, **file** et **fpasssthru**.

7.30.45 fscanf() : Analyse un fichier en fonction d'un format

mixed **fscanf** (resource handle , string format , string var1)

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 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 flût d'entrée. Cela signifie qu'une tabulation `\t` dans la chaîne de format peut remplacer un espace simple dans le flût d'entrée.

Exemple avec **fscanf**

```
<?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);
?>
```

utilisateurs.txt

```
javier  argonaute      pe
```

```
hiroshi sculpteur      jp
robert hockeyuer      us
luigi fleuriste        it
```

Note

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.

Voir aussi [fread](#) , [fgets](#) , [fgetss](#) , [sscanf](#) , [printf](#) et [sprintf](#) .

7.30.46 fseek() : Modifie le 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.

[fseek](#) ne peut pas être utilisé sur les pointeurs retournés par [fopen](#) s'ils sont au format HTTP ou FTP.

Note

Le paramètre ^{whence} a été ajouté en ^{PHP} 4.0.0.

Voir aussi [ftell](#) et [rewind](#) .

7.30.47 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.

[fstat](#) retourne un tableau avec les éléments suivants :

- volume
- inode
- nombre de liens
- id de l'utilisateur propriétaire
- id du groupe propriétaire
- type du volume de l'inode *

- taille en octets
- date du dernier accès
- date de la dernière modification
- date du dernier changement
- taille de bloc du système pour les entrées/sorties(*)
- 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 plus de détails.
Note
Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.

7.30.48 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](#).

Voir aussi [fopen](#), [popen](#), [fseek](#) et [rewind](#).

7.30.49 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.

Voir aussi [fopen](#) et [fseek](#)

Voir aussi [fopen](#) et [fseek](#).

7.30.50 fwrite() : Ecriture du 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 de bytes écrites 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 d'écriture dans un fichier 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($filename, 'a')) {
        print "Impossible d'ouvrir le fichier ($filename)";
        exit;
    }

    // Write $somecontent to our opened file.
    if (!fwrite($handle, $somecontent)) {
        print "Impossible d'écrire dans le fichier ($filename)";
        exit;
    }

    print "L'écriture de ($somecontent) dans le fichier ($filename) a réussi";

    fclose($handle);
} else {
    print "Le fichier $filename n'est pas accessible en écriture.";
}
?>
```

Voir aussi [fread](#), [fopen](#), [fsockopen](#), [popen](#) et [fputs](#).

7.30.51 glob() : Recherche des chemins qui vérifie un masque

array **glob** (string pattern , int flags)

[glob](#) recherche tous les chemins qui vérifient le masque `pattern`, en suivant les règles du shell. 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.

Un moyen pratique pour remplacer [opendir](#) par [glob](#)

```
<?php
foreach (glob("*.txt") as $filename) {
    echo "$filename occupe " . filesize($filename) . " octets\n";
}
?>
```

Le résultat du script ci-dessus pourrait être :

```
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](#).

7.30.52 is_dir() : Indique si le nom de 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](#) plus de détails.

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 [chdir](#) , [opendir](#) , [is_file](#) et [is_link](#) .

7.30.53 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`.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) plus de détails.

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 [is_file](#) et [is_link](#) .

7.30.54 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).

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) plus de détails.

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 [is_dir](#) et [is_link](#) .

7.30.55 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` plus de détails.

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 `is_dir`, `is_file` et `readlink`.

7.30.56 `is_readable()` : Indique un fichier est autorisé en lecture

```
bool is_readable ( string filename )
```

`is_readable` retourne ^{TRUE} si `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).

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction `clearstatcache` plus de détails.

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 `is_writable`, `file_exists` et `fgets`.

7.30.57 `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.

`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.

<?php

```

/* Test utilisateur du fichier téléchargé. */
function is_uploaded_file($filename) {
    if (!$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($_HTTP_POST_FILES['userfile'])) {
    copy($_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](#) .

7.30.58 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).

Note
Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

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 [is_readable](#) , [file_exists](#) et [fwrite](#) .

7.30.59 is_writeable() : Indique si un fichier est autorisé en écriture

```
bool is_writeable ( string filename )
```

[is_writeable](#) est un alias de [is_writable](#) .

7.30.60 link() : Crée un lien

```
bool link ( string target , string link )
```

[link](#) crée un lien en le fichier source ^{link} et le fichier destination ^{target} .

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 [symlink](#) , [readlink](#) et [linkinfo](#) .

7.30.61 linkinfo() : Renvoie les informations à propos d'un lien

```
int linkinfo ( string path )
```

[linkinfo](#) renvoie les informations à propos d'un lien, c'est-à-dire le champ `st_dev` de la structure d'information UNIX (comme en langage C). [linkinfo](#) sert à vérifier si un lien (repéré par `path`) existe (en utilisant la même méthode que la macro `S_ISLNK` de `stat.h`). [linkinfo](#) retourne `FALSE` en cas d'erreur.

Voir aussi [symlink](#), [link](#) et [readlink](#).

7.30.62 lstat() : Retourne les informations à propos d'un fichier ou d'un lien symbolique.

```
array lstat ( string filename )
```

[lstat](#) renvoie les informations à propos du fichier `filename`. Cette fonction est identique à la fonction [stat](#) à part 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](#).

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
Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

Voir aussi [stat](#).

7.30.63 mkdir() : Crée un dossier

```
bool mkdir ( string pathname, int mode )
```

[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 <code>mode</code> 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
<pre><?php mkdir ("/chemin/de/mon/dossier", 0700); ?></pre>

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [rmdir](#).

7.30.64 `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
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
<code>move_uploaded_file</code> n'est pas affectée par les restrictions du <code>safe mode</code> . Ce n'est pas un problème, car <code>move_uploaded_file</code> ne fonctionne que sur des fichiers qui ont été téléchargés avec <code>PHP</code> .

Attention
Si le fichier de destination existe déjà, il sera écrasé.

Voir aussi [is_uploaded_file](#).

7.30.65 `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 configuration qui s'y trouve 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 <code>php.ini</code> . 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 <code>.ini</code> contient des données non alphanumérique, 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 .

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 exemple.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.exemple.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);

?>
```

Cet exemple va produire :

```
Array
(
    [un] => 1
    [cinq] => 5
    [animal] => dodu dodo
    [chemin] => /usr/local/bin
    [URL] => http://www.exemple.com/~username
)
Array
(
    [premiere_section] => Array
        (
            [un] => 1
            [cinq] => 5
            [animal] => dodu dodo
        )

    [seconde_section] => Array
        (
            [chemin] => /usr/local/bin
            [URL] => http://www.exemple.com/~utilisateur
        )
)
```

7.30.66 pathinfo() : Retourne des informations sur un chemin système

array **pathinfo** (string path)

pathinfo retourne un tableau associatif, contenant les informations sur le chemin ^{path} . Les éléments

suivants sont retournés : `dirname` , `basename` et `extension` .

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";

?>
```

Va afficher :

```
/www/htdocs
index.html
html
```

Note

Pour plus d'informations sur la lecture du chemin courant, lisez la section sur les [variables prédéfinies](#) .

Voir aussi [dirname](#) , [basename](#) , [parse_url](#) et [realpath](#) .

7.30.67 `pclose()` : Ferme un processus de pointeur de fichier

```
int pclose ( resource handle )
```

[pclose](#) ferme un processus de pointeur de fichier 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](#) .

7.30.68 `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.

[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 puissions lire stderr */
$handle = popen('/chemin/vers/spooge 2>&1', 'r');
echo "'$handle'; " . gettype($handle) . "\n";
$read = fread($handle, 2096);
echo $read;
pclose($handle);
?>
```

Voir aussi `pclose` , `fopen` et `proc_open` .

7.30.69 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 pour explorer le dossier `include_path` , en passant la valeur de 1.

Voir aussi `fpasssthru` , `file` , `fopen` , `include` , `require` et `virtual` .

7.30.70 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.

Voir aussi `is_link` , `symlink` et `linkinfo` .

7.30.71 realpath() : Retourne le chemin canonique absolu

```
string realpath ( string path )
```

`realpath` résoud 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 `'../'`.

Exemple avec `realpath`

```
<?php
$real_path = realpath ("../../index.php");
?>
```

Voir aussi [basename](#) , [dirname](#) et [pathinfo](#) .

7.30.72 rename() : Renomme un fichier

```
bool rename ( string oldname , string newname )
```

[rename](#) tente de renommer le fichier `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");
?>
```

Voir aussi [copy](#) , [unlink](#) et [move_uploaded_file](#)

Voir aussi [copy](#) , [unlink](#) et [move_uploaded_file](#) .

7.30.73 rewind() : Replace le pointeur de fichier au début

```
bool rewind ( resource handle )
```

[rewind](#) replace 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"), toutes les données que vous écrirez dans ce fichier seront toujours ajoutées, indépendamment de la position du pointeur de fichier.

Voir aussi [fseek](#) et [ftell](#) .

7.30.74 rmdir() : Efface un dossier

```
bool rmdir ( string dirname )
```

[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.

Voir aussi [mkdir](#) et [unlink](#) .

7.30.75 set_file_buffer() : Alias de [stream_set_write_buffer](#)

```
int set_file_buffer ( int handle , int buffer )
```

[set_file_buffer](#) est un alias de [stream_set_write_buffer](#) .

7.30.76 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 clefs, comme mentionné à côté de chaque paramètre; ceci est disponible depuis PHP 4.0.6 :

1. 0 : volume (ou encore ^{dev})
2. 1 : inode (ou encore ^{ino})
3. 2 : droits d'accès au fichier (mode de protection du inode). A convertir en octal. Voir aussi [fileperms](#) . (ou encore ^{mode})
4. 3 : nombre de liens (ou encore ^{nlink})
5. 4 : id de l'utilisateur propriétaire (ou encore ^{uid})
6. 5 : id du groupe propriétaire (ou encore ^{gid})
7. 6 : type du volume de l'inode * (ou encore ^{rdev})
8. 7 : taille en octets (ou encore ^{size})
9. 8 : date du dernier accès (ou encore ^{atime})
10. 9 : date de la dernière modification (ou encore ^{mtime})
11. 10 : date du dernier changement (ou encore ^{ctime})
12. 11 : taille de bloc du système pour les entrées/sorties * (ou encore ^{blksize})
13. 12 : nombre de blocs alloués (ou encore ^{blocks})

* – 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 plus de détails.
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 [lstat](#) , [fstat](#) , [filemtime](#) et [filegroup](#) .

7.30.77 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.

Voir aussi [link](#) , [readlink](#) et [linkinfo](#) .

7.30.78 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.

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.

[tempnam](#) retourne le nom du fichier temporaire, ou la chaîne ^{NULL} en cas d'échec.

Exemple avec [tempnam](#)

```
$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](#) .

7.30.79 tmpfile() : Crée un fichier temporaire

resource **tmpfile** (void)

[tmpfile](#) crée un fichier temporaire avec un nom unique, ouvert en écriture, 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");
fclose($tmp); // Ceci supprime le fichier temporaire
?>
```

Voir aussi [tempnam](#) .

7.30.80 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 nommé `filename` à la date de `time`. Si `time` est omis, c'est l'heure courante qui est utilisée.

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) ) {
    print "La date de modification de $NomDeFichier a
        été fixée à maintenant";
} else {
    print "Désolé, il est impossible de changer
        la date de modification de $NomDeFichier";
}
?>
```

7.30.81 umask() : Change le "umask" courant

int **umask** (int mask)

umask change le umask de ^{PHP} : `mask & 0777` et retourne le vieux umask. Lorsque ^{PHP} est utilisé comme module de serveur, le umask reprend sa valeur à la fin de chaque script.

umask appelé sans arguments, retourne simplement le umask courant.

7.30.82 unlink() : Efface un fichier

bool **unlink** (string filename)

unlink efface `filename`. Identique à la fonction Unix C `unlink()`.

Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

Voir aussi rmdir.

7.31 Forms Data Format

7.31.1 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 remplis 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..

7.31.2 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 librairie FDF toolkit est disponible sous forme de librairie compilée, éditée par Adobe, sur les systèmes d'exploitation Win32, Linux, Solaris et AIX.

7.31.3 Installation

You must compile PHP with `--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 librairie `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 librairie 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 librairies **fdftk.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).

7.31.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.31.5 Types de ressources

7.31.5.1 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`.

7.31.6 Constantes prédefinies

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)
FDFff (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)

```

7.31.7 Exemples

Les exemples suivants montre 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 : '<B>$volume</B>'\<BR>";

$date = fdf_get_value($fdf, "date");
echo "La valeur du champ date était '<B>$date</B>'\<br>";
$comment = fdf_get_value($fdf, "comment");
echo "La valeur du champ comment était '<B>$comment</B>'\<br>";
if(fdf_get_value($fdf, "show_publisher") == "On") {
    $publisher = fdf_get_value($fdf, "publisher");
    echo "La valeur du champ Publisher était : '<B>$publisher</B>'\<br>";
} else
    echo "La valeur du champ Publisher ne doit pas être affichée.<br>";
if(fdf_get_value($fdf, "show_preparer") == "On") {
    $preparer = fdf_get_value($fdf, "preparer");
    echo "La valeur du champ Preparer était : '<B>$preparer</B>'\<br>";
} else
    echo "La valeur du champ Preparer ne doit pas être affiché.<br>";
fdf_close($fdf);
?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [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_opt](#)
- [fdf_set_status](#)

- [fdf_set_submit_form_action](#)
- [fdf_set_target_frame](#)
- [fdf_set_value](#)
- [fdf_set_version](#)

7.31.9 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.

7.31.10 fdf_close() : Ferme un document FDF

```
bool fdf_close ( resource fdf_document )
```

[fdf_close](#) ferme le document FDF.

Voir aussi [fdf_open](#) .

7.31.11 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

```
<?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);
unlink("outtest.fdf");
?>
```

Voir aussi [fdf_close](#) , [fdf_save](#) et [fdf_open](#) .

7.31.12 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.

7.31.13 `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 si 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` .

7.31.14 `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 homis. Sinon, `fdf_error()` est une fonction pratique pour le lire : `fdf_error(fdf_errno())` .

Voir aussi `fdf_errno` .

7.31.15 `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'apparance 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'apparance est stockée dans le fichier `filename` .

7.31.16 `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 uploadé 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 MIME du fichier, si il a été fourni dans le document FDF

Enregistrement d'un fichier téléchargé

```
<?php
    $fdf = fdf_open_string($HTTP_FDF_DATA);
    $data = fdf_get_attachment($fdf, "filename", "/tmpdir");
    echo "Le fichier téléchargé est stocké dans {$data['path']}";
?>
```

7.31.17 fdf_get_encoding() : Lit la valeur de la clé /Encoding

```
string fdf_get_encoding ( resource fdf_document )
```

The [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](#) .

7.31.18 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](#) .

7.31.19 fdf_get_flags() : Lit les attributs d'un champ FDF

```
fdf_get_flags ( void )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.31.20 fdf_get_opt() : Lit une valeur dans un tableau de valeur d'un champ FDF

```
mixed fdf_get_opt ( resource fdfdoc, string fieldname, int element )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.31.21 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](#) .

7.31.22 `fdf_get_value()` : Retourne la valeur d'un champ FDF

```
string 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 optionnelle `which`, sous la forme d'un entier donc 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 ajouté en PHP 4.3.

Voir aussi `fdf_set_value`.

7.31.23 `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`.

7.31.24 `fdf_header()` : Emet les en-têtes HTTP spécifiques à FDF

```
bool 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`.

7.31.25 `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 \$fd = fdf_open(\$HTTP_FDF_DATA); for(\$field = fdf_next_field_name(\$fd); \$field != ""; \$field = fdf_next_field_name(\$fd, \$field)) { echo "Champ : \$field\n"; } ?></pre>

Voir aussi `fdf_enum_fields` et `fdf_get_value`.

7.31.26 `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`.

7.31.27 `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. Actuellement, le fichier doit être créé 'manuellement', en utilisant la fonction `fopen` et en y écrivant le contenu du tableau `HTTP_FDF_DATA` avec la fonction `fwrite`. Un mécanisme comparable aux formulaires HTML qui créent une variable pour chaque champ entrant, n'existe pas.

Accéder aux données du formulaire
<pre><?php // Sauver le fichier FDF dans un fichier temporaire. \$fdf = fopen("test.fdf", "w"); fwrite(\$fdf, \$HTTP_FDF_DATA, strlen(\$HTTP_FDF_DATA)); fclose(\$fdf); // Ouvrir le fichier temporaire, et utiliser les données. \$fdf = fdf_open("test.fdf"); ... fdf_close(\$fdf); ?></pre>

Voir aussi `fdf_open_string`, `fdf_close`, `fdf_create` et `fdf_save`.

7.31.28 `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.

7.31.29 `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 chaîne de caractères.

Lire un document FDF sous forme de chaîne

```

<?php
$fd = fdf_create();
fdf_set_value($fd, "foo", "bar");
$str = fdf_save_string($fd);
fdf_close($fd);
echo $str;
?>
```

affichera

```

%PDF-1.2
%âãÏÓ
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](#) .

7.31.30 `fdf_save()` : Sauver un document FDF

bool **fdf_save** (resource fdf_document , string filename)

fdf_save sauve un document FDF. Le FDF Toolkit fournit un moyen d'envoyer le contenu d'un document FDF à au fichier de sortie stdout si le paramètre `filename` vaut `'.'`. Ceci ne fonctionne pas si PHP est sous la forme d'un module Apache. Dans ce cas, il faudra écrire le résultat dans un fichier, et utiliser [fpassthru](#) pour l'afficher au client.

Voir aussi [fdf_save_string](#) , [fdf_create](#) et [fdf_close](#) .

7.31.31 `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 1=FDFFormalAP, 2=FDFFRolloverAP, 3=FDFFDownAP.

7.31.32 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. Pour Acrobat 5.0, un nom valide peut notamment être " Shift-JIS ", " UHC ", " GBK " ou " BigFive " .

fdf_set_encoding a été ajoutée en PHP 4.1.0.

7.31.33 fdf_set_file() : Assigne une valeur à un champ 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-remplis. Dans un environnement web, c'est une URL (e.g. <http://testfdf/resultlabel.pdf>).

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
<pre><?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(); ?></pre>

Voir aussi fdf_get_file et fdf_set_target_frame .

7.31.34 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 .

7.31.35 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 un javascript au champ `fieldname`, exécuté lors de la validation d'un formulaire.

Voir aussi [fdf_set_submit_form_action](#).

7.31.36 **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](#).

7.31.37 **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.

Voir aussi [fdf_get_status](#).

7.31.38 **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](#).

7.31.39 **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](#).

7.31.40 **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 d'un champ. Le dernier paramètre détermine si la valeur doit être convertie en nom PDF (`isname = 1`) ou affecter une chaîne PDF à un contrôle (`isname = 0`).

Voir aussi [fdf_get_value](#) et [fdf_remove_item](#) .

7.31.41 fdf_set_version() : Modifie le numéro de version du fichier FDF

string **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](#) .

7.32 FriBiDi

7.32.1 Introduction

FriBiDi est une implementation libre de [l'algorithme bidirectionnel unicode](#) .

7.32.2 Pré-requis

Vous devez télécharger et installer la librairie [FriBiDi](#) .

7.32.3 Installation

Pour activer le support FriBiDi, vous devez compiler ^{PHP} avec l'option `--with-fribidi[=DIR]` où DIR est le dossier d'installation de FriBiDi.

7.32.4 Configuration à l'exécution

7.32.5 Types de ressources

7.32.6 Constantes prédefinies

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 prédefinies](#)
- [fribidi_log2vis](#)

7.33 FTP

7.33.1 Introduction

FTP : File Transfer Protocol (Protocole de transfert de fichiers). Ces fonctions implémentent un client pour accéder aux serveurs FTP, comme défini dans <http://www.faqs.org/rfcs/rfc959.html> .

7.33.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.33.3 Installation

Pour activer le module FTP de votre configuration PHP, il faut utiliser l'option `--enable-ftp` en PHP 4, et l'option `--with-ftp` en PHP 3 avec le script de configuration.

La version Windows de ^{PHP} dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

7.33.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.33.5 Types de ressources

This extension uses one resource type, which is the link identifier of the FTP connection.

7.33.6 Constantes prédefinies

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

7.33.7 Exemples

Exemple FTP

```
<?php
// création de la connexion
$conn_id = ftp_connect($ftp_server);

// authentification avec nom de compte et 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 à $ftp_server avec $ftp_user_name";
    die;
} else {
    echo "Connected to $ftp_server, for user $ftp_user_name";
}

// téléchargement d'un fichier
$upload = ftp_put($conn_id, $destination_file, $source_file, FTP_BINARY);

// Vérification de téléchargement
if (!$upload) {
    echo "Le téléchargement FTP a échoué!";
} else {
    echo "Téléchargement de $source_file sur $ftp_server en $destination_file";
}

// fermeture de la connexion FTP.
ftp_close($conn_id);
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [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](#)

7.33.9 ftp_chdir() : Change le dossier FTP courant

bool **ftp_chdir** (resource ftp_stream , string directory)

[ftp_chdir](#) change le dossier courant en `directory` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec [ftp_chdir](#)

```
<?php
// Connexion
$conn_id = ftp_connect($ftp_server);

// Identification par login et mot de passe
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);

// Vérification de la connexion
if ((!$conn_id) || (!$login_result)) {
    die("La connexion FTP a échoué!");
}

echo "Dossier courant : ", ftp_pwd($conn_id), "\n";

if (@ftp_chdir($conn_id, "somedir")) {
    echo "Le dossier courant est maintenant : ", ftp_pwd($conn_id), "\n";
} else {
    echo "Impossible de changer le dossier courant.\n";
}
?>
```

Voir aussi [ftp_cdup](#) .

7.33.10 ftp_chmod() : Modifie les droits d'un fichier via FTP

string **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 leur attribuant les droits de `mode` , spécifié sous la forme d'un entier en base octale.

[ftp_chmod](#) retourne les permissions du fichier en cas de succès et `FALSE` sinon.

Voir aussi [chmod](#) .

7.33.11 ftp_close() : Ferme une connexion FTP

```
void ftp_close ( resource ftp_stream )
```

Note

[ftp_close](#) n'est disponible que dans les sources du CVS.

[ftp_close](#) ferme la connexion `ftp_stream` et libère les ressources. Après avoir appelé cette fonction, vous ne pouvez plus utiliser l'ancienne connexion, et vous devez en ouvrir une nouvelle avec [ftp_connect](#) .

7.33.12 ftp_connect() : Ouvre une connexion FTP

```
resource ftp_connect ( string host , int port , int timeout )
```

[ftp_connect](#) retourne un flôt ^{FTP} en cas de succès, et ^{FALSE} sinon.

[ftp_connect](#) ouvre une connexion FTP avec l'hôte ^{host} . Le paramètre ^{port} spécifie le port de connexion. S'il est omis, le port 21 sera utilisé.

Le paramètre ^{timeout} spécifie la durée d'expiration des opérations réseaux ultérieures. Si il est omis, il vaut par défaut 90 secondes. Ce délai d'expiration peut être modifiée et lu à tout moment avec les fonctions [ftp_set_option](#) et [ftp_get_option](#) .

Note

Le paramètre ^{timeout} a été ajouté en ^{PHP} 4.2.0.

Exemple avec [ftp_connect](#)

```
<?php
$ftp_server = "ftp.exemple.com";
// ouvre une connexion ou échoue
$conn_id = ftp_connect($ftp_server) or die("Impossible de se connecter à $ftp_server");
?>
```

Voir aussi [ftp_close](#) et [ftp_ssl_connect](#) .

7.33.13 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.

Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.33.14 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`.
ftp_exec retourne le résultat de la commande si elle a réussi, et `FALSE` sinon.

Voir aussi ftp_raw.

7.33.15 ftp_fget() : Télécharge un fichier via FTP dans un flôt 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`.
 Le mode de transfert `mode` spécifié doit être `FTP_ASCII` ou `FTP_BINARY`.

Note

La paramètre `resumepos` a été ajouté en PHP 4.3.0.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi ftp_get, ftp_nb_get et ftp_nb_fget.

7.33.16 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. Le résultat est stocké dans le fichier `remote_file` sur le serveur FTP. Le mode de transfert `mode` spécifié doit être `FTP_ASCII` ou `FTP_BINARY`.

Note

Le paramètre `startpos` a été ajouté en PHP 4.3.0.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi ftp_put, ftp_nb_fput et ftp_nb_put.

7.33.17 ftp_get_option() : Lit différentes options pour la connexion FTP courante

```
mixed ftp_get_option ( resource ftp_stream , int option )
```

Note

ftp_get_option n'est disponible que dans le CVS.

ftp_get_option retourne la valeur de l'option `option` en cas de succès, et `FALSE` si l'option `option` n'est pas supportée. Dans ce dernier cas, un message d'erreur est alors affiché.

ftp_get_option retourne la valeur de l'option `option` pour la connexion `ftp_stream`. Actuellement, les options suivantes sont supportées :

FTP_TIMEOUT_SEC

Retourne le délai d'expiration pour les opérations liées au réseau.

Exemple avec `ftp_get_option`

```
<?php
// Lit le délai d'expiration pour la connexion en cours
$timeout = ftp_get_option($conn_id, FTP_TIMEOUT_SEC);
?>
```

Voir aussi `ftp_set_option`.

7.33.18 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`. Le mode de transfert `mode` spécifié doit être soit `FTP_ASCII` ou `FTP_BINARY`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi `ftp_fget`, `ftp_nb_get` et `ftp_nb_fget`.

7.33.19 ftp_login() : Identification avec une connexion FTP

```
bool ftp_login ( resource ftp_stream , string username , string password )
```

`ftp_login` authentifie le flôt FTP.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec `ftp_login`

```
<?php

$ftp_server = "ftp.exemple.com";
$ftp_user = "foo";
$ftp_pass = "bar";

// Connexion
$conn_id = ftp_connect($ftp_server) or die("Couldn't connect to $ftp_server");

// Tentative de login
if (@ftp_login($conn_id, $ftp_user, $ftp_pass)) {
    echo "Connecté avec $ftp_user@$ftp_server\n";
} else {
    echo "Impossible de se connecter avec $ftp_user\n";
}

?>
```

7.33.20 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` retourne un UNIX timestamp en cas de succès, et `FALSE` sinon.

`ftp_mdtm` lit la date de dernière modification d'un fichier et retourne le UNIX timestamp. Si une erreur survient, ou si le fichier n'existe pas, la valeur `-1` est retournée. Certains serveurs FTP ne supportent pas cette fonction.

7.33.21 ftp_mkdir() : Crée un dossier sur un serveur FTP

```
string ftp_mkdir ( resource ftp_stream , string directory )
```

ftp_mkdir retourne le nom du dossier ainsi créé en cas de succès, et `FALSE` sinon.

ftp_mkdir crée le dossier nommé `directory`.

Voir aussi ftp_rmdir.

7.33.22 ftp_nb_continue() : Reprend le téléchargement d'un fichier (non bloquant)

```
bool ftp_nb_continue ( resource ftp_stream )
```

ftp_nb_continue reprend le téléchargement d'un fichier, de manière asynchrone.

ftp_nb_continue retourne `FTP_FAILED` ou `FTP_FINISHED`, ou encore `FTP_MOREDATA`.

7.33.23 ftp_nb_fget() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)

```
bool 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`, puis écrit ce fichier dans le pointeur de fichier `handle`. Le mode de transfert `mode` spécifié peut être `FTP_ASCII` ou `FTP_BINARY`. La différence entre cette fonction et ftp_fget est que cette fonction peut lire le fichier de manière asynchrone, de manière à ce que votre programme fasse autre chose, pendant que le fichier est téléchargé.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi ftp_nb_get, ftp_nb_continue, ftp_fget et ftp_get.

7.33.24 ftp_nb_fput() : Ecrit un fichier sur un serveur FTP, et le lit depuis un fichier (non bloquant)

```
bool 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`, puis lui donne le nom de `handle`. Le mode de transfert `mode` spécifié peut être `FTP_ASCII` ou `FTP_BINARY`. La différence entre cette fonction et ftp_fput est que cette fonction peut lire le fichier de manière asynchrone, de manière à ce que votre programme fasse autre chose, pendant que le fichier est téléchargé.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi ftp_nb_put, ftp_nb_continue, ftp_put et ftp_fput.

7.33.25 ftp_nb_get() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)

`bool 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`, puis écrit ce fichier dans le pointeur de fichier `handle`. Le mode de transfert `mode` spécifié peut être `FTP_ASCII` ou `FTP_BINARY`. La différence entre cette fonction et `ftp_fget` est que cette fonction peut lire le fichier de manière asynchrone, de manière à ce que votre programme fasse autre chose, pendant que le fichier est téléchargé.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec `ftp_nb_get`

```
// Initialisation
$ret = ftp_nb_get($my_connection, "test", "README", FTP_BINARY);
while ($ret == FTP_MOREDATA) {

    // Faites ce que vous voulez
    echo ".";

    // Continue le chargement
    $ret = ftp_nb_continue ($my_connection);
}
if ($ret != FTP_FINISHED) {
    echo "Il y a eu une erreur lors du chargement du fichier...";
    exit(1);
}
```

Reprise d'un chargement avec `ftp_nb_get`

```
// Initialisation
$ret = ftp_nb_get ($my_connection, "test", "README", FTP_BINARY,
    filesize("test"));
// Ou : $ret = ftp_nb_get ($my_connection, "test", "README",
//      FTP_BINARY, FTP_AUTORESUME);
while ($ret == FTP_MOREDATA) {

    // Faites ce que vous voulez
    echo ".";

    // Continue le chargement
    $ret = ftp_nb_continue ($my_connection);
}
if ($ret != FTP_FINISHED) {
    echo "Il y a eu une erreur lors du chargement du fichier...";
    exit(1);
}
```

Reprise du téléchargement à la position 100 avec `ftp_nb_get`

```
// Désactive l'option de Autoseek
ftp_set_option ($my_connection, FTP_AUTOSEEK, FALSE);

// Initialisation
$ret = ftp_nb_get ($my_connection, "newfile", "README", FTP_BINARY, 100);
while ($ret == FTP_MOREDATA) {

    ...

    // Continue le chargement
    $ret = ftp_nb_continue ($my_connection);
}
```

Dans l'exemple ci-dessus, `"newfile"` est 100 octets plus petit que `"README"` sur le site FTP, car nous avons commencé à lire à l'offset 100. Si nous n'avions pas désactivé l'option `FTP_AUTOSEEK`, les premiers 100 octets du fichier seraient complétés avec `'\0'`.

Voir aussi `ftp_nb_fget`, `ftp_nb_continue`, `ftp_get` et `ftp_fget`.

7.33.26 ftp_nb_put() : Envoie un fichier sur le serveur FTP (non-bloquant)

`bool 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`, puis lui donne le nom de `handle`. Le mode de transfert `mode` spécifié peut être `FTP_ASCII` ou `FTP_BINARY`. La différence entre cette fonction et `ftp_fput` est que cette fonction peut lire le fichier de manière asynchrone, de manière à ce que votre programme fasse autre chose, pendant que le fichier est téléchargé.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec `ftp_nb_put`

```
// Initialisation
$ret = ftp_nb_put($my_connection, "test.remote", "test.local", FTP_BINARY);
while ($ret == FTP_MOREDATA) {

    // Faites ce que vous voulez
    echo ".";

    // Continue le téléchargement...
    $ret = ftp_nb_continue ($my_connection);
}
if ($ret != FTP_FINISHED) {
    echo "Il y a eu une erreur lors du chargement...";
    exit(1);
}
```

Resuming an upload with `ftp_nb_put`

```
// Initialisation
$ret = ftp_nb_put ($my_connection, "test.remote", "test.local",
                  FTP_BINARY, ftp_size("test.remote"));
// ou bien : $ret = ftp_nb_put ($my_connection, "test.remote", "test.local",
//                               FTP_BINARY, FTP_AUTORESUME);

while ($ret == FTP_MOREDATA) {

    // Faites ce que vous voulez
    echo ".";

    // Continue le téléchargement...
    $ret = ftp_nb_continue ($my_connection);
}
if ($ret != FTP_FINISHED) {
    echo "Il y a eu une erreur lors du chargement...";
    exit(1);
}
```

Voir aussi `ftp_nb_fput`, `ftp_nb_continue`, `ftp_put` et `ftp_fput`.

7.33.27 ftp_nlist() : Retourne la liste des fichiers dans un dossier

`array ftp_nlist ( resource ftp_stream , string directory )`

`ftp_nlist` retourne un tableau contenant les noms des fichiers en cas de succès, et `FALSE` sinon.

Exemple avec `ftp_nlist`

```
<?php
// Connexion
$conn_id = ftp_connect($ftp_server);
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);

// Vérification de la connexion
if (!$conn_id || !$login_result) {
    die("La connexion a échoué !");
}
```

```
// Lire le contenu du dossier root
$content = ftp_nlist($conn_id, "/");

// Afficher les entrées
foreach ($content as $entry) {
    echo $entry, "<br />\n";
}

?>
```

Voir aussi [ftp_rawlist](#) .

7.33.28 ftp_pasv() : Active ou désactive le mode passif

```
bool ftp_pasv ( resource ftp_stream , bool pasv )
```

[ftp_pasv](#) active le mode passif si `pasv` est à `TRUE` (et le désactive si `pasv` est à `FALSE`). En mode passif, les données de connexion sont initiées par le client, plutôt que par le serveur.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.33.29 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, sous le nom de `remote_file` . Le mode de transfert `mode` spécifié doit être `FTP_ASCII` ou `FTP_BINARY` .

Note

Le paramètre `startpos` a été ajouté en PHP 4.3.0.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec [ftp_put](#)

```
<?php
$upload = ftp_put($conn_id, $destination_file, $source_file, FTP_ASCII);
?>
```

Voir aussi [ftp_fput](#) , [ftp_nb_fput](#) et [ftp_nb_put](#) .

7.33.30 ftp_pwd() : Retourne le nom du dossier courant

```
string ftp_pwd ( resource ftp_stream )
```

[ftp_pwd](#) retourne le nom du dossier courant, ou `FALSE` en cas d'erreur.

7.33.31 ftp_quit() : Ferme une connexion FTP

```
void ftp_quit ( resource ftp_stream )
```

[ftp_quit](#) ferme la connexion `ftp_stream` .

[ftp_quit](#) est un alias de [ftp_close](#) .

7.33.32 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` . [ftp_raw](#) retourne alors la réponse du serveur sous forme d'un tableau de chaînes. Aucun traitement de la réponse n'est fait et [ftp_raw](#) ne tente pas de savoir si la commande a réussi ou échouée.

Utilisation de [ftp_raw](#) pour se logger manuellement

```
<?php
$fp = ftp_connect("ftp.exemple.com");

/* Ceci est la même chose que
   ftp_login($fp, "joeblow", "secret"); */

ftp_raw($fp, "USER joeblow");
ftp_raw($fp, "PASS secret");
?>
```

Voir aussi [ftp_exec](#) .

7.33.33 ftp_rawlist() : Fait une liste détaillée de fichiers dans un dossier

array **ftp_rawlist** (resource ftp_stream , string directory)

[ftp_rawlist](#) exécute la commande FTP LIST, et retourne le résultat dans un tableau. Chaque élément du tableau correspond à une ligne du résultat de la commande. Le résultat n'est pas analysé, et est retourné brut. L'identifiant de système retourné par [ftp_systype](#) sera utile pour déterminer la façon d'interpréter le résultat.

Voir aussi [ftp_nlist](#) .

7.33.34 ftp_rename() : Renomme un fichier sur un serveur FTP

bool **ftp_rename** (resource ftp_stream , string from , string to)

[ftp_rename](#) renomme le fichier ou dossier `from` en `to` , sur le serveur `ftp_stream` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.33.35 ftp_rmdir() : Efface un dossier FTP

bool **ftp_rmdir** (resource ftp_stream , string directory)

[ftp_rmdir](#) efface le dossier `directory` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [ftp_mkdir](#) .

7.33.36 ftp_set_option() : Modifie les options de la connexion FTP

bool **ftp_set_option** (resource ftp_stream , int option , mixed value)

ftp_set_option retourne ^{TRUE} si l'option ^{option} a pu être modifiée avec la nouvelle valeur ^{value}, et ^{FALSE} sinon. Un message d'erreur sera affiché si ^{option} n'est pas supporté, ou si la valeur ^{value} n'est pas une valeur valide pour l'option ^{option}.

Cette fonction contrôle différentes options de connexion. Le paramètre ^{value} dépend de l'option ^{option} choisie. Actuellement, les options suivantes sont supportées :

FTP_TIMEOUT_SEC	Modifie le délai d'expiration pour les opérations réseau. value doit être un entier positif ou null. La valeur par défaut est de 90 secondes.
FTP_AUTOSEEK	Lorsque cette option est activée, les requêtes de GET et PUT avec le paramètre resumepos ou startpos va générer un positionnement automatique du pointeur dans le fichier. C'est le comportement par défaut.

Exemple avec ftp_set_option

```
<?php
// modifie le délai d'expiration pour les opérations
//réseau à 10 secondes
ftp_set_option($conn_id, FTP_TIMEOUT_SEC, 10);
?>
```

Voir aussi ftp_get_option.

7.33.37 ftp_site() : Envoie la commande SITE au serveur

bool **ftp_site** (resource ftp_stream , string cmd)

ftp_site envoie la commande ^{cmd} 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.

Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.33.38 ftp_size() : Retourne la taille d'un fichier.

int **ftp_size** (resource ftp_stream , string remote_file)

ftp_size retourne la taille du fichier en cas de succès, et ^{FALSE} sinon.

ftp_size retourne la taille d'un fichier sur un serveur FTP. Si une erreur survient, ou que le fichier n'existe pas, la valeur -1 est retournée. Certains serveurs FTP ne supportent pas cette fonction.

Voir aussi ftp_rawlist.

7.33.39 ftp_ssl_connect() : Ouvre une connexion FTP sécurisée avec SSL

resource **ftp_ssl_connect** (string host , int port , int timeout)

ftp_ssl_connect retourne un flôt FTP sécurisé par SSL en cas de succès et ^{FALSE} sinon.

`ftp_ssl_connect` ouvre une connexion SSL-FTP avec le serveur FTP `host`. Le paramètre `port` spécifie un port alternatif. Si il est omis, ou bien configuré à 0, alors la valeur par défaut est 21.

Le paramètre `timeout` spécifie la durée d'expiration des opérations de réseaux ultérieures. Si elle est omise, cette valeur vaut 90 secondes. La durée d'expiration des opérations réseau peut être lue et modifiée avec les fonctions `ftp_set_option` et `ftp_get_option`.

Note

Pourquoi cette fonction peut ne pas exister

`ftp_ssl_connect` est uniquement disponible si OpenSSL a été ajouté dans la liste des extensions de `PHP`. Si vous avez compilé `PHP` avec le support FTP mais que cette fonction est indisponible, alors c'est la raison.

Voir aussi `ftp_connect`.

7.33.40 ftp_systype() : Retourne un identifiant de type de serveur FTP.

```
string ftp_systype ( resource ftp_stream )
```

`ftp_systype` retourne le type de serveur, ou `FALSE` en cas d'erreur.

7.34 Fonctions

7.34.1 Introduction

Ces fonctions effectuent les manipulations liées à la gestion des fonctions.

7.34.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.34.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.34.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.34.5 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes prédefinies](#)
- [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](#)

7.34.7 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 <u>call_user_func</u>

```
<?php
function barbier ($type) {
    print "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 `function`.

Exemple avec `call_user_func` et une méthode de classe

```
<?php
class maclasse {
    function dit_bonjour() {
        print "Bonjour!\n";
    }
}

$classname = "maclasse";

call_user_func(array($classname,'dit_bonjour'));
?>
```

Voir aussi [call_user_func_array](#), [call_user_method](#) et [call_user_method_array](#).

7.34.8 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 récoltés 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, $farr) {
    for ($f=0; $f < count($farr); $f++)
        echo $farr[$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 secondes liste de fonctions anonymes\n";
process("Twas brillling and the slithy toves", "Twas the night", $garr);
?>

```

Et lorsque vous utilisez le code ci-dessus, l'affichage sera

```

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 secondes liste de fonctions anonymes
** "Twas the night" et "Twas brillling and the slithy toves"
** Ces chaînes de ressemblent!! (regarde 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); // En PHP 3 utilisez var_dump()
// affiche:
// Array
// (
//     [0] => la maison
//     [1] => une maison
//     [2] => cette maison
//     [3] => une certaine maison
// )
// un tableau de chaîne classé par taille
$sv = array("petite","moyenne","tres longue","vraiment tres longue");
print_r($sv);
// affiche:
// Array
// (
//     [0] => petite
//     [1] => moyenne
//     [2] => tres longue
//     [3] => vraiment tres longue
// )
// Tri par ordre de taille décroissant
usort($sv, create_function('$a,$b','return strlen($b) - strlen($a);'));
print_r($sv);
// outputs:
// Array
// (
//     [0] => vraiment tres longue
//     [1] => tres longue
//     [2] => moyenne
//     [3] => petite
// )
?>

```

7.34.9 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'argument 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.

Si `arg_num` est plus grand que le nombre d'arguments passés, une alerte est générée et la fonction retourne `FALSE`.

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.

Note

`func_get_arg` a été ajoutée en PHP 4.

7.34.10 `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 les éléments correspondent aux éléments de la liste d'arguments de la fonction. `func_get_args` générera une alerte si elle est appelée hors d'une fonction.

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.

Note

`func_get_arg` a été ajoutée en PHP 4.

7.34.11 `func_num_args()` : Retourne le nombre d'arguments passé à la fonction

int `func_num_args` (void)

`func_num_args` retourne le nombre d'arguments passé à la fonction utilisateur courante. `func_num_args` générera une alerte si elle est appelée hors d'une fonction.

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.

Note

`func_get_arg` a été ajoutée en PHP 4.

7.34.12 `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.

Voir aussi `method_exists` et `get_defined_functions`.

7.34.13 `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);
?>
```

Ce script 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 [get_defined_vars](#) et [get_defined_constants](#) .

7.34.14 register_shutdown_function() : Enregistre une fonction pour exécution à l'extinction

```
void register_shutdown_function ( callback function )
```

[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 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.

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](#) .

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` .

Voir aussi [auto_append_file](#) , [exit](#) , et la section sur la [gestion des connexions](#) .

7.34.15 register_tick_function() : Enregistre une fonction exécutée à chaque tick

```
void register_tick_function ( callback function , mixed arg )
```

[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);
?>
```

Voir aussi [declare](#) et [unregister_tick_function](#) .

7.34.16 `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 .

7.35 Gettext (GNU)

7.35.1 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/manual/gettext/index.html> , ou bien sur votre système pour plus d'explications sur ces fonctions.

7.35.2 Pré-requis

To use these functions you must download and install the GNU gettext package from <http://www.gnu.org/software/gettext/gettext.html>

7.35.3 Installation

Pour inclure le module GNU gettext dans ^{PHP} , vous le compiler avec l'option `--with-gettext[=DIR]` , où DIR est le dossier d'installation de gettext. Par défaut, c'est `/usr/local` .

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier les librairies **gnu_gettext.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). Depuis PHP 4.2.3, le nom de la librairie a été changé en **libintl-1.dll**

7.35.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.35.5 Types de ressources

Cette extension ne définit aucune ressource.

7.35.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [bind_textdomain_codeset](#)
- [bindtextdomain](#)

- [dcgettext](#)
- [dcngettext](#)
- [dgettext](#)
- [dngettext](#)
- [gettext](#)
- [ngettext](#)
- [textdomain](#)

7.35.8 bindtextdomain() : Fixe le chemin d'un domaine

```
string bindtextdomain ( string domain , string directory )
```

[bindtextdomain](#) fixe le chemin du domaine `domain` à `directory` .

7.35.9 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.

7.35.10 dcngettext() : Version plurielle de dcgettext

```
string dcngettext ( string domain , string msgid1 , string msgid2 , int n , int category )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.35.11 dgettext() : Remplace le domaine courant

```
string dgettext ( string domain , string message )
```

[dgettext](#) remplace le domaine courant.

7.35.12 dngettext() : Version plurielle de dgettext

```
string dngettext ( string domain , string msgid1 , string msgid2 , int n )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.35.13 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
putenv("LANG=de");
// Spécifie la localisation des tables de traduction
bindtextdomain("myPHPApp", "./locale");
// Choisit le domaine
textdomain("myPHPApp");
// Affiche un message de test
print (gettext ("Bienvenue sur mon application PHP"));
?>
```

7.35.14 `ngettext()` : Version plurielle de `gettext`

```
string ngettext ( string msgid1 , string msgid2 , int n )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.35.15 `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. Le domaine par défaut précédent est retourné. Appelez cette fonction avec `NULL` comme paramètre pour avoir la valeur courante, sans la modifier.

7.36 GMP

7.36.1 Introduction

Ces fonctions vous permettent de travailler avec des nombres de taille arbitraire, en utilisant la librairie GNU ^{GMP} .

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 ^{resource} . 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 faite pas, PHP va interpréter votre entier, et le transformer en une représentation interne, qui vous fera surement perdre de la précision.

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.36.2 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} version 2 ou plus récent pour utiliser ces fonctions. Certaines d'entre elles peuvent requérir une version encore plus récente de ^{GMP} .

7.36.3 Installation

Pour pouvoir y accéder, vous devez compiler PHP avec le support ^{GMP} en utilisant l'option `--with-gmp` .

7.36.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.36.5 Types de ressources

Cette extension ne définit aucune ressource.

7.36.6 Constantes prédefinies

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*)

7.36.7 Exemples

Factorielle avec GMP
<pre><?php function fact(\$x) { if(\$x <= 1) return 1; else return gmp_mul(\$x,fact(\$x-1)); } print gmp_strval(fact(1000))."\n"; ?></pre>

Cet exemple va calculer factorielle de 1000 (un plutôt grand nombre) très vite.

7.36.8 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 prédefinies](#)
- [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_sqrtrm](#)
- [gmp_strval](#)
- [gmp_sub](#)
- [gmp_xor](#)

7.36.10 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.

7.36.11 gmp_and() : ET logique

resource **gmp_and** (resource a , resource b)

[gmp_and](#) calcule le ET logique de ^a et ^b .

7.36.12 gmp_clrbit() : Annule un bit

resource **gmp_clrbit** (resource &a , int index)

[gmp_clrbit](#) met le bit ^{index} à 0 dans le nombre GMP ^a .

7.36.13 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 .

7.36.14 gmp_com() : Calcule le complémentaire d'un nombre

resource **gmp_com** (resource a)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.36.15 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.

Voir aussi gmp_div_r et gmp_div_qr.

7.36.16 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, "0xDEFE75");
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.

7.36.17 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.

Voir gmp_div_q pour une description du paramètre ^{round}.

Voir aussi gmp_div_q et gmp_div_qr.

7.36.18 gmp_div() : Divise deux nombres GMP

resource **gmp_div** (resource *a* , resource *b*)

gmp_div est un alias de gmp_div_q .

7.36.19 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 .

7.36.20 gmp_fact() : Factorielle GMP

resource **gmp_fact** (int *a*)

gmp_fact calcule la factorielle de a : $a!$.

7.36.21 gmp_gcd() : PGCD

resource **gmp_gcd** (resource *a* , resource *b*)

gmp_gcd calcule de 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.

7.36.22 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 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
```

7.36.23 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.

7.36.24 gmp_init() : Crée un nombre GMP

resource **gmp_init** (mixed number)

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.

Création d'un nombre GMP

```
<?php
$a = gmp_init (123456);
$b = gmp_init ("0xFFFFFDEBACDFEDF7200");
?>
```

Attention

Si vous devez explicitement spécifier un entier de grande taille, faites le avec une chaîne. Sinon, PHP va interpréter l'entier littéralement, et vous y perdrez en précision avant que les fonctions ^{GMP} n'entre en jeu.

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 par exemple 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.

7.36.25 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.

7.36.26 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.

7.36.27 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.

7.36.28 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.

7.36.29 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.

7.36.30 gmp_mul() : Multiplication de 2 nombres GMP

resource **gmp_mul** (resource a , resource b)

gmp_mul multiplie les nombres GMP a et b . Le résultat est un nombre GMP.

7.36.31 gmp_neg() : Opposé de nombre GMP

resource **gmp_neg** (resource a)

gmp_neg retourne l'opposé de a ($-1 * a$).

7.36.32 gmp_or() : OU logique

resource **gmp_or** (resource a , resource b)

gmp_or calcule le OU logique de a et b .

7.36.33 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.

Voir aussi gmp_sqrt et gmp_sqrtrem.

7.36.34 gmp_popcount() : Compte de population

```
int gmp_popcount ( resource a )
```

gmp_popcount dénombre la population de ^a.

7.36.35 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.

7.36.36 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.

7.36.37 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.

7.36.38 gmp_random() : Nombre GMP aléatoire

```
resource gmp_random ( int limiter )
```

gmp_random génère un nombre aléatoire. Ce nombre est limité par ^{limiter}. Si ^{limiter} est négatif, un nombre négatif est généré.

7.36.39 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.

7.36.40 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.

7.36.41 gmp_setbit() : Modifie un bit

resource **gmp_setbit** (resource &a , int index , bool set_clear)

gmp_setbit modifie le bit ^{index} dans ^a. ^{set_clear} indique si le bit est mis à 0 ou 1. Par défaut, il est mis à 1.

7.36.42 gmp_sign() : Signe du nombre GMP

int **gmp_sign** (resource a)

gmp_sign retourne le signe de ^a : 1 si ^a est positif et -1 s'il est négatif.

7.36.43 gmp_sqrt() : Racine carrée GMP

resource **gmp_sqrt** (resource a)

gmp_sqrt calcule la racine carrée de ^a.

7.36.44 gmp_sqrtrm() : Racine carrée avec reste GMP

array **gmp_sqrtrm** (resource a)

gmp_sqrtrm 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).

7.36.45 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));
?>
```

7.36.46 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.

7.36.47 gmp_xor() : OU exclusif logique

resource **gmp_xor** (resource a , resource b)

gmp_or calcule le OU exclusif logique de ^a et ^b .

7.37 HTTP

7.37.1 Introduction

Ces fonctions permettent de travailler sur les informations transmises au navigateur, via le protocole HTTP.

7.37.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.37.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.37.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.37.5 Types de ressources

Cette extension ne définit aucune ressource.

7.37.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [header](#)
- [headers_sent](#)
- [setcookie](#)

7.37.8 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 reliées. Une autre option est d'utiliser le contrôle de sortie.

Note

Nouveaux paramètres

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.nexen.net/');
    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.nexen.net/');
    exit;
} else {
    print "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.nexen.net/\">ici</a>.\n";
    exit;
}
?>
```

Voir aussi `ob_start`, `trigger_error` et `header` pour plus de détails sur les tenants et aboutissants.

7.37.9 setcookie() : Envoie un cookie

```
bool setcookie ( string name , string value , int expire , string path , string domain , int secure )
```

`setcookie` définit un cookie qui sera envoyé avec le reste des en-têtes. Les cookies doivent passer avant toute 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>`.

Note

En 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. Le paramètre `expire` est un timestamp UNIX, du même genre que celui retourné par `time` ou `mktime`. Le paramètre `secure` indique que le cookie doit être uniquement transmis à travers une connexion HTTPS sécurisée.

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`.

Erreurs communes :

- Les cookies ne seront accessibles qu'au chargement de la prochaine page, ou au rechargement de la page courante.
- Les cookies doivent être effacés avec les mêmes paramètres que ceux utilisés lors de leur création.

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. En PHP 4, les appels multiples à `setcookie` sont effectués dans un ordre naturel.

Les appels multiples à `setcookie` dans la même page seront réalisés dans l'ordre inverse. Si vous essayez d'effacer un cookie avant d'insérer une autre valeur, il faut placer l'insertion avant l'effacement.

Quelques exemples :

Exemples avec `setcookie`

```
<?php
setcookie("TestCookie","Valeur de test");
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. 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 la date d'expiration du cookie 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/", ".exemple.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 PHP, les valeurs seront placées dans un tableau :

Utilisation des tableaux avec `setcookie`

```
<?php
setcookie( "cookie[three]", "cookiethree" );
setcookie( "cookie[two]", "cookietwo" );
setcookie( "cookie[one]", "cookieone" );
if ( isset( $cookie ) ) {
```

```
while( list( $name, $value ) = each( $cookie ) ) {  
    echo "$name == $value<br>\n";  
}  
?>
```

Note

Pour plus d'informations sur les cookies, voyez les spécifications de cookies de Netscape à http://www.netscape.com/newsref/std/cookie_spec.html et [RFC 2965](#) .

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.

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 `time>` ne sont pas fournis.

Voir aussi [header](#) .

7.38 Hyperwave

7.38.1 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, si mes souvenirs sont bons).

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 paire 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 géré 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 hyper liens contenus dans un documents 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 hyper-liens 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 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 n'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 en 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 de 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 par exemple : `'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 par exemple celles qui commencent par `http://host/Hyperwave`. Voici, de manière simple, la manière de partager un espace de nom entre un serveur web et un serveur Hyperwave.

Basé sur le mécanisme précédent, on trouve l'insertion dans les documents.

Il est plus compliqué d'avoir PHP ne fonctionne 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 initie des actions, comme par exemple, 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* language. *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 fournit d'autres commandes puissantes, comme par exemple, 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 utilisées 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 object records. 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 par exemple 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'objet dont l'attribut PresentationHints est CollectionHead.

FullCollectionHead

Nombre d'objet 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.

7.38.2 Pré-requis

Cette extension requiert un serveur Hyperwave, téléchargeable sur le site de <http://www.hyperwave.com/>.

7.38.3 Installation

Pour activer le support Hyperwave, compilez PHP avec l'option `--with-hyperwave`.

7.38.4 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 alors 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
RewriteRule ^/(.*) /usr/local/apache/htdocs/HyperWave/\$1 [L]

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)
RewriteRule ^/hw/(.*) /usr/local/apache/htdocs/hw/\$1 [L]

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 définie. 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

Démarrer les règles de réécriture d'Apache pour HyperWave
RewriteEngine on

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.

7.38.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>hyperwave.allow_persistent</code>	"0"	PHP_INI_SYSTEM
<code>hyperwave.default_port</code>	"418"	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

7.38.6 Types de ressources

Cette extension ne définit aucune ressource.

7.38.7 Constantes prédéfinies

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*)

7.38.8 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 prédefinies](#)
- [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_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_username](#)
- [hw_identify](#)
- [hw_incollections](#)
- [hw_info](#)
- [hw_inscoll](#)
- [hw_insdock](#)
- [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](#)

7.38.10 hw_changeobject() : Change les attributs d'un objet (obsolète)

```
void hw_changeobject ( int link , int objid , array attributes )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.38.11 hw_children() : Liste des object ids des objets fils

```
array hw_children ( int connection , int objectid )
```

[hw_children](#) retourne un tableau avec des object ids. Chaque object id est celui d'un des fils du groupe dont l'id est `objectid`. Ce tableau contient tous les fils, documents et groupes.

7.38.12 hw_childrenobj() : Liste des object records des objets fils

```
array hw_childrenobj ( int connection , int objectid )
```

hw_childrenobj retourne un tableau avec des object records. Chaque object records est celui d'un des fils du groupe dont l'id est `objectid`. Ce tableau contient tous les fils, documents et groupes

7.38.13 hw_close() : Ferme la connexion Hyperwave

```
int 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.

7.38.14 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.

7.38.15 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.

7.38.16 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'object id `destination_id`.

La valeur retournée est le nombre d'objets copiés.

Voir aussi hw_mv.

7.38.17 hw_deleteobject() : Efface des objets

`int 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, et sinon, `FALSE` .

Voir aussi hw_mv .

7.38.18 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` .

7.38.19 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` .

7.38.20 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 ascendente, hw_document_attributes est aussi accepté, mais il est déconseillé.

7.38.21 hw_document_bodytag() : Balise de corps d'un document

`string hw_document_bodytag ( int hw_document )`

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 ascendente, hw_document_bodytag est aussi accepté, mais il est déconseillé.

7.38.22 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.

Pour la compatibilité ascendente, hw_document_content est aussi accepté, mais il est déconseillé.

7.38.23 hw_document_setcontent() : Modifie/remplace le contenu d'un document

```
string 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 qui est placé au-delà de la balise BODY. Les informations de HEAD et de la balise BODY sont enregistrés 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.

Pour la compatibilité ascendente, hw_documentsetcontent est aussi accepté, mais il est déconseillé.

7.38.24 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 ascendente, hw_documentsize est aussi accepté, mais il est déconseillé.

7.38.25 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.

7.38.26 hw_edittext() : Retourne un document texte

```
int hw_edittext ( 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](#).

7.38.27 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.

7.38.28 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.

7.38.29 hw_free_document() : Détruit un document

```
int hw_free_document ( int hw_document )
```

[hw_free_document](#) détruit un document Hyperwave.

7.38.30 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`.

7.38.31 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`.

7.38.32 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](#).

7.38.33 hw_getchildcoll() : Identifiant d'objet 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.

7.38.34 hw_getchildcollobj() : object records d'un groupe d'enfants

```
array hw_getchildcollobj ( int connection , int objectid )
```

hw_getchildcollobj retourne un tableau d'object record. Chaque object records 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.

7.38.35 hw_getchilddoccoll() : ids 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.

7.38.36 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.

7.38.37 hw_getobject() : Lit les attributs d'un objet Hyperwave

```
array 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 le 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 régulière */

`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 retourné dépend de la requête de la requête, et des autorisations d'accès aux objets.

Voir aussi [hw_getandlock](#) et [hw_getobjectbyquery](#).

7.38.38 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'objet 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](#).

7.38.39 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](#).

7.38.40 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'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_getobjectbyquerycoll](#).

7.38.41 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'objet 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](#).

7.38.42 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`.

7.38.43 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`.

7.38.44 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.

7.38.45 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](#).

7.38.46 hw_getremotechildren() : Retourne les fils d'un document distant

```
int 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 rendu plus sélective, comme expliqué dans Hyperwave Programmers' Guide. Si le nombre de fils est 1 la fonction va retourner le document lui-même, la fonction retournera le document lui-même, formaté Hyperwave Gateway Interface (HGI). Si le nombre de fils est supérieur 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 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](#).

7.38.47 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](#).

7.38.48 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 seront déjà insérés. 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 par exemple '/mon_script.php3/film_internet'. 'mon_script.php3' devra analyser \$PATH_INFO et savoir recherche 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-internet_movie' 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 serait: `<A HREF="..c/internet_movie">`. 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`.

7.38.49 `hw_username()` : Nom de l'utilisateur actuellement identifié

```
string hw_username ( int connection )
```

`hw_username` retourne le nom d'utilisateur utilisé par la connexion.

7.38.50 `hw_identify()` : Identifie un utilisateur

```
int hw_identify ( 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`.

7.38.51 `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 groupe 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.

7.38.52 `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>`

7.38.53 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`.

7.38.54 hw_insdoc() : Insère un document

```
int hw_insdoc ( int connection , int parentid , string object_record , string text )
```

[hw_insdoc](#) insère un nouveau document avec les attributs `object_record`, dans le groupe `parentid`. Cette fonction insère soit un objet avec ses seuls attributs, soit pur 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](#).

7.38.55 hw_insertanchors() : Insère uniquement des ancrs dans du texte

```
string 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.

7.38.56 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 attributs suivants : 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](#).

7.38.57 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 un ancre, l'attribut Position doit être mis à la valeur start/end (début ou fin) ou encore 'invisible'. Les positions invisibles sont nécessaire 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](#).

7.38.58 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 dans le fichier hg_comm.c

7.38.59 hw_modifyobject() : Modifie les attributs d'objet record

```
int 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 la liste 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, ou 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. Dans le cas où 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 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 st 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 manières : 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 deviendra 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' gère 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.

7.38.60 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](#).

7.38.61 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](#).

7.38.62 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 par exemple 'Title', dans différentes langues, seront rassemblées 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' seront traités 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](#).

7.38.63 hw_output_document() : Affiche hw_document

```
int 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.

7.38.64 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 authentification ne sera faite, (connexion anonyme). Cette fonction retourne un index de connexion qui sera utilisée par les autres fonctions Hyperwave. Vous pouvez ouvrir plusieurs connexions simultanées.

Voir aussi [hw_connect](#).

7.38.65 `hw_pipedocument()` : Retourne un document

```
int hw_pipedocument ( int connection , int objectid )
```

[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](#).

7.38.66 `hw_root()` : Object id de la racine

```
int hw_root ( )
```

[hw_root](#) retourne l'object id de la racine. Actuellement, cette identifiant est toujours 0. L'ensemble des fils de la racine est celui du serveur courant.

7.38.67 `hw_setlinkroot()` : Configure l'identifiant vers lequel les liens sont calculés

```
void hw_setlinkroot ( int link , int rootid )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.38.68 `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.

7.38.69 `hw_unlock()` : Déverrouille un objet

```
int hw_unlock ( int connection , int objectid )
```

[hw_unlock](#) déverrouille un document, et laisse l'accès aux autres utilisateurs.

Voir aussi [hw_getandlock](#).

7.38.70 hw_who() : Liste des utilisateurs actuellement identifiés

`int 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' contient l'élément est l'utilisateur qui a appelé la fonction.

7.39 Hyperwave API

7.39.1 Introduction

Hyperwave a été développé à l'IICM à Graz. Le projet a commencé avec le nom de ^{HYP}ER-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 métas 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.

7.39.2 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.

7.39.3 Installation

Après avoir installé le SDK Hyperwave, compilez PHP avec l'option `--with-hwapi[=DIR]`.

7.39.4 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.

7.39.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>hwapi.allow_persistent</code>	<code>"0"</code>	<code>PHP_INI_SYSTEM</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

7.39.6 Types de ressources

Cette extension ne définit aucune ressource.

7.39.7 Constantes prédefinies

Cette extension ne définit aucune constante.

7.39.8 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`.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [intégration dans Apache](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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->dstofsrcanchors](#)
- [hw_api_error->count](#)
- [hw_api_error->reason](#)
- [hw_api->find](#)
- [hw_api->ftstat](#)
- [hwapi_hgcsp](#)
- [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](#)

7.39.10 `hw_api_attribute->langdepvalue()` : Retourne la valeur d'un attribut dans une langue

`string langdepvalue ( string language )`

`hw_api_attribute->langdepvalue` retourne la valeur de l'attribut dans la langue `language`.

Voir aussi [hwapi_attribute_value](#).

7.39.11 `hw_api_attribute->value()` : Retourne la valeur de l'attribut HW

`string value ( void )`

`hw_api_attribute->value` retourne la valeur de l'attribut.

Voir aussi [hwapi_attribute_key](#) et [hwapi_attribute_values](#).

7.39.12 `hw_api_attribute->values()` : Retourne toutes les valeurs de l'attribut sous la forme d'un tableau de chaînes

`array 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](#).

7.39.13 `hw_api_attribute()` : Crée une nouvelle instance de la classe `hw_api_attribute`

`object 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`.

7.39.14 `hw_api->checkin()` : Archive un objet

`object 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 version majeure et mineure, séparée 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 vérifié récursivement, si ils sont des documents. Tenter d'archiver une collection générera une erreur.

`HW_API_CHECKIN_FORCE_VERSION_CONTROL`

Archive un objet même si il ne passe pas sous le contrôle de versions.

`HW_API_CHECKIN_REVERT_IF_NOT_CHANGED`

Vérifie si 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 du plus récent objet.

`HW_API_CHECKIN_NO_AUTO_COMMIT`

L'objet n'est pas automatiquement validé lors de l'archivage.

Voir aussi [hwapi_checkout](#) .

7.39.15 hw_api->checkout() : Extrait un objet

object **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](#) .

7.39.16 hw_api->children() : Retourne le fils d'un objet

array **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](#) .

7.39.17 hw_api_content->mimetype() : Retourne le type MIME

string **mimetype** (void)

hw_api_content->mimetype retourne le type MIME du contenu.

7.39.18 hw_api_content->read() : Lit le contenu

string **read** (string buffer , integer len)

[hw_api_content->read](#) lit `len` octets dans le buffer `buffer` .

7.39.19 `hw_api->content()` : Retourne le contenu d'un objet

object **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 liens.

7.39.20 `hw_api->copy()` : Copies physically

object **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](#).

7.39.21 `hw_api->dbstat()` : Retourne des statistiques sur le serveur de bases

object **dbstat** (array parameter)

`dbstat` retourne des statistiques sur le serveur de bases.

Voir aussi [hwapi_dcstat](#), [hwapi_hwstat](#) et [hwapi_ftstat](#).

7.39.22 `hw_api->dcstat()` : Retourne des statistiques sur le serveur de cache de document

object **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](#).

7.39.23 `hw_api->dstanchors()` : Liste toutes les ancrs de destination

object **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](#).

7.39.24 `hw_api->dstofsrcanchors()` : Retourne la destination d'une ancre source

object **dstofsrcanchors** (array parameter)

[`hw\_api->dstofsrcanchors`](#) 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\_objectbvanchor`](#).

7.39.25 `hw_api_error->count()` : Retourne le nombre de raisons d'erreurs

int **count** (void)

[`hw\_api\_error->count`](#) retourne le nombre de raisons d'erreurs.

Voir aussi [`hwapi\_error\_reason`](#).

7.39.26 `hw_api_error->reason()` : Retourne la première raison d'erreur

object **reason** (void)

[`hw\_api\_error->reason`](#) retourne la première raison d'erreur.

Voir aussi [`hwapi\_error\_count`](#).

7.39.27 `hw_api->find()` : Recherche des objets

array **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', 'lanugages' et 'attributeselector'. En cas de recherche incrémentale, les éléments optionnels 'startIndex', 'numberOfObjectsToGet' et 'exactMatchUnit' peuvent aussi être passés.

7.39.28 `hw_api->ftstat()` : Retourne des statistiques sur le serveur de textes

object **ftstat** (array parameter)

[`ftstat`](#) retourne des statistiques sur le serveur de textes.

Voir aussi [hwapi_dcstat](#) , [hwapi_dbstat](#) et [hwapi_hwstat](#) .

7.39.29 hwapi_hgcsp() : Retourne un objet de classe hw_api

object **hwapi_hgcsp** (string hostname , int port)

[hwapi_hgcsp](#) ouvre une connexion au serveur Hyperwave sur l'hôte `hostname` . Le protocole utilisé est HGCS. Si vous l'omettez, le port 418 sera utilisé.

Voir aussi [hwapi_hwtg](#) .

7.39.30 hw_api->hwstat() : Retourne les statistiques sur le serveur Hyperwave

object **hwstat** (array parameter)

[hwstat](#) retourne les statistiques sur le serveur Hyperwave.

Voir aussi [hwapi_dcstat](#) , [hwapi_dbstat](#) et [hwapi_ftstat](#) .

7.39.31 hw_api->identify() : S'identifie auprès du serveur Hyperwave

object **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.

7.39.32 hw_api->info() : Retourne des information sur la configuration du serveur

object **info** (array parameter)

[hw_api->info](#) retourne des information sur la configuration du serveur.

Voir aussi [hwapi_dcstat](#) , [hwapi_dbstat](#) , [hwapi_ftstat](#) et [hwapi_hwstat](#) .

7.39.33 hw_api->insert() : Insère un nouvel objet

object **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 élément 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, qu 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](#) .

7.39.34 hw_api->insertanchor() : Insère un nouvel objet de type ancre

```
object insertanchor ( array parameter )
```

[hw_api->insertanchor](#) est un raccourci pour [hwapi_insert](#) . Elle insère un objet de type ancre ('anchor') et assigne les attribus 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' si 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](#) .

7.39.35 hw_api->insertcollection() : Insère un nouvel objet de type collection

```
object 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 'parentIdentifier', et les éléments optionnels 'parameter' et 'attributeSelector'. Voyez [hwapi_insert](#) pour connaître le détails de chaque élément.

Voir aussi [hwapi_insertdocument](#) , [hwapi_insertanchor](#) et [hwapi_insert](#) .

7.39.36 hw_api->insertdocument() : Insère un nouvel objet de type document

```
object 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', 'parentIdentifier' 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](#) .

7.39.37 hw_api->link() : Crée un lien avec un objet

object **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 'objectIdentifier' et 'destinationParentIdentifier'. 'destinationParentIdentifier' est la collection de destination.

hw_api->link retourne ^{TRUE} en cas de succès, et sinon, un objet d'erreur.

Voir aussi hwapi_copy .

7.39.38 hw_api->lock() : Verrouille un objet

object **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 'objectIdentifier' et les paramètres optionnels 'mode' et 'objectquery'. 'mode' détermine comment l'objet est verrouillé. ^{HW_API_LOCK_NORMAL} signifie que le '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 .

7.39.39 hw_api->move() : Déplace un objet d'une collection à l'autre

object **move** (array parameter)

hw_api->move déplace un objet d'une collection à l'autre.

Voir aussi hw_obirec2array .

7.39.40 hw_api->content() : Crée une nouvelle instance de la classe hw_api_content

string **content** (string content , string mimetype)

hw_api-> crée un nouvel objet de contenu à partir de la chaîne ^{content} . Le type MIME utilisé est ^{mimetype} .

7.39.41 hw_api_object->assign() : Clone un objet

object **assign** (array parameter)

hw_api_object->assign clone les attributs d'un objet.

7.39.42 `hw_api_object->attreditable()` : Vérifie si un attribut est éditable

`bool attreditable ( array parameter )`

`hw_api_object->attreditable` vérifie si un attribut est éditable.

7.39.43 `hw_api_object->count()` : Retourne le nombre d'attributs

`int count ( array parameter )`

`hw_api_object->count` retourne le nombre d'attributs.

7.39.44 `hw_api_object->insert()` : Insère un nouvel attribut HyperWave

`bool insert ( object 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`.

7.39.45 `hw_api_object()` : Crée une nouvelle instance de la classe `hw_api_object`

`object hw_api_object ( array parameter )`

`hw_api_object` crée une nouvelle instance de la classe `hw_api_object`.

Voir aussi `hwapi_lock`.

7.39.46 `hw_api_object->remove()` : Supprime un attribut HyperWave

`bool 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`.

7.39.47 `hw_api_object->title()` : Retourne le titre de l'attribut

`string title ( array parameter )`

7.39.48 `hw_api_object->value()` : Retourne la valeur de l'attribut

`string value ( string name )`

`hw_api_object->value` retourne la valeur de l'attribut `name`, ou bien `FALSE` si une erreur est survenue.

7.39.49 hw_api->object() : Lit les informations d'attribut

object **hw_api->object** (array parameter)

`hw_api->object` lit les informations d'attributs 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 = $error->reason(0);
    echo "Type: <B>";
    switch($reason->type()) {
        case 0:
            echo "Erreur";
            break;
        case 1:
            echo "Alerte";
            break;
        case 2:
            echo "Message";
            break;
    }
    echo "</B><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 ALIGN=right bgcolor=#c0c0c0><B>%s</B></TD><TD bgcolor=#F0F0F0>%s</TD>\n",
            $attr->key(), $attr->value());
    }
    echo "</TABLE>\n";
}

$hwapi = hwapi_hgcsp($g_config[HOSTNAME]);
$params = array("objectIdentifier"=>"rootcollection", "attributeSelector"=array("Title", "Name", "DocumentType"));
$root = $hwapi->object($params);
if(get_class($root) == "HW_API_Error") {
    handle_error($root);
    exit;
}
list_attr($root);
?>
```

Voir aussi [hwapi_content](#) .

7.39.50 hw_api->objectbyanchor() : Retourne l'objet propriétaire d'une ancre

object **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_dstofsrcanchors](#) , [hwapi_srcanchors](#) et [hwapi_dstanchors](#) .

7.39.51 hw_api->parents() : Retourne le parent d'un objet

array **parents** (array parameter)

[hw_api->parents](#) lit le parent d'un objet. Le parent peut être filtrés en spécifiant un objet de requête. Le paramètre `parameter` est un tableau qui contient l'élément obligatoire 'objectidentifiant' 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](#) .

7.39.52 hw_api_reason->description() : Retourne la description d'une raison

string **description** (void)

[hw_api_reason->description](#) retourne la description d'une raison.

7.39.53 hw_api_reason->type() : Retourne le type de raison

object **type** (void)

[hw_api_reason->type](#) retourne le type de raison.

7.39.54 hw_api->remove() : Efface un objet

object **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 filtrés. 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 comme 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](#) .

7.39.55 hw_api->replace() : Remplace un objet

object **replace** (array parameter)

[hw_api->replace](#) remplace les attributs et le contenu d'un bjet. Le paramètre `parameter` est un tableau qui contient les éléments obligatoires 'objectIdentifier' et 'object' et les paramètres optionnels 'content', 'parameters', 'mode' et 'attributeSelector'. 'objectIdentifier' 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](#) .

7.39.56 hw_api->setcommittedversion() : Valide une version autre que la dernière

```
object setcommittedversion ( array parameter )
```

[hw_api->setcommittedversion](#) valide une version du document. La version validée est celle qui est visible aux 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](#) .

7.39.57 hw_api->srcanchors() : Retourne la liste de toutes les ancrs

```
object 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 'objectIdentifier' et les éléments optionnels 'attributeSelector' wr 'objectQuery'.

Voir aussi [hwapi_dstanchors](#) .

7.39.58 hw_api->srcsofdst() : Retourne les sources d'un objet de destination

```
object 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 'objectIdentifier' et les éléments optionnels 'attributeSelector' et 'objectQuery'. La fonction retourne un tableau d'objet ou une erreur.

Voir aussi [hwapi_dstofsrcanchor](#) .

7.39.59 hw_api->unlock() : Déverrouille un objet HyperWave verrouillé

object **unlock** (array parameter)

hw_api->unlock déverrouille un objet verrouillé. Seul, 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 ^{objectIdentifier} 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.

7.39.60 hw_api->user() : Retourne le propriétaire d'un objet

object **user** (array parameter)

hw_api->user retourne le propriétaire d'un objet.

Voir aussi hwapi_userlist.

7.39.61 hw_api->userlist() : Retourne la liste des utilisateurs connectés

object **userlist** (array parameter)

hw_api->userlist retourne la liste des utilisateurs connectés.

Voir aussi hwapi_user.

7.40 Iconv

7.40.1 Introduction

Ce module est une interface vers la librairie 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, vous devez installer la librairie tout de même.

7.40.2 Pré-requis

Pour cela, vous devez avoir la fonction iconv() dans votre librairie standard C, ou bien la librairie libiconv installée sur votre système. La librairie libiconv est disponible à <http://www.gnu.org/software/libiconv/> .

7.40.3 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 les librairies iconv-1.3.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). Depuis PHP 4.2.1, le nom de cette librairie a été changée en iconv.dll .

7.40.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
iconv.input_encoding	ICONV_INPUT_ENCODING	PHP_INI_ALL
iconv.output_encoding	ICONV_OUTPUT_ENCODING	PHP_INI_ALL
iconv.internal_encoding	ICONV_INTERNAL_ENCODING	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [ini_set](#) .

7.40.5 Types de ressources

Cette extension ne définit aucune ressource.

7.40.6 Constantes prédefinies

Depuis PHP 4.3.0, il est possible d'identifier durant l'exécution du script l'implémentation de iconv qui est utilisée par l'extension.

constante	type	description

ICONV_IMPL	string	Le nom de l'implémentation
ICONV_VERSION	string	La version de l'implémentation

Note
Ecrire des scripts dépendants de la version de iconv est fortement découragé.

7.40.7 Voir aussi

Voir aussi les fonctions [GNU Recode](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Voir aussi](#)
- [iconv_get_encoding](#)
- [iconv_set_encoding](#)
- [iconv](#)
- [ob_iconv_handler](#)

7.40.9 iconv_set_encoding() : Modifie le jeu courant de caractères courant

`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
<pre><?php iconv_set_encoding("internal_encoding", "UTF-8"); iconv_set_encoding("output_encoding", "ISO-8859-1"); ?></pre>

Voir aussi [iconv_get_encoding](#) et [ob_iconv_handler](#) .

7.40.10 iconv() : Convertit une chaîne dans un jeu de caractères

`string iconv ( string in_charset , string out_charset , string str )`

[iconv](#) convertit la chaîne ^{string} 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.

Exemple avec `iconv`

```
<?php
echo iconv("ISO-8859-1", "UTF-8", "Ceci est un test.");
?>
```

7.40.11 `ob_iconv_handler()` : Gestionnaire de sortie pour maîtriser le jeu de caractères de sortie

array **`ob_iconv_handler`** (string contents , int status)

`ob_iconv_handler` convertit la chaîne utilisant le jeu de caractères `internal_encoding` en une chaîne utilisant le jeu de caractères `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](#) .

7.41 Images

7.41.1 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, jpg, wbmp et xpm. Et même, PHP peut générer directement des images pour le navigateur, avec la librairie GD. GD et PHP auront aussi besoin d'autres librairies, 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.

7.41.2 Pré-requis

Si vous disposez de la librairie ^{GD} (disponible à <http://www.boutell.com/gd/>) vous pourrez aussi créer et manipuler ces images.

Les formats des images que vous pourrez manipuler dépendent de la version de GD que vous installerez, et de toute autre librairie dont GD a besoin pour traiter ces images. Les versions antérieures à la version 1.6 supportent le ^{GIF}, mais pas le ^{PNG}. Pour les versions plus récentes, c'est le contraire.

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 librairie 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	Librairie à télécharger	Notes
gif		Uniquement supporté en versions de GD antérieures à gd-1.6. Le support des images GIF est disponible en lecture seule depuis PHP 4.3.0, et dans la version de la librairie qui est fournie avec la distribution de PHP.
jpeg-6b	ftp://ftp.uu.net/graphics/jpeg/	
png	http://www.libpng.org/pub/png/libpng.html	Uniquement supporté avec GD plus récente que gd-1.6.
xpm	ftp://metalab.unc.edu/pub/Linux/libs/X/INDEX.html	Il est probable que vous ayez déjà cette librairie disponible si votre système dispose d'un environnement X.

Vous pouvez aussi améliorer GD en lui ajoutant des fonctionnalités de manipulation de polices. Les librairies suivantes sont supportées :

Librairie de police	Téléchargement	Notes
FreeType 1.x	http://www.freetype.org/	
FreeType 2	http://www.freetype.org/	
Ttlib	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 entêtes des images ^{JPEG} et ^{TIFF}. De cette façon, vous pouvez lire les métadonnées générées par les appareils numériques. Ces fonctions ne requièrent par la

librairie ^{GD} .

Note
PHP ne requiert aucune librairie supplémentaire pour gérer les données EXIF.

7.41.3 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 distribué avec PHP, en utilisant simplement l'option `--with-gd` . Sous Windows, vous devez inclure la librairie `php_gd2.dll` comme extension dans le fichier `php.ini` . Il y a aussi la librairie `php_gd.dll` pour GD version 1.x mais elle n'est pas recommandée.

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 de compilations suivantes :

Format d'image	Option de compilation
jpeg-6b	Pour activer le support de la librairie JPEG-6b, ajoutez l'option <code>--with-jpeg-dir=DIR</code> .
png	Pour activer le support de la librairie PNG, ajoutez l'option <code>--with-png-dir=DIR</code> . Notez que libpng requiert la librairie <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 librairie XPM, ajoutez l'option <code>--with-xpm-dir=DIR</code> . Si le script de compilation n'est pas capable de trouver les librairies nécessaires, il vous faudra ajouter le chemin vers les librairies X11.

Augmentez les possibilités de GD pour qu'elle manipule différents types de polices de caractères en ajoutant les options de compilations suivantes :

Librairie	Option de configuration
FreeType 1.x	Pour activer le support de FreeType 1.x, ajoutez l'option <code>--with-ttf[=DIR]</code> .
FreeType 2	Pour activer le support de FreeType 2, ajoutez l'option <code>--with-freetype-dir=DIR</code> .
Ttlib	Pour activer le support de Ttlib (Type 1 fonts), ajoutez l'option <code>--with-ttlib[=DIR]</code> .
Native TrueType string function	Pour activer le support des chaînes de caractères TrueType, ajoutez l'option <code>--enable-gd-native-ttf</code> .

7.41.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Exif supporte automatiquement la conversion en Unicode et JIS pour les commentaires des utilisateurs, lorsque le module `mbstring` est disponible. Cela se fait en décodant le commentaire avec le bon jeu de caractères. Le résultat peut alors être encodé dans le jeu de caractères de votre serveur HTTP.

Nom	Par défaut	Modifiable
<code>exif.encode_unicode</code>	"ISO-8859-15"	PHP_INI_ALL
<code>exif.decode_unicode_motorola</code>	"UCS-2BE"	PHP_INI_ALL
<code>exif.decode_unicode_intel</code>	"UCS-2LE"	PHP_INI_ALL
<code>exif.encode_jis</code>	" "	PHP_INI_ALL
<code>exif.decode_jis_motorola</code>	"JIS"	PHP_INI_ALL
<code>exif.decode_jis_intel</code>	"JIS"	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini set](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`exif.encode_unicode string`

`exif.encode_unicode` définit le jeu de caractères UNICODE utilisé pour les commentaires. Par défaut, c'est ^{ISO-8859-15}, qui devrait fonctionner pour les pays non asiatiques. Cette option peut être vide, ou bien contenir un jeu de caractères qui est supporté par `mbstring`. Si cette option est vide, la configuration de `mbstring` sera utilisée.

`exif.decode_unicode_motorola string`

`exif.decode_unicode_motorola` définit le jeu de caractères interne pour les utilisateurs Unicode, si le commentaire de l'image est en big-endian (ordre de motorola). Cette option ne peut être vide, mais vous pouvez y spécifier une liste de jeux de caractères utilisés par `mbstring`. Par défaut, c'est ^{UCS-2BE}.

`exif.decode_unicode_intel string`

`exif.decode_unicode_intel` définit le jeu de caractères interne pour les utilisateurs Unicode, si le commentaire de l'image est en little-endian (ordre d'intel). Cette option ne peut être vide, mais vous pouvez y spécifier une liste de jeux de caractères utilisés par `mbstring`. Par défaut, c'est ^{UCS-2LE}.

`exif.encode_jis string`

`exif.encode_jis` définit le jeux de caractères pour les commentaires en JIS. Par défaut, c'est une valeur vide, ce qui force la fonction à utiliser le jeu de caractères interne de `mbstring`.

`exif.decode_jis_motorola string`

`exif.decode_jis_motorola` définit le jeu de caractères interne pour les utilisateurs JIS, si le commentaire de l'image est en big-endian (ordre de motorola). Cette option ne peut être vide, mais vous pouvez y spécifier une liste de jeux de caractères utilisés par `mbstring`. Par défaut, c'est ^{JIS}.

`exif.decode_jis_intel string`

`exif.decode_jis_intel` définit le jeu de caractères interne pour les utilisateurs JIS, si le commentaire de l'image est en little-endian (ordre d'intel). Cette option ne peut être vide, mais vous pouvez y spécifier une liste de jeux de caractères utilisés par `mbstring`. Par défaut, c'est ^{JIS}.

7.41.5 Types de ressources

Cette extension ne définit aucune ressource.

7.41.6 Constantes prédefinies

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_JPG (*entier*)
 IMAGETYPE_PNG (*entier*)
 IMAGETYPE_SWF (*entier*)
 IMAGETYPE_PSD (*entier*)
 IMAGETYPE_BMP (*entier*)
 IMAGETYPE_TIFF_II (*entier*)
 IMAGETYPE_TIFF_MM (*entier*)
 IMAGETYPE_JPC (*entier*)
 IMAGETYPE_JP2 (*entier*)
 IMAGETYPE_JPX (*entier*)
 IMAGETYPE_SWC (*entier*)

7.41.7 Exemples

Cr  ation 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   tre appel   depuis une page HTML avec une balise image telle que : ``. Le script ci-dessus, `button.php`, prend la cha   ne "text" et l'inscrit sur le fond d'image appel   "images/button1.png" et l'affiche. C'est une m  thode tr  s pratique pour   viter de redessiner un nouveau bouton, d  s que vous changez le texte d'un bouton. De cette fa   on, ils sont g  n  r  s dynamiquement.

- [Introduction](#)
- [Pr  -requis](#)
- [Installation](#)
- [Configuration    l'ex  cution](#)
- [Types de ressources](#)
- [Constantes pr  d  finies](#)
- [Exemples](#)
- [exif_imagetype](#)
- [exif_read_data](#)
- [exif_thumbnail](#)
- [gd_info](#)
- [getimagesize](#)
- [image_type_to_mime_type](#)
- [image2wbmp](#)
- [imagealphablending](#)
- [imageantialias](#)
- [imagearc](#)
- [imagechar](#)
- [imagecharup](#)
- [imagecolorallocate](#)

- [imagecolorallocatealpha](#)
- [imagecolorat](#)
- [imagecolorclosest](#)
- [imagecolorclosestalpha](#)
- [imagecolorclosesthwb](#)
- [imagecolordeallocate](#)
- [imagecolorexact](#)
- [imagecolorexactalpha](#)
- [imagecolormatch](#)
- [imagecolorresolve](#)
- [imagecolorresolvealpha](#)
- [imagecolorset](#)
- [imagecolorsforindex](#)
- [imagecolorstotal](#)
- [imagecolortransparent](#)
- [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](#)
- [imagefontheight](#)
- [imagefontwidth](#)
- [imageftbbox](#)
- [imagefttext](#)
- [imagegammacorrect](#)
- [imagegd](#)
- [imagegd2](#)
- [imagegif](#)
- [imageinterlace](#)
- [imageistruecolor](#)
- [imagejpeg](#)
- [imageline](#)
- [imageloadfont](#)

- [imagepalettecopy](#)
- [imagepng](#)
- [imagepolygon](#)
- [imagepsbbox](#)
- [imagepscopyfont](#)
- [imagepsencodefont](#)
- [imagepsextendfont](#)
- [imagepsfreefont](#)
- [imagepsloadfont](#)
- [imagepslantfont](#)
- [imagepstext](#)
- [imagerectangle](#)
- [imagerotate](#)
- [imagesavealpha](#)
- [imagesetbrush](#)
- [imagesetpixel](#)
- [imagesetstyle](#)
- [imagesetthickness](#)
- [imagesettile](#)
- [imagestring](#)
- [imagestringup](#)
- [imagesx](#)
- [imagesy](#)
- [imagetruecolortopalette](#)
- [imagettfbbox](#)
- [imagettftext](#)
- [imagetypes](#)
- [imagewbmp](#)
- [iptcembed](#)
- [iptcparse](#)
- [jpeg2wbmp](#)
- [png2wbmp](#)
- [read_exif_data](#)

7.41.9 `exif_read_data()` : Lit les en-têtes EXIF des images JPEG et 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. Elle retourne un tableau associatif avec comme index les noms des en-têtes, et en valeur, les valeurs des en-têtes. Si aucune donnée n'a pu être lue, elle retourne `FALSE`.

`filename` est le nom du fichier à lire. Il ne peut pas être une URL.

`sections` est une liste de valeur séparée par des virgules des sections qui devront être présentées dans le tableau de résultat.

FILE	FileName (nom du fichier), FileSize (taille du fichier), FileDateTime (date de modification du fichier), SectionsFound (sections trouvées)
COMPUTED	html (attributs HTML), Width (largeur), Height (Hauteur), IsColor (couleur ou noir et blanc) et quelques autres si ils sont disponibles.
ANY_TAG	Toutes les informations concernant cette balise, comme par exemple IFD0, EXIF, ...
IFD0	Toutes les balises IFD0. Dans les images normales, ils contiennent les dimensions de l'images, etc...
THUMBNAIL	Un fichier qui contient une minitature, si il y a un second IFD. Toutes les informations mises en balises à propos de cette miniature seront stockées dans cette section.
COMMENT	En-tête de commentaire des images JPEG.

EXIF	La section EXIF est une sous section de la section IFD0. Elle contient des informations plus détaillées sur les images. La plupart de ces index sont reliés aux appareils numériques.

`arrays` spécifie si chaque section doit être un tableau ou non. Les sections **FILE**, **COMPUTED** et **THUMBNAIL** seront toujours transformées en tableau, car elle contiennent des noms qui risquent d'être en conflit.

`thumbnail` indique si il faut lire la miniature elle-même, et pas seulement les données en balises.

Note

Les en-têtes Exif sont souvent présents dans les en-têtes d'images JPEG/TIFF, générées par les appareils numériques, mais malheureusement, chaque fabricant d'appareils ont tous une idée différente pour étiquetter leurs images, et vous ne pourrez pas compter sur des en-têtes communs.

Exemple avec <code>exif_read_data</code>

```
<?php
echo "test1.jpg : <br />\n";
$exif = exif_read_data ('tests/test1.jpg','IFD0');
echo $exif===false ? "No header data found.<br />\n" : "Image contains headers<br />";
$exif = exif_read_data ('tests/test2.jpg',0,true);
echo "test2.jpg : <br />\n";
foreach($exif as $key=>$section) {
    foreach($section as $name=>$val) {
        echo "$key.$name: $val<br />\n";
    }
}
}??>
```

La première commande échoue, car l'image n'a pas d'en-têtes d'informations.

```
test1.jpg:
No header data found.
test2.jpg:
FILE.FileName: test2.jpg
FILE.FileDateTime: 1017666176
FILE.FileSize: 1240
FILE.FileType: 2
FILE.SectionsFound: ANY_TAG, IFD0, THUMBNAIL, COMMENT
COMPUTED.html: width="1" height="1"
COMPUTED.Height: 1
COMPUTED.Width: 1
COMPUTED.IsColor: 1
COMPUTED.ByteOrderMotorola: 1
COMPUTED.UserComment: Exif test image.
COMPUTED.UserCommentEncoding: ASCII
COMPUTED.Copyright: Photo (c) M.Boerger, Edited by M.Boerger.
COMPUTED.Copyright.Photographer: Photo (c) M.Boerger
COMPUTED.Copyright.Editor: Edited by M.Boerger.
IFD0.Copyright: Photo (c) M.Boerger
IFD0.UserComment: ASCII
THUMBNAIL.JPEGInterchangeFormat: 134
THUMBNAIL.JPEGInterchangeFormatLength: 523
COMMENT.0: Comment #1.
COMMENT.1: Comment #2.
COMMENT.2: Comment #3end
THUMBNAIL.JPEGInterchangeFormat: 134
THUMBNAIL.Thumbnail.Height: 1
THUMBNAIL.Thumbnail.Width: 1
```

Note

Si l'image contient des données IFD0, alors la section **COMPUTED** contient l'entrée **ByteOrderMotorola** qui vaut 0 pour little-endian (intel) et 1 pour big-endian (motorola). Ceci a été ajouté en PHP 4.3.

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 valeur séparée par des virgules de photos et de copyright de l'auteur.

Note

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.

Si les commentaires utilisateurs utilisent l'Unicode ou bien le JIS et que le module mbstring est accessible, ces jeux de caractères seront automatiquement changés en fonction la configuration de EXIF dans le fichier `php.ini`. Cette possibilité a été ajoutée en PHP 4.3.

Note

La hauteur (Height) et la largeur (Width) sont calculés de la même façon que la fonction [getimagesize](#) le fait, et leurs valeurs ne devraient jamais différer. De même, l'index html est la représentation textuelle utilisée dans une balise image HTML classique.

Note

Depuis PHP 4.3, cette fonction peut aussi lire les sections ^{IFD} incluent dans les tableaux, et retournée sous cette forme. De plus, la taille d'une miniature intégrée est retournée dans le sous tableau **THUMBNAIL** et la fonction [exif_read_data](#) peut retourner les miniatures au format ^{TIFF}. Enfin, il n'y a plus de longueur maximale pour les valeurs retournées (hormis la place en mémoire).

Note

Cette fonction est uniquement disponible en PHP 4, si il a été compilé avec l'option `--enable-exif`. Ses fonctionnalités et comportement ont changés en PHP 4.2. Les premières versions étaient très instables.

Depuis PHP 4.3, les commentaires utilisateurs peuvent être automatiquement convertis d'un jeu de caractères à l'autre avec l'extension `--enable-mbstring`.

Cette fonction ne requiert pas la librairie GD.

Voir aussi [exif_thumbnail](#) et [getimagesize](#).

7.41.10 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 il n'y a pas de miniature, `FALSE` sera retourné.

Les paramètres `width`, `height` et `imagetype` sont disponibles depuis PHP 4.3 et retourne les dimensions de la miniature, ainsi que son type. Il est possible que [exif_thumbnail](#) ne puisse pas créer une image, mais en détermine la taille. Dans ce cas, la valeur retournée par la fonction est `FALSE`, mais `width` et `height` se voient assignés une valeur.

Si vous voulez exploiter les miniatures avec cette fonction, il faut nevoyer les en-têtes HTTP avec la fonction [header](#). L'exemple ci-dessous montre comment faire :

Exemple d'utilisation de miniature avec [exif_thumbnail](#)

```
<?php
```

```

if (array_key_exists('file',$REQUEST)) {
    $image = exif_thumbnail($REQUEST['file'], $width, $height, $type);
} else {
    $image = false;
}
if ($image!==false) {
    header("Content-type: ".image_type_to_mime_type($type));
    echo $image;
    exit;
} else {
    // aucune miniature, gestion d'erreur
    echo "Pas de miniature disponible";
}
?>

```

Depuis PHP 4.3, la fonction `exif_thumbnail` lit aussi les miniatures dans les images au format `TIFF`.

Note

Cette fonction n'est disponible que si vous avez compilé PHP avec l'option `--enable-exif`. Les fonctionnalités et le comportement de cette fonction ont changé en PHP 4.2

Cette fonction ne requiert pas la librairie GD.

Voir aussi `exif_read_data` et `image_type_to_mime_type`.

7.41.11 gd_info() : Retourne des informations à propos de la librairie GD installée

array **gd_info** (void)

Retourne un tableau associatif décrivant la version et les possibilités de la librairie GD installée.

Attribut	Signification
GD Version	chaîne de caractères décrivant la version de libgd qui est installée.
Freetype Support	booléen . TRUE si le support Freetype est installé.
Freetype Linkage	chaîne de caractères décrivant la façon de laquelle Freetype a été lié. Les valeurs attendues sont : 'with freetype', 'with TTF library', et 'with unknown library'. Cet élément ne sera défini que si Freetype Support est évalué TRUE .
T1Lib Support	booléen . TRUE si le support T1Lib est inclus.
GIF Read Support	booléen . TRUE si le support pour la lecture des images GIF est inclus.
GIF Create Support	booléen . TRUE si le support pour la création des images GIF est inclus.
JPG Support	booléen . TRUE si le support de JPG est inclus.
PNG Support	booléen . TRUE si le support de PNG est inclus.
WBMP Support	booléen . TRUE si le support de WBMP est inclus.
XBM Support	booléen . TRUE si le support de XBM est inclus.

Utilisation de `gd_info`

```

<?php
var_dump(gd_info());

/* Sortie typique
=====
array(9) {
  ["GD Version"]=>
  string(24) "bundled (2.0 compatible)"
  ["FreeType Support"]=>
  bool(false)
  ["T1Lib Support"]=>
  bool(false)
  ["GIF Read Support"]=>
  bool(true)
  ["GIF Create Support"]=>
  bool(false)
  ["JPG Support"]=>
  bool(false)
  ["PNG Support"]=>
  bool(true)
  ["WBMP Support"]=>

```



```

bool(true)
["XBM Support"]=>
bool(false)
}
*/
?>

```

Voir aussi [imagepng](#) , [imagejpeg](#) , [imagegif](#) , [imagewbmp](#) , et [imagetypes](#) .

7.41.12 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} , ^{PSD} ou ^{BMP} et en retourner les dimensions, le type d'image, et une chaîne type "height/width", à placer dans une balise ^{HTML} ou ^{IMG} normale.

Si l'accès au fichier ^{filename} est impossible, ou si ce n'est pas un format d'image supporté, [getimagesize](#) retournera ^{FALSE} et générera une alerte.

[getimagesize](#) retourne un tableau de 4 éléments. L'index 0 contient la longueur. L'index 1 contient la largeur. L'index 2 contient le type de l'image : 1 = ^{GIF} , 2 = ^{JPG} , 3 = ^{PNG} , 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 ajoutée en PHP 4.3. L'index 3 contient la chaîne à placer dans les balises ^{HTML} : "height=xxx width=xxx" .

Exemple avec [getimagesize](#)

```

<?php
list($width, $height, $type, $attr) = getimagesize("img/flag.jpg");
echo "<img src=\"img/flag.jpg\" $attr>";
?>

```

[getimagesize](#) avec une URL

```

<?php
$size = getimagesize("http://www.php.net/gifs/logo.gif");
?>

```

Avec les images ^{JPEG} , deux en-têtes supplémentaires sont retournés : ^{channel} et ^{bits} . ^{channel} vaudra 3 avec les images RGB, et 4 avec les images CMYK. ^{bits} est le nombre de bits de chaque couleur.

Depuis PHP 4.3, ^{bits} et ^{channels} sont présent pour tous les 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 images. Dans ces cas, [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);
$f=fopen($filename, "rb");

```

```

if ($size && $fp) {
    header("Content-type: {$size['mime']}");
    fpassthru($fp);
    exit;
} else {
    // error
}
?>

```

Si l'accès à `filename` est impossible, ou si ce n'est pas une image valide, `getimagesize` retournera `NULL` et générera une alerte.

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);
}
?>

```

Note

`getimagesize` ne requiert pas la bibliothèque GD.

Le support URL a été ajouté en PHP 4.0.5.

Voir aussi `image_type_to_mime_type`, `exif_imagetype`, `exif_read_data` et `exif_thumbnail`.

Le support du format TIFF a été ajouté en PHP 4.2.

7.41.13 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` (file)

```

<?php
header ("Content-type: ".image_type_to_mime_type (IMAGETYPE_PNG));
?>

```

Note

Cette fonction ne requiert par la librairie GD.

Voir ausi `getimagesize`, `exif_imagetype`, `exif_read_data` et `exif_thumbnail`.

7.41.14 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, l'image sera créée dans ce fichier, et sinon, elle sera envoyée au navigateur. `image` est une image valide, créée avec la fonction [imagecreate](#).

Le nom de fichier `filename` est optionnel, et si 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.

Note

Le support WBMP n'est disponible qu'avec GD-1.8 ou plus récent.

Voir aussi [imagewbmp](#).

7.41.15 imagealphablending() : Modifie le mode de blending d'une image

```
int imagealphablending ( resource image , bool blendmode )
```

[imagealphablending](#) fournit deux modes de dessins 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.

Note

[imagealphablending](#) a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

7.41.16 imageantialias() : Utiliser ou non les fonctions d'antialias

```
bool imageantialias ( int im , bool on )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Voir aussi [imagecreatetruecolor](#).

7.41.17 imagearc() : Dessine une ellipse partielle

```
int 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 = imagecreate(200, 200);

// Allocation de couleurs
$white = imagecolorallocate($img, 255, 255, 255);

// Dessin d'un cercle blanc
imagearc($img, 100, 100, 150, 150, 0, 360, $white);

// Affichage au navigateur
header("Content-type: image/png");
imagepng($img);

// Libération de la mémoire
imagedestroy($img);

?>

```

Voir aussi [imageellipse](#) , [imagefilledellipse](#) et [imagefilledarc](#) .

7.41.18 imagechar() : Dessine un caractère horizontalement

```
int 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 grande est la police).

Voir aussi [imageloadfont](#) .

7.41.19 imagecharup() : Dessine un caractère verticalement

```
int imagecharup ( resource image , int font , int x , int y , string c , int color )
```

[imagecharup](#) 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 grande est la police).

Voir aussi [imageloadfont](#) .

7.41.20 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 [imagecreate](#) . [imagecolorallocate](#) doit être appelée pour créer chaque couleur qui sera représentée par ^{image} .

Exemple avec `imagecolorallocate`

```

<?php
$white = imagecolorallocate($im, 255,255,255);
$black = imagecolorallocate($im, 0,0,0);

?>

```

Voir aussi [imagecolorallocatealpha](#) et [imagecolordeallocate](#) .

7.41.21 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.

Voir aussi [imagecolorallocate](#) et [imagecolordeallocate](#) .

7.41.22 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 librairie GD library 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 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](#) .

7.41.23 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.

Voir aussi [imagecolorexact](#) .

7.41.24 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`.

Voir aussi [imagecolorexactalpha](#).

Note

imagecolorclosestalpha a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

7.41.25 imagecolorclosesthw() : Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir (hue, white and blackness)

```
int imagecolorclosesthw ( resource image , int red , int green , int blue )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.41.26 imagecolordeallocate() : Supprime une couleur d'une image

```
int imagecolordeallocate ( resource image , int color )
```

[imagecolordeallocate](#) désalloue la couleur `color` précédemment allouée avec la fonction [imagecolorallocate](#), pour l'image `image`.

Exemple avec imagecolordeallocate

<pre><?php \$white = imagecolorallocate(\$im, 255, 255, 255); imagecolordeallocate(\$im, \$white); ?></pre>

Voir aussi [imagecolorallocate](#) et [imagecolorallocatealpha](#).

7.41.27 imagecolorexact() : Retourne l'index de la couleur donnée

```
int imagecolorexact ( resource image , int red , int green , int blue )
```

[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.

Voir aussi [imagecolorclosest](#).

7.41.28 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.

Voir aussi [imagecolorclosestalpha](#).

7.41.29 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.

Voir aussi [imagecreatetruecolor](#).

7.41.30 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.

Voir aussi [imagecolorclosest](#).

7.41.31 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.

Voir aussi [imagecolorclosestalpha](#).

Note

imagecolorresolvealpha a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

7.41.32 imagecolorset() : Change la couleur dans une palette à l'index donné.

```
bool 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 .

7.41.33 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) qui contiennent les valeurs de la couleur correspondante.

Voir aussi imagecolorat et imagecolorexact .

7.41.34 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 .

7.41.35 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 imagecreate et ^{color} est un identifiant de couleur retourné par imagecolorallocate .

L'identifiant de la nouvelle (ou courante) couleur transparente est retourné.

7.41.36 imagecopy() : Copie une partie d'une image

```
int 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} .

7.41.37 imagecopymerge() : Copie et fusionne une partie d'une image

```
int 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 .

Note

imagecopymerge a été ajoutée en PHP 4.0.6.

7.41.38 imagecopymergegray() : Copie et fusionne une partie d'une image en niveaux de gris

```
int 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

imagecopymergegray a été ajoutée en PHP 4.0.6.

7.41.39 imagecopyresampled() : Copie, redimensionne, rééchantillonne une image

```
int imagecopyresampled ( resource dst_im , resource src_im , int dstx , int dsty , int srcx , int srcy , int dstw , int dsth , int srcw , int srch )
```

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. `dst_im` est l'image de destination, `src_im` est l'image source. Si les hauteurs et largeur 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, mais si les régions se chevauchent, les résultats sont imprévisibles.

Note

Il y a un problème du 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 true color, comme celles générée par la fonction [imagecreatetruecolor](#).

Note

[imagecopyresampled](#) a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

Voir aussi [imagecopyresized](#).

7.41.40 imagecopyresized() : Copie et redimensionne une partie d'une image

```
int imagecopyresized ( resource dst_im , resource src_im , int dstx , int dsty , int srcx , int srcy , int dstw ,
int dsth , int srcw , int srch )
```

[imagecopyresized](#) copie une partie rectangulaire d'une image dans une autre image de destination. ^{dst_im} est l'image de destination, ^{src_im} 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_im} et ^{src_im} sont identiques : mais si les régions se chevauchent, le résultat risque d'être incohérent.

Note

Il y a un problème du 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 true color, comme celles générée par la fonction [imagecreatetruecolor](#).

Voir aussi [imagecopyresampled](#).

7.41.41 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 (50, 100);
or die ("Impossible d'initialiser la librairie 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);
?>
```

Voir aussi [imagedestroy](#) et [imagecreatetruecolor](#).

7.41.42 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

La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

7.41.43 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.

Array

Attention

La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

7.41.44 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.

Array

Attention

La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

7.41.45 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.

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 (gracieusement offert par vic@zysys.com)

```

<?php
function LoadGif ($imgname) {
    $im = @imagecreatefromgif ($imgname); /* Tentative d'ouverture */
    if (!$im) { /* Test d'échec */
        $im = imagecreate (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;
}
?>

```

Note

Etant donné que toutes les fonctions de gestion des ^{GIF} ont été supprimées de la bibliothèque GD version 1.6, cette fonction n'est pas disponible si vous utilisez cette version de la librairie.

Array

Attention

La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

7.41.46 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 s'affiche 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 (gracieusement offert par vic@zysys.com)

```

<?php
function loadjpeg($imgname) {
    $im = @imagecreatefromjpeg($imgname); /* Tentative d'ouverture */
    if (!$im) { /* Vérification */
        $im = imagecreate(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

La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

7.41.47 imagecreatefrompng() : Crée une nouvelle image ^{PNG} à partir d'un fichier ou d'une URL

resource **imagecreatefrompng** (string filename)

7.41.46 imagecreatefromjpeg() : Crée une nouvelle image JPEG à partir d'un fichier ou d'une URL

imagecreatefrompng 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 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 (gracieusement offert par vic@zysys.com)

```
<?php
function LoadPNG($imgname) {
    $im = @imagecreatefrompng($imgname); /* Tentative d'ouverture */
    if (!$im) { /* Vérification */
        $im = imagecreate(150, 30); /* Création d'une image blanche */
        $bgc = imagecolorallocate($im, 255, 255, 255);
        $stc = imagecolorallocate($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", $stc);
    }
    return $im;
}
?>
```

Array

Attention

La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

7.41.48 imagecreatefromstring() : Crée une image à partir d'une chaîne

resource **imagecreatefromstring** (string image)

imagecreatefromstring retourne un identifiant d'image représentant la chaîne `image`.

7.41.49 imagecreatefromwbmp() : Crée une image depuis un fichier WBMP

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 WBMP (gracieusement proposé par vic@zysys.com)

```
<?php
function loadwbmp($imgname) {
    $im = @imagecreatefromwbmp($imgname); /* Tentative d'ouverture */
    if (!$im) { /* Vérification que cela s'est bien passé */
        $im = imagecreate(20, 20); /* Crée une image blanche */
        $bgc = imagecolorallocate($im, 255, 255, 255);
        $stc = imagecolorallocate($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", $stc);
    }
    return $im;
}
?>
```

Note

Le support WBMP n'est disponible qu'avec GD-1.8 ou plus récent.

Array

Attention

La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si `allow_url_fopen` est activé.

7.41.50 imagecreatefromxbm() : Crée une image à partir d'un fichier XBM

resource **imagecreatefromxbm** (string filename)

imagecreatefromxbm retourne un identifiant d'image représentant l'image obtenue à partir du fichier XBM filename .

Array

Attention

La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si `allow_url_fopen` est activé.

7.41.51 imagecreatefromxpm() : Crée une image à partir d'un fichier XPM

resource **imagecreatefromxpm** (string filename)

imagecreatefromxpm retourne un identifiant d'image représentant l'image obtenue à partir du fichier XPM filename .

Array

Attention

La version Windows de ^{PHP} ne supporte pas l'accès aux fichiers distants avec cette fonction, même si `allow_url_fopen` est activé.

7.41.52 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 .

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);
?>
```

Note

imagecreatetruecolor a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

Note
<code>imagecreatetruecolor</code> ne fonctionne pas avec le format ^{GIF} .

Voir aussi `imagedestroy` et `imagecreate`.

7.41.53 imagedashedline() : Dessine une ligne pointillée

```
int 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.

7.41.54 imagedestroy() : détruit une image

```
int imagedestroy ( resource image )
```

`imagedestroy` libère toute la mémoire associée à l'image `image`. `image` est un identifiant d'image valide retourné par `imagecreate`.

7.41.55 imageellipse() : Dessine une ellipse

```
int 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`.

Note
<code>imageellipse</code> a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

Voir aussi `imagearc`.

7.41.56 imagefill() : Remplit

```
int 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)).

7.41.57 imagefilledarc() : Dessine une ellipse partielle et la remplit

```
int 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`. `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.

Note

[imagefilledarc](#) a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

7.41.58 imagefilledellipse() : Dessine une ellipse pleine

```
int imagefilledellipse ( resource image , int cx , int cy , int w , int h , int color )
```

[imagefilledellipse](#) dessine une ellipse centrée sur le point (c_x , c_y). 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`.

Note

[imagefilledellipse](#) a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

Voir aussi [imagefilledarc](#).

7.41.59 imagefilledpolygon() : Dessine un polygone rempli

```
int imagefilledpolygon ( resource image , array points , int num_points , int color )
```

[imagefilledpolygon](#) dessine un polygone rempli dans l'image `image`. `points` est un tableau PHP qui contient les sommets des polygones sous la forme : `points[0] = x0`, `points[1] = y0`, `points[2] = x1`, `points[3] = y1`, etc. `num_points` est le nombre total de sommets.

7.41.60 imagefilledrectangle() : Dessine un rectangle rempli

```
int 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 (x_1 , y_1) et finissant au sommet inférieur droit (x_2 , y_2). Le coin supérieur gauche est l'origine (0, 0).

7.41.61 imagefilltoborder() : Remplit avec une région avec une couleur spécifique

```
int 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)).

7.41.62 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 .

7.41.63 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 .

7.41.64 imageftbbox() : Calcule le rectangle d'encadrement pour un texte, en utilisant la police courante et freetype2

```
array imageftbbox ( int size , int angle , string font_file , string text , array extrainfo )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.41.65 imagefttext() : Ecrit du texte dans une image avec la police courante et FreeType 2

```
array imagefttext ( resource image , int size , int 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.

7.41.66 imagegammacorrect() : Applique une correction gamma à l'image

```
int 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 est ^{outputgamma} .

7.41.67 imagegd() : Génère une image au format GD, vers le navigateur ou un fichier

```
int imagegd ( resource image , string filename )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.41.68 imagegd2() : Génère une image au format GD2, vers le navigateur ou un fichier

```
int imagegd2 ( resource image , string filename , int chunk_size , int type )
```

imagegd2 génère une image au format GD2, vers le navigateur ou un fichier.

Le paramètre optionnel `type` vaut soit `raw` (brut) soit `compressed` (compressé).

Note

Les paramètres optionnels <code>chunk_size</code> et <code>type</code> ont été ajouté en ^{PHP} 4.3.1.

7.41.69 imagegif() : Envoie une image GIF vers un navigateur ou un fichier

```
int 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 imagecreate.

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 nom du fichier est optionnel, et dans ce cas, l'image sera transmise directement à la sortie standard. En envoyant une 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 librairie.

Le code suivant vous permet d'écrire des scripts PHP plus portables : le type de GD est automatiquement détecté. Il remplace la séquence <code>Header("Content-type: image/gif"); ImageGif(\$im);</code> 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($im);
}
elseif (function_exists("imagebmp")) {
    header("Content-type: image/vnd.wap.wbmp");
    imagebmp($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](#) , [imageipeq](#) et [imagetypes](#) .

7.41.70 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 interlacée, et 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.

7.41.71 imageistruecolor() : Détermine si une image est une image truecolor.

```
bool imageistruecolor ( resource image )
```

[imageistruecolor](#) détermine si l'image ^{image} est une image truecolor.

Voir aussi [imagecreatetruecolor](#) .

7.41.72 imagejpeg() : Envoie une image JPEG vers un navigateur ou un fichier.

```
int 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 [imagecreate](#) .

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/jpeg content-type} (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 est compilé avec GD-1.8 ou plus récent.

^{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 à 100.

Si vous voulez générer des images JPEG progressive, vous devez choisir l'entrelacement à l'aide de la fonction [imageinterlace](#).

Voir aussi [imagepng](#), [imagewbmp](#), [imagegif](#), [imageinterlace](#), [imagegif](#) et [imagetypes](#).

7.41.73 imageline() : Dessine une ligne

```
int imageline ( resource image , int x1 , int y1 , int x2 , int y2 , int color )
```

[imageline](#) dessine une ligne depuis le point (x^1 , y^1) jusqu'au point (x^2 , y^2) (le coin supérieur gauche est l'origine (0,0)) dans l'image [image](#) et avec la couleur [color](#).

Voir aussi [imagecreate](#) et [imagecolorallocate](#).

7.41.74 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.

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_caractères*largeur*hauteur) octets.

Voir aussi [imagefontwidth](#) et [imagefontheight](#).

7.41.75 imagepalettecopy() : Copie la palette d'une image à l'autre

```
int imagepalettecopy ( resource destination , resource source )
```

[imagepalettecopy](#) copie la palette de l'image [source](#) dans l'image [destination](#).

7.41.76 imagepng() : Envoie une image PNG vers un navigateur ou un fichier.

```
int 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'envoie 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](#).

7.41.77 imagepolygon() : Dessine un polygone

```
int 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.

Voir aussi [imagecreate](#) et [imagecreatetruecolor](#).

7.41.78 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.

[imagepsbbox](#) retourne un tableau contenant les éléments suivants :

0	Abscisse inférieure gauche
1	Ordonnée inférieure gauche
2	Abscisse supérieure droite

Voir aussi [imagepstext](#) .

7.41.79 imagepscopyfont() : Fait une copie d'une police chargée pour modification ultérieure.

```
int imagepscopyfont ( int fontindex )
```

Utilisez cette fonction si vous devez faire des modifications dans une polie, comme par exemple 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.

Voir aussi [imagepsloadfont](#) .

7.41.80 imagepsencodefont() : Change le codage vectoriel d'un caractère dans une police

```
int imagepsencodefont ( int 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.

7.41.81 imagepsextendfont() : Etend ou condense une police de caractères

```
bool imagepsextendfont ( int font_index , float extend )
```

[imagepsextendfont](#) étend ou condense la police de caractères `font` . Si la valeur de `extend` est inférieure à 1, ce sera une condensation.

7.41.82 imagepsfreefont() : Libère la mémoire occupée par une police PostScript Type 1

```
void imagepsfreefont ( int fontindex )
```

[imagepsfreefont](#) libère la mémoire occupée par une police PostScript Type 1.

Voir aussi [imagepsloadfont](#).

7.41.83 imagepsloadfont() : Charge une police PostScript Type 1 depuis un fichier

`int 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.

```
<?php
header ("Content-type: image/jpeg");
$im = imagecreate (350, 45);
$black = imagecolorallocate ($im, 0, 0, 0);
$white = imagecolorallocate ($im, 255, 255, 255);
$font = imagepsloadfont ("bcbi.pfb"); // ou bien situez un de vos fichiers .pfb
imagepsetext ($im, "Test ... Ca marche!", $font, 32, $white, $black, 32, 32);
imagepsfreefont ($font);
imagejpeg ($im, "", 100); //Meilleure qualité...
imagedestroy ($im);
?>
```

Voir aussi [imagepsfreefont](#).

7.41.84 imagepsslantfont() : Incline une police de caractères PostScript

`bool imagepsslantfont ( int font_index , float slant )`

[imagepsslantfont](#) met en italique la police de caractères `font` avec le coefficient `slant`.

7.41.85 imagepsetext() : Dessine un texte sur une image avec une police PostScript Type1

`array imagepsetext ( resource image , string text , int font , int size , int foreground , int background , int x , int y , int space , int tightness , float angle , int antialias_steps )`

`foreground` est la couleur dans 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 recommandé 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.

`imagepext` 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 `imagepsbbox`.

7.41.86 imagerectangle() : Dessine un rectangle

```
int 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`, et en commençant au point supérieur gauche (x^1 , y^1), et en finissant au point inférieur droit (x^2 , y^2). Le coin supérieur gauche est l'origine (0,0).

7.41.87 imagerotate() : Fait tourner une image d'un angle

```
resource imagerotate ( resource src_im , float angle , int bgd_color )
```

`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.

7.41.88 imagesavealpha() : Définit l'option pour enregistrer les informations complètes du canal alpha lors de la sauvegarde 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 votre, essayez de charger le script avec un navigateur compatible avec les canaux alpha, c'est à

563

```
$w2 = imagecolorallocate($brush, 255, 255, 255);
imagecolortransparent ($brush, $w2);
imagesetbrush ($im, $brush);
imageline ($im, 100, 0, 0, 100, IMG_COLOR_STYLEDBRUSHED);

imagejpeg ($im);
imagedestroy ($im);
?>
```

Note

Cette fonction a été ajoutée en PHP 4.0.6.

Voir aussi [imagesetbrush](#) et [imageline](#) .

7.41.92 imagesetthickness() : Modifie l'épaisseur d'un trait

```
void 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.

Note

[imagesetthickness](#) a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

7.41.93 imagesettile() : Modifie l'image utilisée pour le carrelage

```
int imagesettile ( resource image , resource tile )
```

[imagesettile](#) remplace l'image de pavement 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.

7.41.94 imagestring() : Dessine une chaîne horizontale

```
int imagestring ( resource image , int font , int x , int y , string s , int col )
```

[imagestring](#) dessine une la chaîne sur une ligne horizontale, dans l'image `image` , aux coordonnées (`x` , `y`) (le coin supérieur gauche est l'origine (0,0)) dans la couleur `color` . Si l'argument de police vaut 1, 2, 3, 4 ou 5, une des polices par défaut sera utilisée.

Voir aussi [imageloadfont](#) .

7.41.95 imagestringup() : Dessine une chaîne verticale

`int 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.

7.41.96 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
$img = imagecreate(300, 200);
echo imagesx($img); // 300
?>
```

Voir aussi getimagesize, imagecreate et imagesy.

7.41.97 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
$img = imagecreate(300, 200);
echo imagesy($img); // 200
?>
```

Voir aussi getimagesize, imagecreate et imagesx.

7.41.98 imagetruecolortopalette() : Convertit une image en vraies couleurs en image à palette

`void 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 librairie du "Independent JPEG Group", qui est tout simplement génial. Le code a été modifié pour préserver l'essentiel du canal alph 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 a toujours le meilleur rendu.

Si `dither` vaut `TRUE`, cela indique que l'image doit être ditherée : 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
<code>imagetruecolorpalette</code> a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

7.41.99 `imageftbbox()` : Retourne le rectangle entourant un texte et dessiné avec une police TrueType.

array **imageftbbox** (int size , int angle , string fontfile , string text)

[`imageftbbox`](#) calcule et retourne le rectangle entourant le texte `text`, écrit avec une police truetype.

`text`
La chaîne à mesurer.

`size`
La taille de la police en pixel.

`fontfile`
Le nom de la police TrueType (peut aussi être une URL.)

`angle`
Angle en degré dans lequel le texte `text` va être mesuré.

[`imageftbbox`](#) 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.

[`imageftbbox`](#) requiert les bibliothèques GD et Freetype.

Voir aussi [`imagefttext`](#).

7.41.100 `imagefttext()` : Dessine un texte avec une police TrueType

array **imagefttext** (resource image , int size , int angle , int x , int y , int color , string fontfile , string text)

[`imagefttext`](#) dessine la chaîne `text` dans l'image `image`, en commençant aux coordonnées (`x` , `y`) (le coin supérieur gauche est l'origine (0,0)), avec un angle de `angle`, et dans la couleur `color`, en utilisant la police TrueType identifiée par `fontfile`. Suivant la version de GD dont vous disposez, lorsque le paramètre `fontfile` ne commence pas par un slash '/', l'extension '.ttf' sera ajoutée au nom du fichier et la librairie essaiera de trouver un fichier avec ce nom là, dans les chemins prédéfinies de la

librairie pour les polices.

Les coordonnées (x , y) serviront de référence pour le premier caractère (en gros, le coin inférieur gauche du caractère). C'est différent de [imagestring](#) , qui utilise le coin supérieur droit.

`angle` est donné en degrés, avec degré 0 pour un texte horizontal, et en comptant les angles dans le sens inverse des aiguilles d'une montre (sens direct).

`fontfile` est le chemin jusqu'à la police TrueType à utiliser.

`text` est le texte à dessiner, incluant aussi des séquences de caractères UTF-8 (de la forme: `{`) pour générer des caractères au-delà de 255.

`color` est l'index de la couleur dans la palette. Utiliser des index négatifs, revient à supprimer l'anti-aliasing.

[imagettftext](#) retourne un tableau de 8 éléments représentant les 4 points marquants les limites du texte. L'ordre des points est: supérieur gauche, supérieur droit, inférieur droit, inférieur gauche. Les points sont nommés relativement au texte à l'horizontale [imagecolorexact](#) .

Cet exemple va générer une image ^{JPEG} noire de 400x30 pixels, avec les mots "Test en cours...Oméga: Ω" en police blanche, type Arial.

Exemple avec [imagettftext](#)

```
<?php
header("Content-type: image/jpeg");
$im = imagecreate(400,30);
$white = imagecolorallocate($im, 255,255,255);
$black = imagecolorallocate($im, 0,0,0);

// Remplacez path par votre chemin d'accès aux polices
imagettftext($im, 20, 0, 10, 20, $white, "/path/arial.ttf",
"Test en cours... Oméga: &#937;");
imagejpeg($im);
imagedestroy($im);
?>
```

[imagettftext](#) requiert les bibliothèques GD ainsi que [FreeType](#) .

Voir aussi [imagettfbbox](#) .

7.41.101 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 valables : `IMG_GIF` | `IMG_JPG` | `IMG_PNG` | `IMG_WBMP` . Pour vous assurer du support ^{PNG} , faites ceci :

Exemple avec [imagetypes](#)

```
<?php
if (imagetypes() & IMG_PNG) {
    echo "Le type PNG est supporté";
}
?>
```

7.41.102 imagewbmp() : Affiche une image WBMP

```
int 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 `imagecreate`.

`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 noir.

Voir aussi `image2wbmp`, `imagepng`, `imagegif`, `imagejpeg` et `imagetypes`.

7.41.103 iptcembed() : Intègre des données binaires IPTC dans une image JPEG

```
array iptcembed ( string iptcdata , string jpeg_file_name , int spool )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.41.104 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, retourne `FALSE`.

Voir `getimagesize` pour un exemple.

7.41.105 jpeg2wbmp() : Convertit une image JPEG en image WBMP

```
int jpeg2wbmp ( string jpegname , string wbmpname , 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 `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 plus récent.

Voir aussi `png2wbmp`.

7.41.106 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 plus récent.

Voir aussi [jpeg2wbmp](#).

7.41.107 read_exif_data() : Alias de [exif_read_data](#)

[read_exif_data](#) est un alias de [exif_read_data](#).

7.42 IMAP

7.42.1 Introduction

These functions are not limited to the `IMAP` protocol, despite their name. The underlying `c-client` library also supports `NNTP`, `POP3` and local mailbox access methods.

7.42.2 Pré-requis

Cette extension requiert la librairie C cliente. Téléchargez cette dernière version à <ftp://ftp.cac.washington.edu/imap/> et compilez la.

Puis, copiez le fichier `c-client/c-client.a` dans `/usr/local/lib` ou n'importe quel autre dossier qui soit dans le chemin de link. Enfin, copiez les fichiers `c-client/rfc822.h`, `mail.h` et `linkage.h` dans `/usr/local/include` ou n'importe quel autre dossier qui soit dans le chemin d'inclusion.

Note

Assurez vous que `c-client.a` est bien dans votre chemin de link, sous le nom `libc-client.a`.

Copiez les fichiers suivants dans le dossier `/usr/local/include` ou dans n'importe quel dossier qui soit dans votre chemin d'inclusion.

- `c-client.h`
- `imap4ri.h`
- `rfc822.h`
- `mail.h`
- `linkage.h`
- `env.h`
- `env_unix.h`
- `fs.h`
- `ftl.h`
- `misc.h`
- `nntp.h`
- `nl.h`
- `osdep.h`
- `smtp.h`
- `tcp.h`

Note

Le nom de `env_unix.h` dépend de votre OS. Assurez vous que ces fichiers sont bien copiés avec l'option de suivi des liens. Pour compiler la librairie C cliente avec SSL et/ou avec le support Kerberos, lisez la doc fournie dans la distribution.

7.42.3 Installation

Pour avoir accès à ces fonctions, vous devez compiler PHP avec l'option `--with-imap`. 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 librairie DLL `php_imap.dll` dans le `php.ini`.

Note

Suivant la librairie C cliente qui a été configurée, 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 de PHP.

Attention

L'extension IMAP ne peut être utilisée simultanément avec l'extension recode ou l'extension YAZ. Ceci est dû au fait que ces deux librairies utilisent un symbole interne commun.

7.42.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.42.5 Types de ressources

7.42.6 Constantes prédefinies

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*)
 OP_ANONYMOUS (*entier*)
 OP_SHORTCACHE (*entier*)
 OP_SILENT (*entier*)
 OP_PROTOTYPE (*entier*)
 OP_HALFOPEN (*entier*)
 OP_EXPUNGE (*entier*)
 OP_SECURE (*entier*)
 CL_EXPUNGE (*entier*)
 FT_UID (*entier*)
 FT_PEEK (*entier*)
 FT_NOT (*entier*)
 FT_INTERNAL (*entier*)
 FT_PREFETCHTEXT (*entier*)
 ST_UID (*entier*)
 ST_SILENT (*entier*)
 ST_SET (*entier*)
 CP_UID (*entier*)
 CP_MOVE (*entier*)
 SE_UID (*entier*)
 SE_FREE (*entier*)
 SE_NOPREFETCH (*entier*)
 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*)
 LATT_NOSELECT (*entier*)
 LATT_MARKED (*entier*)
 LATT_UNMARKED (*entier*)
 SORTDATE (*entier*)
 SORTARRIVAL (*entier*)
 SORTFROM (*entier*)
 SORTSUBJECT (*entier*)
 SORTTO (*entier*)

SORTCC (*entier*)
 SORTSIZE (*entier*)
 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*)

7.42.7 Voir aussi

Ce document ne peut entrer dans les détails de toutes les sujets abordés. Plus d'informations sont disponibles avec la documentation de la librairie C (`docs/internal.txt`) ainsi que les RFC suivantes : documents:

- [RFC821](#) : Simple Mail Transfer Protocol (^{SMTP}).
- [RFC822](#) : 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](#) & [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.

Attention

Des crashes et des problèmes de démarrage de ^{PHP} sont possibles lorsque vous chargez cette extension en même temps que l'extension [recode](#). Voyez l'extension [recode](#) pour plus d'informations.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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_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_uid](#)
- [imap_undelete](#)

- [imap_unsubscribe](#)
- [imap_utf7_decode](#)
- [imap_utf7_encode](#)
- [imap_utf8](#)

7.42.9 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 appelé, la pile d'alertes est vidée.

7.42.10 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'option ^{flags} est utilisée, ^{flags} 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);
print "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);
print "Nombre de messages après ajout : ". $check->Nmsgs."\n";
imap_close($stream);
?>
```

7.42.11 imap_base64() : Décode un texte encodé en BASE64

string **imap_base64** (string text)

[imap_base64](#) décode un texte encodé en BASE64. Le texte décodé est retourné sous la forme d'une chaîne (selon la [RFC 2045](#) , Section 6.8).

Voir aussi [imap_binary](#) , [base64_encode](#) et [base64_decode](#) .

7.42.12 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 RFC 2045 , Section 6.8).

imap_binary retourne la chaîne codée.

Voir aussi imap_base64 .

7.42.13 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. L'option ^{flags} est un masque qui peut contenir les valeurs suivantes :

- FT_UID – msgno est un UID
- FT_PEEK – Ne pas lever le drapeaux \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.

7.42.14 imap_bodystruct() : Lit la structure d'une section du corps d'un mail

object **imap_bodystruct** (resource stream_id , int msg_no , int section)

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.42.15 imap_check() : Vérifie le courrier de la boîte aux lettres courante.

object **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
- 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

7.42.16 imap_clearflag_full() : Supprime un flag 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 flût imap ^{stream}. Les flags ^{flag} que vous pouvez effacer sont "\\Seen", "\\Answered", "\\Flagged", "\\Deleted" et "\\Draft" (tels que définis dans la RFC2060).

Les options sont un masque de bit, qui accepte les valeurs suivantes :

Constante	Signification
ST_UID	la séquence contient des UIDs au lieu de numéro de séquence.

7.42.17 imap_close() : Termine un flût IMAP

```
bool imap_close ( resource imap_stream , int flag )
```

imap_close termine un flût IMAP. imap_close prend un argument optionnel ^{flag}, CL_EXPUNGE, qui va retirer automatiquement de la liste la boîte aux lettres.

7.42.18 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
$mbox = imap_open("{your.imap.host}", "utilisateur", "mot_de_passe", OP_HALFOPEN)
    or die("connexion impossible: ".imap_last_error());
$name1 = "nouvellepbox";
$name2 = imap_utf7_encode("nouvellepboxé");
$newname = $name1;
echo "Le nouveau nom sera '$name1'<BR>\n";
// Nous allons créer maintenant une nouvelle boîte aux lettres "phpptestbox"
// dans votre dossier inbox, vérifier son état et finalement, la supprimer
// pour remettre votre inbox dans son état initial.
if(@imap_createmailbox($mbox,imap_utf7_encode("{your.imap.host}INBOX.$newname"))){
    $status = @imap_status($mbox, "{your.imap.host}INBOX.$newname", SA_ALL);
    if($status) {
        print("Votre nouvelle boîte '$name1' est dans l'état suivant :<BR>\n");
        print("Messages:   ". $status->messages   )."<BR>\n";
        print("Récent:     ". $status->recent     )."<BR>\n";
        print("Non lus:      ". $status->unseen      )."<BR>\n";
        print("UID suivant:   ". $status->uidnext     )."<BR>\n";
        print("UID validité:". $status->uidvalidity)."<BR>\n";
        if(imap_renamemailbox($mbox, "{your.imap.host}INBOX.$newname", "{your.imap.host}INBOX.$name2")) {
            echo "renomage de la boîte aux lettres '$name1' en '$name2'<BR>\n";
            $newname=$name2;
        } else {
            print "imap_renamemailbox sur la nouvelle boîte aux lettres a échoué : ".imap_last_error()."<BR>\n";
        }
    } else {
        print "imap_status sur la nouvelle boîte aux lettres a échoué : ".imap_last_error()."<BR>\n";
    }
}
if(@imap_deletemailbox($mbox, "{your.imap.host}INBOX.$newname")) {
    print "new mailbox supprimée pour remettre tout en état<BR>\n";
} else {
    print "imap_deletemailbox ur la nouvelle boîte aux lettres a échoué : ".implode("<BR>\n",imap_errors())."<BR>\n";
}
} else {
    print "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` .

7.42.19 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](#) 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](#) .

[imap_delete](#) retourne `TRUE` .

Exemple avec [imap_delete](#)

```
<?php
$mbx = imap_open ("{your.imap.host}INBOX", "utilisateur", "mot_de_passe")
    or die ("connexion impossible: " . imap_last_error());
$check = imap_mailboxmsginfo ($mbx);
print "Nombre de messages avant effacement : " . $check->Nmsgs . "<BR>\n" ;
imap_delete ($mbx, 1);
$check = imap_mailboxmsginfo ($mbx);
print "Nombre de messages après effacement: " . $check->Nmsgs . "<BR>\n" ;
imap_expunge ($mbx);
$check = imap_mailboxmsginfo ($mbx);
print "Nombre de messages après imap_expunge: " . $check->Nmsgs . "<BR>\n" ;
imap_close ($mbx);
?>
```

7.42.20 imap_deletemailbox() : Efface une boîte aux lettres

```
bool imap_deletemailbox ( resource imap_stream , string mbox )
```

[imap_deletemailbox](#) efface la boîte aux lettres (voir [imap_open](#) pour connaître le format des noms de `mbox`).

[imap_deletemailbox](#) retourne `TRUE` en cas de succès, et `FALSE` en cas d'erreur.

Voir aussi [imap_createmailbox](#) , [imap_renamemailbox](#) et [imap_open](#) pour le format du paramètre `mbox` .

7.42.21 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 appelés, la pile d'erreur est vidée.

7.42.22 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_expunge` retourne `TRUE`.

7.42.23 `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'indice de message ou d'UIDs, si cotient FT_UID. La valeur retournée est un tableau d'objets, un par message d'en-tête décrit :

- `subject` – Le sujet du message
- `from` – Expéditeur
- `date` – Date d'expédition
- `message_id` – Identification du message
- `references` – est une référence sur l'id de ce 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
$mbox = imap_open("{your.imap.host:143}", "username", "password")
or die("connexion impossible : ".imap_last_error());

$overview = imap_fetch_overview($mbox, "2,4:6", 0);

if(is_array($overview)) {
    reset($overview);
    while( list($key,$val) = each($overview)) {
        print    $val->msgno
        . " - " . $val->date
        . " - " . $val->subject
        . "\n";
    }
}

imap_close($mbox);
?>
```

7.42.24 `imap_fetchbody()` : Retourne une section extraite du corps d'un message

string `imap_fetchbody` (resource `imap_stream` , int `msg_number` , string `part_number` , flags `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 – msgno 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](#) .

7.42.25 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 brute et complète [RFC2822](#) du message ^{msgno}, et le retourne 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).

7.42.26 imap_fetchstructure() : Lit la structure d'un message.

```
object imap_fetchstructure ( resource imap_stream , int msg_number , int options )
```

[imap_fetchstructure](#) la structure du message ^{msg_number}. [imap_fetchstructure](#) dispose d'une option *flags*, qui 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 :

type	Type primaire de corps
encoding	Codage de transfert du corps
ifsubtype	TRUE s'il y a une chaîne de sous type
subtype	sous type MIME
ifdescription	TRUE s'il y a une chaîne de description
description	Chaîne de description du contenu
ifid	TRUE s'il y a une chaîne d'identification
id	Chaîne d'identification
lines	Nombre de lignes
bytes	Nombre d'octets
ifdisposition	TRUE s'il y a une chaîne de disposition
disposition	Chaîne de disposition
ifdparameters	TRUE s'il y a un tableau de paramètres dparameters
dparameters	tableau de disposition
ifparameters	TRUE si le tableau de paramètres existe
parameters	Tableau de paramètres MIME
parts	Tableau d'objet décrivant chaque partie du message

Note

1. dparameters est un tableau d'objet où chaque objet à un "attribut" et une "valeur".
2. parameter est un tableau d'objet où chaque objet à un "attribut" et une "valeur".
3. parts est un tableau d'objets de même structure que l'objet supérieur, mais qui ne contient pas d'autres objets de même sorte.

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](#) .

7.42.27 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 librairies c-client2000.

`imap_stream` doit avoir été créé avec la fonction [imap_open](#) . Ce flôt est nécessairement ouvert en tant qu'administrateur du serveur, pour que les droits nécessaires lui soit alloué. `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)) {
    print "Utilisation actuelle : " . $quota_value['usage'];
    print "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

```
$mbx = imap_open("{your.imap.host}", "mailadmin", "password", OP_HALFOPEN)
    or die("Impossible de se connecter : ".imap_last_error());
```

```

$quota_values = imap_get_quota($mbox, "user.kalowsky");
if(is_array($quota_values)) {
    $storage = $quota_values['STORAGE'];
    print "Utilisation actuelle de la capacité de stockage : " . $storage['usage'];
    print "Quota actuel de stockage : " . $storage['limit'];

    $message = $quota_values['MESSAGE'];
    print "Niveau d'utilisation de MESSAGE : " . $message['usage'];
    print "Quota de MESSAGE : " . $message['limit'];

    /* ... */
}

imap_close($mbox);

```

Voir aussi [imap_open](#), [imap_set_quota](#) et [imap_get_quotaroot](#).

7.42.28 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 librairie 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](#)

```

$mbox = imap_open("{your.imap.host}", "kalowsky", "password", OP_HALFOPEN)
or die("can't connect: ".imap_last_error());

$quota = imap_get_quotaroot($mbox, "INBOX");
if(is_array($quota)) {
    $storage = $quota_values['STORAGE'];
    print "STORAGE usage level is: " . $storage['usage'];
    print "STORAGE limit level is: " . $storage['limit'];

    $message = $quota_values['MESSAGE'];
    print "MESSAGE niveau d'utilisation : " . $message['usage'];
    print "MESSAGE niveau d'utilisation : " . $message['limit'];

    /* ... */
}

imap_close($mbox);

```

Voir aussi [imap_open](#), [imap_set_quota](#) et [imap_get_quota](#).

7.42.29 imap_getmailboxes() : Liste les boîtes aux lettres, et retourne le détail pour 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 a les attributs 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.

`ref` ne devrait être que le serveur IMAP sous la forme `{imap_server:imap_port}`, 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
$mailbox = imap_open("{your.imap.host}", "utilisateur", "mot_de_passe", OP_HALFOPEN)
    or die("connexion impossible : ".imap_last_error());
$list = imap_getmailboxes($mailbox, "{votre.hote.imap}", "*");
if(is_array($list)) {
    reset($list);
    while (list($key, $val) = each($list)) {
        print "($key) ";
        print imap_utf7_decode($val->name).", ";
        print "'". $val->delimiter. "' , ";
        print $val->attributes. "<BR>\n";
    }
} else {
    print "imap_getmailboxes a échoué : ".imap_last_error()."\n";
}
imap_close($mailbox);
?>
```

Voir aussi imap_getsubscribed.

7.42.30 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.

7.42.31 imap_header() : Lit l'en-tête d'un message

object **imap_header** (resource imap_stream , int msg_number , int fromlength , int subjectlength , string defaulthost)

imap_header est un alias de imap_headerinfo et lui est identique en tous points.

7.42.32 imap_headerinfo() : Lit l'en-tête du message

```
object imap_headerinfo ( resource imap_stream , int msg_number , int fromlength , int subjectlength , string
default host )
```

imap_headerinfo retourne un objet contenant divers éléments d'en-tête.

```
    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)

```

7.42.33 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 chaîne contenant les en-tête des messages. Une chaîne par message.

7.42.34 imap_last_error() : Retourne la dernière erreur (si elle existe) qui est survenu 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 survenu lors de la dernière requête. La pile d'erreur n'est pas touchée. Appeler imap_last_error successivement dans nouvelles erreurs retournera la même erreur.

7.42.35 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
$mbx = imap_open("{your.imap.host}", "username", "password", OP_HALFOPEN)
    or die("can't connect: ".imap_last_error());

$list = imap_list($mbx, "{your.imap.host}", "*");
if(is_array($list)) {
    reset($list);
    while (list($key, $val) = each($list))
        print imap_utf7_decode($val). "<br>\n";
} else
    print "imap_list a échoué : ".imap_last_error(). "\n";

imap_close($mbx);
?>

```

7.42.36 imap_listmailbox() : Liste les boîtes aux lettres

```
array imap_listmailbox ( resource imap_stream , string ref , string pattern )
```

imap_listmailbox est un alias de imap_list .

7.42.37 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 leurs messages. 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` .

7.42.38 imap_listsubscribed() : Liste les boîtes aux lettres souscrites

```
array imap_listsubscribed ( resource imap_stream , string ref , string pattern )
```

imap_listsubscribed retourne un tableau avec toutes les boîtes aux lettres auxquelles vous avez souscrit. Les arguments `ref` et `pattern` indiquent respectivement, le dossier où chercher et le nom des boîtes recherchées, sous la forme d'un masque.

7.42.39 imap_lsub() : Liste toutes les boîtes aux lettres enregistrées

```
array imap_lsub ( resource imap_stream , string ref , string pattern )
```

imap_lsub retourne un tableau contenant toutes les boîtes aux lettres que vous avez enregistré. C'est presque le même résultat que imap_listmailbox , mais cette fonction ne va retourner que des boîtes aux lettres que l'utilisateur avec lequel vous vous êtes connecté sur le serveur imap a enregistré.

7.42.40 imap_mail_compose() : Crée un message MIME

```
string imap_mail_compose ( array envelope , array body )
```

Exemple avec imap_mail_compose

```
<?php
$envelope["from"]="musone@afterfive.com";
$envelope["to"]="musone@darkstar";
$envelope["cc"]="musone@edgeglobal.com";
$part1["type"]=TYPEMULTIPART;
$part1["subtype"]="mixed";
$filename="/tmp/imap.c.gz";
$f=fopen($filename,"r");
$content=fread($f,filesize($filename));
fclose($f);
$part2["type"]=TYPEAPPLICATION;
$part2["encoding"]=ENCODING;
$part2["subtype"]="octet-stream";
$part2["description"]=basename($filename);
$part2["contents.data"]=$content;
$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));
?>

```

7.42.41 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`. `msglist` est un intervalle, et pas seulement une liste numéros de message (comme décrit dans la [RFC2060](#)).

imap_mail_copy retourne `TRUE` en cas de succès et `FALSE` en cas d'erreur.

`flags` 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.

7.42.42 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`. `msglist` est un intervalle, et pas seulement une liste de messages (comme décrit dans la [RFC2060](#)).

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

`flags` est un champ de bit et peut contenir une seule valeur :

- CP_UID – La séquence de nombres contient UIDS

7.42.43 imap_mail() : Envoie un message mail

```
string imap_mail ( string to , string subject , string message , string additional_headers , string cc , string bcc , string rpath )
```

imap_mail est uniquement disponible sous PHP 3.

7.42.44 imap_mailboxmsginfo() : Lit les informations à propos de la boîte aux lettres courante.

```
object 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, et 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
$mbox = imap_open("{your.imap.host}INBOX","utilisateur", "mot_de_passe")
    or die("connexion impossible: ".imap_last_error());
$check = imap_mailboxmsginfo($mbox);
if($check) {
    print "Date: " . $check->Date . "<BR>\n" ;
    print "Pilote: " . $check->Driver . "<BR>\n" ;
    print "Mailbox: " . $check->Mailbox . "<BR>\n" ;
    print "Messages: " . $check->Nmsgs . "<BR>\n" ;
    print "Récent: " . $check->Recent . "<BR>\n" ;
    print "Non lus: " . $check->Unread . "<BR>\n" ;
    print "Effacés: " . $check->Deleted . "<BR>\n" ;
    print "Taille: " . $check->Size . "<BR>\n" ;
} else {
    print "imap_check() a échoué: ".imap_last_error(). "<BR>\n";
}
imap_close($mbox);
?>
```

7.42.45 `imap_mime_header_decode()` : Décode les éléments MIME d'une 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" & "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.

7.42.46 `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 contraire de imap_uid.

7.42.47 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.

7.42.48 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 message récents dans la boîte aux lettres courante.

Voir aussi imap_num_msg et imap_status.

7.42.49 imap_open() : Ouvre un flôt ^{IMAP} vers une boîte aux lettres

```
resource imap_open ( string mailbox , string username , string password , int options )
```

imap_open retourne un flôt ^{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 constitué 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.

Les options sont 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 flôt.

Pour se connecter à un serveur ^{IMAP}, on peut utiliser la commande suivante :

Connexion à un serveur IMAP avec imap_open

```
<?php
$mailbox = imap_open("{localhost:143}INBOX","user_id","password");
?>
```

Pour se connecter à un serveur ^{POP3} qui fonctionne sur le port 110 de la machine locale on peut utiliser la commande suivante :

Connexion à un serveur POP3 avec imap_open

```
<?php
$mbox = imap_open("{localhost:110/pop3}INBOX", "user_id", "password");
?>
```

Pour se connecter à un serveur ^{IMAP} SSL ou ^{POP3} SSL, ajoutez ^{/ssl} après le protocole :

Connexion à un serveur IMAP SSL ou POP3 SSL avec imap_open

```
<?php
$mbox = imap_open ("{localhost:993/imap/ssl}INBOX", "user_id", "password");
?>
```

Pour se connecter à un serveur SSL IMAP ou POP3 avec un certificat ajoutez ^{/ssl/novalidate-cert} après le protocole :

Connexion à un serveur IMAP SSL ou POP3 SSL avec imap_open et un certificat

```
<?php
$mbox = 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:

Connexion à un serveur NNTP avec imap_open

```
<?php
$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.

Exemple avec imap_open

```
<?php
$mbox = imap_open ("{votre.hote.imap:143}", "nom_utilisateur", "mot de passe");
echo "<p><h1>Mailboxes</h1>\n";
$folders = imap_listmailbox ($mbox, "{votre.hote.imap:143}", "");
if ($folders ==
<TT>FALSE</TT>
) {
    echo "Appel échoué<BR>\n";
} else {
    while (list ($key, $val) = each ($folders)) {
        echo $val."<BR>\n";
    }
}
echo "<p><h1>en-têtes dans INBOX</h1>\n";
$headers = imap_headers ($mbox);
if ($headers ==
<TT>FALSE</TT>
) {
    echo "Appel échoué<BR>\n";
} else {
    while (list ($key,$val) = each ($headers)) {
        echo $val."<BR>\n";
    }
}
imap_close($mbox);
?>
```

7.42.50 imap_ping() : Vérifie que le flût IMAP est toujours actif

bool **imap_ping** (resource imap_stream)

imap_ping retourne ^{TRUE} si le flût ^{imap_stream} existe toujours, et ^{FALSE} sinon.

[imap_ping](#) vérifie que le flôt ^{IMAP} est toujours actif, en lui envoyant un ping. Cette fonction permet de se rendre compte que du 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.

7.42.51 `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).

[imap_qprint](#) retourne une chaîne 8 bits (binaire).

Voir aussi [imap_8bit](#) .

7.42.52 `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} .

7.42.53 `imap_reopen()` : Ouvre un flôt 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.

Les options sont des masques de bit, qui peuvent 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 flôt. (voir [imap_delete](#) et [imap_expunge](#)).

Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.42.54 `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
- 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)) die("Erreur!\n");

reset($address_array);
while(list($key,$val)=each($address_array)){
    print "boîte      : ".$val->mailbox."<br>\n";
    print "hôte       : ".$val->host."<br>\n";
    print "nom        : ".$val->personal."<br>\n";
    print "adl        : ".$val->adl."<br>\n";
}
?>
```

7.42.55 `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}.

7.42.56 `imap_rfc822_write_address()` : Retourne une adresse email formatée correctement

string `imap_rfc822_write_address` (`string mailbox` , `string host` , `string personal`)

`mailbox` retourne une adresse email formatée correctement selon la [RFC2822](#), à partir du nom de la boîte aux lettres de l'hôte `host`, et des informations personnelles `personal`.

Exemple avec `imap_rfc822_write_address`

```
<?php
print imap_rfc822_write_address("hartmut","cvs.php.net","Hartmut Holzgraefe")."\n";
?>
```

7.42.57 `imap_scanmailbox()` : Lit la liste des boîtes aux lettres, et y recherche une chaîne

array `imap_scanmailbox` (`resource imap_stream` , `string ref` , `string pattern` , `string content`)

`imap_scanmailbox` est un alias de `imap_listscan`.

7.42.58 `imap_search()` : Retourne un tableau de message après recherche.

```
array imap_search ( resource imap_stream , string criteria , int options )
```

`imap_search` effectue une recherche dans la boîte aux lettres courante, sur le flût ^{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 message qui vérifie 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 dans le mot clés "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.

7.42.59 `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 flût est nécessairement ouvert en tant qu'administrateur du serveur, pour que les droits nécessaires lui soit alloué. `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
$mbox = imap_open ("{your.imap.host:143}", "mailadmin", "password");

if(!imap_set_quota($mbox, "user.kalowsky", 3000)) {
    print "Error in setting quota\n";
    return;
}

imap_close($mbox);
?>
```

Voir aussi `imap_open` et `imap_set_quota`.

7.42.60 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.

7.42.61 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 sequence donné.

Les flags que vous pouvez modifier sont "\Seen", "\Answered", "\Flagged", "\Deleted", "\Draft" et "\Recent" (comme défini dans la RFC2060).

Les options sont un masque de bits, et peuvent contenir les valeurs suivantes :

Constante	Signification
ST_UID	la séquence contient des UIDs au lieu de numéro de séquence.

Exemple avec `imap_setflag_full`

```
<?php
$mbox = imap_open("{your.imap.host:143}", "username", "password")
    or die("can't connect: ".imap_last_error());

$status = imap_setflag_full($mbox, "2,5", "\Seen \Flagged");

print gettype($status)."\n";
print $status."\n";

imap_close($mbox);
?>
```

7.42.62 imap_sort() : Trie des messages

array **imap_sort** (resource stream , int criteria , int reverse , int options , string search_criteria)

imap_sort retourne un tableau de nombre de message, triés suivant les paramètres suivants :

`reverse` vaut 1 pour signifier : tri inverse.

Les critères peuvent être un (et un seul) parmi les suivants :

<code>SORTDATE</code>	Date du message
<code>SORTARRIVAL</code>	Date d'arrivée
<code>SORTFROM</code>	Nom de la première boîte aux lettres de l'adresse d'origine (From address)
<code>SORTSUBJECT</code>	Sujet du message
<code>SORTTO</code>	Nom de la première boîte aux lettres de destination (To address)
<code>SORTCC</code>	Nom de la boîte aux lettres de copie cachée (cc address)
<code>SORTSIZE</code>	Taille du message en octets

Les flags dont des masques de bits, d'un ou plusieurs des éléments suivants :

<code>SE_UID</code>	Retourne l'UIDs à la place d'une séquence de nombres.
<code>SE_NOPREFETCH</code>	Ne pas pré-télécharger les messages trouvés.

7.42.63 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 boîtes 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 les valeurs de de toutes les précédents.

`status->flags` est aussi fixé : c'est un masque de bit qui peut contenir tous les flags ci-dessus.

Exemple 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) {
    print("Messages:   ". $status->messages   )."<BR>\n";
    print("Récents:    ". $status->recent     )."<BR>\n";
    print("Non lus:     ". $status->unseen     )."<BR>\n";
    print("UIDnext:    ". $status->uidnext    )."<BR>\n";
}
```



```

    print("UIDvalidité:". $status->uidvalidity)."<BR>\n";
} else {
    print "imap_status a échoué : ".imap_last_error()."\n";
}
imap_close($mbox);
?>

```

7.42.64 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.

7.42.65 imap_thread() : Retourne l'arbre de REFERENCE organisé par thread

```
array imap_thread ( resource stream_id , int options )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.42.66 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 contraire de imap_msgno .

Note

Cette fonctionnalité n'est pas supportées par les boîtes aux lettres POP3.

7.42.67 imap_undelete() : Enlève la marque d'effacement d'un message

```
bool imap_undelete ( resource imap_stream , int msg_number )
```

imap_undelete enlève la marque d'effacement du message `msg_number` , placée avec imap_delete .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.42.68 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.

7.42.69 `imap_utf7_decode()` : Décode une chaîne modifiée UTF-7.

```
string imap_utf7_decode ( string text )
```

`imap_utf7_decode` décode la chaîne UTF-7 `text` en données 8 bits.

`imap_utf7_decode` retourne les données 8bits décodées, ou `FALSE` si la chaîne `text` n'est pas au format UTF-7. Cette fonction sert à décoder des noms de boîtes aux lettres qui contiennent des caractères internationaux hors de l'espace ASCII. Le standard UTF-7 est défini dans la [RFC 2060](#) , section 5.1.3 (l'original UTF-7 a été défini dans [RFC1642](#)).

7.42.70 `imap_utf7_encode()` : Convertit des données 8bit en texte UTF-7.

```
string imap_utf7_encode ( string data )
```

`imap_utf7_encode` retourne les données `data` 8bits encodées, ou `FALSE` si une erreur est survenue. Cette fonction sert à encoder des noms de boîtes aux lettres qui contiennent des caractères internationaux hors de l'espace ASCII. Le standard UTF-7 est défini dans la [RFC 2060](#) , section 5.1.3 (l'original UTF-7 a été défini dans [RFC1642](#)).

`imap_utf7_encode` retourne un texte UTF-7.

7.42.71 `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](#) .

7.43 Informix

7.43.1 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.

7.43.2 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.

7.43.3 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.

7.43.4 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 la 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 libraries 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 BLOBs 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 types `"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 par 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 de 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 de 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 temporaire 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
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 [ini_set](#) .

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 integer

Nombre maximum de connexions persistantes à une base de données Informix, par processus.

ifx.max_links integer

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`](#).

7.43.5 Types de ressources

7.43.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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](#)

7.43.8 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.

7.43.9 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.

7.43.10 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](#).

Voir aussi [ifx_connect](#) et [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);
?>
```

7.43.11 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 .

Voir aussi ifx_pconnect et ifx_close .

Connection à un serveur Informix

```
<?php
$conn_id = ifx_pconnect ("mydb@ol_srv1", "i myself", "mypassword");
?>
```

7.43.12 ifx_copy_blob() : Duplique un objet BLOB

```
int ifx_copy_blob ( int bid )
```

ifx_copy_blob retourne `FALSE` en cas d'erreur, et sinon, l'identifiant du nouvel objet.

7.43.13 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.

7.43.14 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.

7.43.15 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 , pour un exemple.

7.43.16 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}) formaté 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 décrit 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 caractères, décrivant le résultat général de la commande, et aussi SQLSTATE et SQLCODE associé 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 .

7.43.17 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 présent, le message d'erreur associé à *errorcode*.

Exemple avec ifx_errormsg

```
<?php
printf("%s\n<BR>", ifx_errormsg(-201));
?>
```

Voir aussi ifx_error.

7.43.18 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.

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 résultats. (%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);
?>

```

7.43.19 ifx_fieldproperties() : Liste les propriétés des champs SQL

array **ifx_fieldproperties** (int result_id)

ifx_fieldproperties retourne un tableau associatif avec les nom des champs comme clé, et les données de propriétés des champs comme valeur. 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

```

<?php
$properties = ifx_fieldtypes ($resultid);
if (! isset($properties)) {
    // ... erreur ...
}
for ($i = 0; $i < count($properties); $i++) {
    $fname = key ($properties);
    printf ("%s:\t type = %s\n", $fname, $properties[$fname]);
    next ($properties);
}
?>

```

7.43.20 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 valeur. En cas d'erreur, retourne ^{FALSE}.

Nom de champs et type SQL.

```

<?php
$types = ifx_fieldtypes ($resultid);
if (! isset ($types)) {
    // ... erreur ...
}
for ($i = 0; $i < count($types); $i++) {
    $fname = key($types);
    printf ("%s :\t type = %s\n", $fname, $types[$fname]);
    next($types);
}
?>

```

7.43.21 ifx_free_blob() : Supprime un objet BLOB

int **ifx_free_blob** (int bid)

ifx_free_blob supprime l'objet BLOB ^{bid}. ifx_free_blob retourne ^{FALSE} en cas d'erreur, et sinon ^{TRUE}. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.43.22 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.

7.43.23 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.

7.43.24 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`.

7.43.25 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`.

7.43.26 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 ligne 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.sqlerrd')", étant donné qu'elle retourne les valeurs qui ont été sauvées par le pilote ifx au moment approprié.

Lire les valeurs de sqlca.sqlerrd[x]
<pre><?php /* On suppose que la première colonne d'une table 'quelconque' est un numéro de série */ \$sqlid = ifx_query("insert into sometable values(0, '2nd column', 'another column' ", \$connid); if (!\$sqlid) { // ... erreur ... } \$sqlca = ifx_getsqlca (\$sqlid);</pre>

```
$serial_value = $sqlca["sqlerrd1"];
echo "Le numéro de série de la valeur insérée
     est : " . $serial_value . "<BR>\n";
?>
```

7.43.27 ifx_htmltbl_result() : Lit toutes les lignes d'un tableau, et la 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 la met sous la forme d'un tableau HTML, ou ^{FALSE} en cas d'erreur.

Affiche les lignes avec des balises HTML. Le second argument permet de modifier les options de 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);
?>
```

7.43.28 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".

7.43.29 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 colonne dans la requête.

7.43.30 ifx_num_rows() : Compte le nombre de ligne déjà lues dans un résultat.

```
int ifx_num_rows ( int result_id )
```

ifx_num_rows compte le nombre de ligne déjà lues dans le résultat result_id après ifx_query ou ifx_do.

7.43.31 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 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.

7.43.32 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 ligne 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 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](#) .

7.43.33 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 4 d'autres fonctions pour lire les résultats. Pour un exemple, reportez-vous à [ifx_affected_rows](#) pour connaître le nombre de lignes affectées.

[ifx_query](#) envoie une requête au serveur actif courant, associé à l'identifiant de connexion `link_identifier` . Si `link_identifier` n'est pas fourni, la dernière connexion ouverte sera utilisée. Si aucune connexion n'a été ouverte, [ifx_query](#) va essayer d'en créer une, en appelant [ifx_connect](#) .

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 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_connect](#) .

```
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,'HR0',?,?)";
$res_id = ifx_query($query, $conn_id, $BLOBidarray);
if (! $res_id) {
    // ... erreur ...
}
// libération du résultat
ifx_free_result($res_id);
?>

```

7.43.34 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.

7.43.35 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.

7.43.36 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.

7.43.37 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.

7.43.38 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.

7.43.39 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.

7.43.40 ifxus_open_slob() : Ouvre un objet SLOB

```
int ifxus_open_slob ( long 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 une ressource en cas de réussite, et ^{FALSE} sinon.

7.43.41 ifxus_read_slob() : Lit n bytes d'un objet SLOB

```
int ifxus_read_slob ( long bid , long 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.

7.43.42 ifxus_seek_slob() : Fixe le fichier courant, ou la position courante

```
int ifxus_seek_slob ( long bid , int mode , long 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.

7.43.43 ifxus_tell_slob() : Retourne le fichier courant, ou la position courante

```
int ifxus_tell_slob ( long 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.

7.43.44 ifxus_write_slob() : Ecrit une chaîne dans un objet SLOB

```
int ifxus_write_slob ( long 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.

7.44 InterBase

7.44.1 Introduction

Interbase est une base de données populaire, créée par Borland/Inprise. Pour plus d'informations sur Interbase, allez à <http://www.interbase.com/> . Par ailleurs, Interbase vient de rejoindre le mouvement Open Source!

Note
Le support intégral de InterBase 6 a été ajouté à PHP 4.0.
Cette base de données utilise les guillemets simples (') pour échapper les caractères, un peu comme le fait Sybase. Ajoutez à votre fichier <code>php.ini</code> la directive suivante :
Configuration InterBase
<code>magic_quotes_sybase = On</code>

7.44.2 Pré-requis

7.44.3 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 librairies <code>gds32.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>). 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 prêtes. Par conséquent, vous n'avez pas à copier la librairie <code>gds32.dll</code> .

7.44.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<code>ibase.allow_persistent</code>	<code>"1"</code>	PHP_INI_SYSTEM
<code>ibase.max_persistent</code>	<code>"-1"</code>	PHP_INI_SYSTEM
<code>ibase.max_links</code>	<code>"-1"</code>	PHP_INI_SYSTEM
<code>ibase.default_user</code>	NULL	PHP_INI_ALL
<code>ibase.default_password</code>	NULL	PHP_INI_ALL
<code>ibase.timestampformat</code>	<code>"%m/%d/%Y%H:%M:%S"</code>	PHP_INI_ALL
<code>ibase.dateformat</code>	<code>"%m/%d/%Y"</code>	PHP_INI_ALL
<code>ibase.timeformat</code>	<code>"%H:%M:%S"</code>	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#) .

7.44.5 Types de ressources

7.44.6 Constantes prédefinies

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.

IBASE_DEFAULT (*entier*)
 IBASE_TEXT (*entier*)
 IBASE_UNIXTIME (*entier*)
 IBASE_READ (*entier*)
 IBASE_COMMITTED (*entier*)
 IBASE_CONSISTENCY (*entier*)
 IBASE_NOWAIT (*entier*)
 IBASE_TIMESTAMP (*entier*)
 IBASE_DATE (*entier*)
 IBASE_TIME (*entier*)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [ibase_add_user](#)
- [ibase_affected_rows](#)
- [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_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_modify_user](#)
- [ibase_name_result](#)

- [ibase_num_fields](#)
- [ibase_num_params](#)
- [ibase_param_info](#)
- [ibase_pconnect](#)
- [ibase_prepare](#)
- [ibase_query](#)
- [ibase_rollback_ret](#)
- [ibase_rollback](#)
- [ibase_set_event_handler](#)
- [ibase_timefmt](#)
- [ibase_trans](#)
- [ibase_wait_event](#)

7.44.8 [ibase_affected_rows\(\)](#) : Retourne le nombre de lignes affectées par la dernière requête iBase

```
int ibase\_affected\_rows ( resource link_identifiant )
```

[ibase_affected_rows](#) retourne le nombre de lignes qui ont été affectées par la dernière requête qui a été exécuté dans le contexte de transaction spécifié par `link_identifiant`. Si `link_identifiant` est une ressource de connexion, sa transaction par défaut est utilisée.

Voir aussi [ibase_query](#) et [ibase_execute](#).

7.44.9 [ibase_blob_add\(\)](#) : Ajoute des données dans un BLOB iBase fraîchement créé

```
bool 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](#). Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [ibase_blob_cancel](#), [ibase_blob_close](#), [ibase_blob_create](#) et [ibase_blob_import](#).

7.44.10 [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](#) si 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](#).

7.44.11 [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](#).

7.44.12 [ibase_blob_create\(\)](#) : Crée un BLOB iBase pour ajouter des données

`resource ibase_blob_create ( resource link_identifiant )`

[ibase_blob_create](#) crée un nouveau BLOB à remplir avec des données, sur la connexion InterBase `link_identifiant`. 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](#).

7.44.13 [ibase_blob_echo\(\)](#) : Affiche le contenu d'un BLOB iBase au navigateur

`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](#).

7.44.14 [ibase_blob_get\(\)](#) : Lit les 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 );
print     ibase_blob_get( $blob_hndl, $blob_data[0] );
?>
```

Cette 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](#).

7.44.15 `ibase_blob_import()` : Créé un BLOB iBase, y copie un fichier et le referme

```
string ibase_blob_import ( resource link_identifiant , resource file_handle )
```

`ibase_blob_import` crée un nouveau BLOB sur la connexion iBase `link_identifiant`, 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.

Voir aussi `ibase_blob_add`, `ibase_blob_cancel`, `ibase_blob_close` et `ibase_blob_create`.

7.44.16 `ibase_blob_info()` : Retourne la taille d'un BLOB iBase et d'autres informations utiles

```
array ibase_blob_info ( string blob_id )
```

`ibase_blob_info` 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.

7.44.17 `ibase_blob_open()` : Ouvre un BLOB iBase pour récupérer des parties de données

```
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`.

7.44.18 `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`.

7.44.19 `ibase_commit_ret()` : Valide une transaction iBase sans la refermer

```
bool ibase_commit_ret ( resource link_identifiant )
```

Si `ibase_commit_ret` est appelée sans argument, elle valide la transaction par défaut de la connexion par défaut. Si l'argument `link_identifiant` est un identifiant de connexion, sa transaction par défaut est validée. Si l'argument `link_identifiant` 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.

7.44.20 `ibase_commit()` : Valide une transaction iBase

```
bool ibase_commit ( resource link_identifiant , int trans_number )
```

`ibase_commit` valide la transaction `trans_number`, qui a été préparée avec `ibase_trans`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.44.21 `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 )
```

`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 du PHP `ibase.default_user` et `ibase.default_password`. `charset` est le jeu de caractères 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 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';
$stmt = ibase_query($dbh, $stmt);
while ($row = ibase_fetch_object($stmt)) {
    print $row->email . "\n";
}
ibase_close($dbh);
?>
```

Note

`buffers` a été ajouté en PHP 4.0.0-RC2.

Note

`dialect` a été ajouté en PHP 4.0.0-RC2. Il n'est opérationnel qu'avec les versions InterBase 6 et plus récentes.

Note

`role` a été ajouté en PHP 4.0.0-RC2. Il n'est opérationnel qu'avec les versions InterBase 5 et plus

récentes.

Voir aussi [ibase_pconnect](#) et [ibase_close](#) .

7.44.22 `ibase_delete_user()` : Efface un utilisateur d'une base de données de sécurité (uniquement pour IB6 ou plus récent)

```
bool ibase_delete_user ( string server , string dba_user_name , string dba_user_password , string user_name )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Voir aussi [ibase_add_user](#) et [ibase_modify_user](#) .

7.44.23 `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](#) .

7.44.24 `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](#) .

7.44.25 `ibase_errmsg()` : Retourne un message d'erreur

```
string ibase_errmsg ( void )
```

[ibase_errmsg](#) retourne une chaîne contenant les messages d'erreurs.

7.44.26 `ibase_execute()` : Exécute une requête iBase préparée

```
resource ibase_execute ( resource query , int bind_args )
```

[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
    $updates = array(
        1 => 'Eric',
        5 => 'Filip',
        7 => 'Larry'
    );
    $query = ibase_prepare("UPDATE FOO SET BAR = ? WHERE BAZ = ?");
    while (list($baz, $bar) = each($updates)) {
        ibase_execute($query, $bar, $baz);
    }
?>
```

7.44.27 [ibase_fetch_assoc\(\)](#) : Récupère une ligne du résultat d'une requête dans un tableau associatif

array **ibase_fetch_assoc** (resource result , int blob_flag)

[ibase_fetch_assoc](#) retourne un tableau associatif qui correspond à la ligne récupérée. Les appels suivants renvoient la ligne suivante dans le jeu de résultats, ou `FALSE` si 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 les accéder avec leurs indices numériques en utilisant [ibase_fetch_row](#) ou en utilisant des alias dans votre requête.

Voir aussi [ibase_query](#), [ibase_fetch_row](#), et [ibase_fetch_object](#).

7.44.28 [ibase_fetch_object\(\)](#) : Lit une ligne dans une base Interbase dans un objet

object **ibase_fetch_object** (resource result_id , int blob_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';
    $sth = ibase_query ($dbh, $stmt);
    while ($row = ibase_fetch_object ($sth)) {
        print $row->email . "\n";
    }
    ibase_close ($dbh);
?>
```

Voir aussi [ibase_fetch_row](#) et [ibase_fetch_assoc](#).

7.44.29 [ibase_fetch_row\(\)](#) : Lit une ligne dans une base Interbase

array **ibase_fetch_row** (resource result_identifiant , int blob_flag)

[ibase_fetch_row](#) retourne la prochaine ligne spécifiée dans le résultat obtenu de [ibase_query](#), ou `FALSE` si 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` si il n'y a plus de ligne.

Voir aussi [ibase_fetch_assoc](#) et [ibase_fetch_row](#).

7.44.30 `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](#)

```
<para>
  <example>
    <programlisting role="php">
<?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](#).

7.44.31 `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](#).

7.44.32 `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.

7.44.33 `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.

7.44.34 `ibase_modify_user()` : Modifie un utilisateur dans une base de données de sécurité (uniquement pour InterBase6 ou plus récent)

`bool ibase_modify_user ( string server , string dba_user_name , string dba_user_password , string user_name , string password , string first_name , string middle_name , string last_name )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Voir aussi `ibase_add_user` et `ibase_delete_user`.

7.44.35 `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`.

7.44.36 `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](#) .

7.44.37 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](#) .

7.44.38 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](#) .

7.44.39 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 )
```

[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-RC2.

Note

`dialect` a été ajouté en PHP 4-RC2. Il n'est opérationnel qu'avec les versions InterBase 6 et plus récentes.

Note

`role` a été ajouté en PHP 4-RC2. 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.

7.44.40 `ibase_prepare()` : Prépare une requête iBase pour lier les paramètres et l'exécuter ultérieurement

resource **ibase_prepare** (resource link_identifiant , string query)

[ibase_prepare](#) prépare une requête pour l'exécuter

7.44.41 `ibase_query()` : Exécute une requête sur une base iBase

resource **ibase_query** (resource link_identifiant , 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';

$stmt = ibase_query($dbh, $stmt) or die(ibase_errmsg());

?>
```

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](#).

7.44.42 `ibase_rollback_ret()` : Annule une transaction sans la refermer

bool **ibase_rollback_ret** (resource link_identifiant)

Si [ibase_rollback_ret](#) est appelée sans argument, elle annule la transaction par défaut du lien par défaut. Si l'argument `link_identifiant` est un identifiant de connexion, la transaction par défaut de cette connexion sera annulée. Si l'argument `link_identifiant` 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.

7.44.43 `ibase_rollback()` : Annule une transaction iBase

```
int ibase_rollback ( int link_identifiant , int trans_number )
```

`ibase_rollback` annule la transaction `trans_number` qui a été préparée avec `ibase_trans`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.44.44 `ibase_set_event_handler()` : Enregistre une fonction de callback sur événement iBase

```
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.

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`.

7.44.45 `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 dates, 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

`columntype` a été ajouté en PHP 4.0. Il n'a aucun sens jusqu'à InterBase version 6 et plus récent.

Note

Une modification incompatible avec l'existant est apparue en PHP 4.0 lorsque la directive PHP `ibase.timeformat` a été renommée en `ibase.timestampformat` et les directives `ibase.dateformat` et `ibase.timeformat` ont été ajoutée, de manière à les adapter à leur fonction.

7.44.46 `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`.

7.44.47 `ibase_wait_event()` : Attend un événement iBase

```
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`.

7.45 Ingres II

7.45.1 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 fait automatiquement à l'envoi de la première requête).

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.

7.45.2 Pré-requis

Pour compiler PHP avec le support Ingres, vous aurez besoin de la librairie ouverte et des fichiers d'entêtes fournis avec Ingres II.

7.45.3 Installation

Pour pouvoir utiliser ces fonctions, vous devez compiler PHP avec le support Ingres, en utilisant l'option `--with-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.

7.45.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<code>ingres.allow_persistent</code>	"1"	<code>PHP_INI_SYSTEM</code>
<code>ingres.max_persistent</code>	"-1"	<code>PHP_INI_SYSTEM</code>
<code>ingres.max_links</code>	"-1"	<code>PHP_INI_SYSTEM</code>
<code>ingres.default_database</code>	NULL	<code>PHP_INI_ALL</code>
<code>ingres.default_user</code>	NULL	<code>PHP_INI_ALL</code>
<code>ingres.default_password</code>	NULL	<code>PHP_INI_ALL</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#) .

7.45.5 Types de ressources

Cette extension ne définit aucune ressource.

7.45.6 Constantes prédefinies

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*)
 INGRES_NUM (*entier*)
 INGRES_BOTH (*entier*)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [ingres_autocommit](#)
- [ingres_close](#)
- [ingres_commit](#)
- [ingres_connect](#)
- [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](#)

7.45.8 ingres_close() : Ferme une connexion à un serveur Ingres

`bool ingres_close ( resource link )`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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`.

7.45.9 ingres_commit() : Valide une transaction

`bool ingres_commit ( resource link )`

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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`.

7.45.10 ingres_connect() : Ouvre une connexion à un serveur Ingres

`resource ingres_connect ( string database , string username , string password )`

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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");
    print("Connexion réussie");
    ingres_close();
?>
```

Voir aussi [ingres_pconnect](#) et [ingres_close](#).

7.45.11 [ingres_fetch_array\(\)](#) : Récupère une ligne de résultat Ingres dans un tableau

array [ingres_fetch_array](#) (int result_type , resource link)

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[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 `II_NUM` pour un tableau à indices numériques, `II_ASSOC` pour un tableau associatif, ou `II_BOTH` (défaut) pour un tableau mixte (accessible selon les 2 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](#) .

7.45.12 [ingres_fetch_object\(\)](#) : Récupère une ligne de résultat dans un objet

object [ingres_fetch_object](#) (int result_type , resource link)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[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 `II_ASSOC`, `II_NUM`, et `II_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](#) .

7.45.13 [ingres_fetch_row\(\)](#) : Récupère une ligne de résultat dans un tableau énuméré.

array [ingres_fetch_row](#) (resource link)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[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 champs 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](#) .

7.45.14 `ingres_field_length()` : Retourne la taille d'un champ

```
int ingres_field_length ( int index , resource link )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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](#) .

7.45.15 `ingres_field_name()` : Retourne le nom d'un champ dans le résultat d'une requête.

```
string ingres_field_name ( int index , resource link )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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](#) .

7.45.16 `ingres_field_nullable()` : Teste si un champ est annulable

```
bool ingres_field_nullable ( int index , resource link )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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`.

7.45.17 `ingres_field_precision()` : Retourne la précision d'un champ

`int ingres_field_precision ( int index , resource link )`

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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`.

7.45.18 `ingres_field_scale()` : Retourne l'échelle d'un champ

`int ingres_field_scale ( int index , resource link )`

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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`.

7.45.19 `ingres_field_type()` : Retourne le type d'un champ dans le résultat d'une requête.

`string ingres_field_type ( int index , resource link )`

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[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](#).

7.45.20 [ingres_num_fields\(\)](#) : Retourne le nombre de champs renvoyés par la dernière requête.

```
int ingres\_num\_fields ( resource link )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[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](#).

7.45.21 [ingres_num_rows\(\)](#) : Retourne le nombre de lignes affectées ou retournées par la dernière requête.

```
int ingres\_num\_rows ( resource link )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

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](#).

7.45.22 `ingres_pconnect()` : Ouvre une connexion persistante à un serveur Ingres.

resource **ingres_pconnect** (string database , string username , string password)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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` .

7.45.23 `ingres_query()` : Envoie une requête SQL à un serveur Ingres II

bool **ingres_query** (string query , resource link)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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](#) .

7.45.24 ingres_rollback() : Annule une transaction

bool **ingres_rollback** (resource link)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[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](#) .

7.46 IRC

7.46.1 Introduction

Avec IRCG, vous pouvez rapidement lire des flûts de données XML avec des milliers d'utilisateurs simultanés. Vous pouvez l'utiliser pour mettre en place rapidement des plate-formes puissantes et extensibles, pour gérer des jeux en ligne ou des salons de discussions. IRCG dispose aussi d'un mode non-streamé, où une application reformate 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 v3, IRCG fonctionne sur ces systèmes d'exploitation :

- AIX
- FreeBSD
- HP-UX
- Irix
- Linux
- Solaris
- Tru64

7.46.2 Installation

Les instructions d'installation détaillées sont disponibles sur le site d'[IRC](#) . Nous nous recommandons vivement d'utiliser le script d'installation fourni.

7.46.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.46.4 Types de ressources

7.46.5 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [ircg_channel_mode](#)
- [ircg_disconnect](#)
- [ircg_fetch_error_msg](#)
- [ircg_get_username](#)
- [ircg_html_encode](#)
- [ircg_ignore_add](#)
- [ircg_ignore_del](#)

- [ircg_is_conn_alive](#)
- [ircg_join](#)
- [ircg_kick](#)
- [ircg_lookup_format_messages](#)
- [ircg_msg](#)
- [ircg_nick](#)
- [ircg_nickname_escape](#)
- [ircg_nickname_unescape](#)
- [ircg_notice](#)
- [ircg_part](#)
- [ircg_pconnect](#)
- [ircg_register_format_messages](#)
- [ircg_set_current](#)
- [ircg_set_file](#)
- [ircg_set_on_die](#)
- [ircg_topic](#)
- [ircg_whois](#)

7.46.7 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](#).

7.46.8 ircg_fetch_error_msg() : Retourne le message d'erreur de la dernière opération IRCG

`array ircg_fetch_error_msg ( resource connection )`

[ircg_fetch_error_msg](#) retourne le message d'erreur de la dernière opération IRCG.

Note

Le code d'erreur est stocké dans le premier élément du tableau, et le message dans le second. Le code d'erreur est équivalent aux codes de réponses définis par la RFC 2812.

Exemple avec [ircg_fetch_error_msg](#)

```
if (!ircg_join($id, "#php")) {
    $error = ircg_fetch_error_msg($id);
    print ("Impossible de rejoindre le canal #php. Code d'erreur : {$error[0]}; Description : {$error[1]}");
}
```

7.46.9 ircg_get_username() : Lit le nom de l'utilisateur de la connexion

`string ircg_get_username ( resource connection )`

[ircg_get_username](#) retourne le nom de l'utilisateur pour la connexion `connection`. [ircg_get_username](#) retourne `FALSE` si `connection` n'est plus valide, ou que la connexion est interrompue.

7.46.10 `ircg_html_encode()` : Prépare l'affichage pour le HTML

```
bool ircg_html_encode ( string html_string )
```

...

Voir aussi

7.46.11 `ircg_ignore_add()` : Ajoute un utilisateur sur la liste des utilisateurs indésirables

```
bool 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](#).

7.46.12 `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](#).

7.46.13 `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 la connexion n'est plus disponible.

7.46.14 `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`.

7.46.15 `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.

7.46.16 `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`.

7.46.17 `ircg_msg()` : Envoie un message à un canal ou un utilisateur

```
bool ircg_msg ( resource connection , string recipient , string message , boolean suppress )
```

`ircg_msg` envoie le message `message` à l'utilisateur ou au canal `recipient`, sur le serveur `connection`. Un destinataire `recipient` commençant par # ou & représente un canal IRC, et sinon, un utilisateur.

En donnant la valeur `TRUE` au paramètre `suppress`, vous éviterez que vos propres messages soient affichés sur votre connexion `connection`.

7.46.18 `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`

7.46.19 `ircg_nickname_escape()` : Prépare les caractères spéciaux d'un nick pour être compatible IRC

```
string ircg_nickname_escape ( string nick )
```

`ircg_nickname_escape` prépare les caractères spéciaux du nick `nick` pour être compatible IRC.

Voir aussi `ircg_nickname_unescape`.

7.46.20 `ircg_nickname_unescape()` : Decodes encoded nickname

```
string ircg_nickname_unescape ( string nick )
```

`ircg_nickname_unescape` retourne la forme originale du nick `nick` qui a été préparé par `ircg_nickname_escape`.

Voir aussi `ircg_nickname_escape`.

7.46.21 ircg_notice() : Envoie une note (notice) à un utilisateur

```
bool ircg_notice ( resource connection , string nick , 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.

7.46.22 ircg_part() : Quitte le canal

```
bool ircg_part ( resource connection , string channel )
```

ircg_part quitte le canal `channel` sur le serveur représenté par `connection`.

7.46.23 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 )
```

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 nick (nom d'utilisateur en 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`.

`ctcp_messages`

`user_settings`

Voir aussi ircg_disconnect, ircg_is_conn_alive et ircg_register_format_messages.

7.46.24 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 statu 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](#).

7.46.25 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](#).

7.46.26 ircg_set_file() : Configure le fichier d'historique de la connexion

```
bool ircg_set_file ( resource connection , string path )
```

[ircg_set_file](#) configure le fichier d'historique dans le dossier `path`, dans lequel tous les affichages de la connexion `connection` seront enregistrés. [ircg_set_file](#) retourne `TRUE` en cas de succès, et `FALSE` sinon.

7.46.27 `ircg_set_on_die()` : Configure les opérations à exécuter lorsque la connexion s'interrompt

```
bool ircg_set_on_die ( resource connection , string host , int port , string data )
```

Dans le cas où la connexion IRCG `connection` se termine, IRCG essaiera de se connecter à l'hôte `host` a avec le port `port` (Note : l'hôte doit être une adresse IPv4, car IRCG ne résoud pas les noms d'hôte, pour des raisons de blocage), puis envoie les données `data` au nouvel hôte, puis attend que l'hôte distant referme la connexion. Cette fonction permet de déclencher un script PHP lors de cette terminaison.

Cette fonctionnalité requiert IRCG 3.

7.46.28 `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` .

7.46.29 `ircg_whois()` : Requier les informations sur un utilisateur

```
bool ircg_whois ( resource connection , string nick )
```

Voir aussi

7.47 Java

7.47.1 Introduction

Il y a deux moyens de connecter PHP et Java : soit en intégrant Java directement dans PHP, ce qui est la solution la plus stable et la plus efficace, ou en intégrant PHP dans un environnement de Servlet Java. La première solution est fournie par cette extension Java, et la dernière par le module SAPI qui s'interface avec un serveur de Servlet.

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 inter-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.

7.47.2 Pré-requis

Vous avez besoin d'une Machine virtuelle Java installée sur votre machine pour utiliser cette extension.

7.47.3 Installation

Les instructions d'installation de cette extension sont fournis dans [php4/ext/java/README](#).

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier les librairies **jvm.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).

7.47.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier [php.ini](#).

Nom	Par défaut	Modifiable
java.class.path	NULL	PHP_INI_ALL
java.home	NULL	PHP_INI_ALL
java.library.path	NULL	PHP_INI_ALL
java.library	JAVAILIB	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [ini_set](#).

7.47.5 Types de ressources

Cette extension ne définit aucune ressource.

7.47.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.47.7 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
print 'Java version=' . $system->getProperty('java.version') . ' <br>';
print 'Java vendor=' . $system->getProperty('java.vendor') . ' <br>';
print 'OS=' . $system->getProperty('os.name') . ' ' .
      $system->getProperty('os.version') . ' on ' .
      $system->getProperty('os.arch') . ' <br>';

// Exemple avec java.util.Date
$formatter = new Java('java.text.SimpleDateFormat',
                      "EEEE, MMMM dd, yyyy 'at' h:mm:ss a zzzz");

print $formatter->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 surchargements de fonctions sont des problèmes épineux, étant donné les différences de type de valeurs entre les deux langages. L'extension Java de PHP utilise une métrique simple mais efficace pour déterminer la meilleure 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 hachage peuvent être interchangeables, et fonctionnent de la même façon. Notez que les tables de hachage 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.

7.47.8 Java Servlet SAPI

L'interface PHP4 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 bugs, 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 prédefinies](#)
- [Exemples](#)
- [Java Servlet SAPI](#)
- [java_last_exception_clear](#)
- [java_last_exception_get](#)

7.47.10 java_last_exception_get() : Lit la dernière exception Java

exception `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 :

Gestionnaire d'exception Java

```
<?php
$stack = new Java('java.util.Stack');
$stack->push(1);

// Cela doit marcher
$result = $stack->pop();
$ex = java_last_exception_get();
if (!$ex) print "$result\n";

// Cela doit échouer (le rapport d'erreurs est supprimé par @)
$result = @$stack->pop();
$ex = java_last_exception_get();
if ($ex) print $ex->toString();

// Efface la dernière exception
java_last_exception_clear();
?>
```

7.48 LDAP

7.48.1 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 écrire accéder à un serveur de dossier à moins que vous ne connaissiez sa structure, de même que vous ne pouvez une base de données sans en connaître les tables et les bases.

Bien plus d'informations sont disponibles aux URL suivantes (en anglais) :

- [Netscape](#)
- [OpenLDAP Project](#)
- [LDAP World](#)

Le SDK Netscape contient un guide du programmeur ([Programmer's Guide](#)) très utile, au format HTML (et en anglais).

7.48.2 Pré-requis

Vous devez télécharger et compiler les librairies clients LDAP, soit celles de University of Michigan [ldap-3.3 package](#), soit le [Netscape Directory SDK 3.0](#) ou [OpenLDAP](#), pour compiler le support LDAP.

7.48.3 Installation

Le support LDAP de PHP n'est pas activé par défaut. Vous devez utiliser l'option de configuration `--with-ldap` lorsque vous compilez PHP.

Note
Note aux utilisateurs Win32
Afin d'activer ce module dans l'environnement Windows, vous devez copier les librairies libsasl.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).

7.48.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
ldap.max_links	"-1"	PHP_INI_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [ini_set](#).

7.48.5 Types de ressources

Cette extension ne définit aucune ressource.

7.48.6 Constantes prédefinies

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_TOWAY_AUTH (*entier*)

7.48.7 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."<p>";

if ($ds) {
    echo "Liaison ...";
    $r=ldap_bind($ds); // this is an "anonymous" bind, typically
    // read-only access
    echo "Le résultat de connexion est ".$r."<p>";

    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."<p>";

    echo "Le nombre d'entrées retournées est ".ldap_count_entries($ds,$sr)."<p>";

    echo "Lecture des entrées ...<p>";
    $info = ldap_get_entries($ds, $sr);
    echo "Données pour ".$info["count"]." entrées:<p>";

    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] . "<p>";
    }

    echo "Fermeture de la connexion";
    ldap_close($ds);
} else {
    echo "<h4>Impossible de se connecter au serveur LDAP.</h4>";
}
?>
```

7.48.8 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 commadnes 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 prédefinies](#)
- [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_search](#)
- [ldap_set_option](#)
- [ldap_set_rebind_proc](#)
- [ldap_sort](#)
- [ldap_start_tls](#)

- [ldap_t61_to_8859](#)
- [ldap_unbind](#)

7.48.10 ldap_add() : Ajoute une entrée dans un dossier LDAP

`bool ldap_add ( resource link_identifiant , string dn , array entry )`

`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. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec. 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";
}
?>
```

7.48.11 ldap_bind() : Authentification au serveur LDAP

`bool ldap_bind ( resource link_identifiant , string bind_rdn , string bind_password )`

`ldap_bind` s'authentifie auprès du serveur LDAP `link_identifiant`, 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 tentative de connexion anonyme est tentée.

Identification avec LDAP

```
<?php
// Eléments d'identification LDAP
$ldaprdn = 'nom d'utilisateur'; // DN ou RDN LDAP
$ldapass = '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, $ldaprtn, $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";
    }
}

?>

```

7.48.12 ldap_close() : Ferme une connexion au serveur LDAP

bool ldap_close (resource link_identifieur)

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

`ldap_close` ferme la connexion au serveur LDAP, représentée par `link_identifieur`.

Cet appel est identique, en interne, à `ldap_unbind`. L'API LDAP utilise l'appel à `ldap_unbind`, alors peut être, devriez vous l'utiliser au lieu de `ldap_close`.

Note

`ldap_close` est un alias de `ldap_unbind`.

7.48.13 ldap_compare() : Compare une entrée avec des valeurs d'attributs

bool ldap_compare (resource link_identifieur , 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.

7.48.14 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.

ldap_connect établit une connexion avec un serveur. Le serveur LDAP situé sur l'hôte ^{hostname} et ^{port}. Les deux arguments sont optionnels. Sans argument, l'identifiant de la dernière connexion ouverte sera retournée. 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}" );

?>
```

7.48.15 ldap_count_entries() : Compte le nombre d'entrées après une recherche

```
int ldap_count_entries ( resource link_identifiant , resource result_identifiant )
```

ldap_count_entries retourne le nombre d'entrées trouvées dans le résultat `result_identifiant`, sur la connexion `link_identifiant`, ou `FALSE` en cas d'erreur.

ldap_count_entries retourne le nombre d'entrées stockées dans le résultat de la recherche précédente. `result_identifiant` représente un résultat interne LDAP.

7.48.16 ldap_delete() : Efface une entrée dans un dossier

```
bool ldap_delete ( resource link_identifiant , 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_identifiant`.

7.48.17 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.

7.48.18 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'erreurs localisés. N'utilisez jamais les messages d'erreur pour identifier une erreur, mais bien les numéros.

Enumérer tous les messages d'erreur LDAP

```
<?php
for($i=0; $i<100; $i++) {
    printf("Erreur numéro $i : %s<br>\n", ldap_err2str($i));
}
?>
```

Voir aussi [ldap_errno](#) et [ldap_error](#).

7.48.19 ldap_errno() : Retourne le numéro d'erreur LDAP de la dernière commande exécutée.

int **ldap_errno** (resource link_identifiant)

[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_identifiant`. 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.

Générer et intercepter une erreur

```
<?php
// Cet exemple contient une erreur, que nous intercepterons.
$ld = ldap_connect("localhost");
$bind = ldap_bind($ld);
// erreur de syntaxe dans l'expression du filtre (errno 87),
// must be "objectclass=*" to work.
$res = @ldap_search($ld, "o=Myorg, c=DE", "objectclass");
if (!$res) {
    printf("LDAP-Errno: %s<br>\n", ldap_errno($ld));
    printf("LDAP-Error: %s<br>\n", ldap_error($ld));
    die("Argh!<br>\n");
}
$info = ldap_get_entries($ld, $res);
printf("%d entrées trouvées.<br>\n", $info["count"]);
?>
```

Voir aussi [ldap_err2str](#) et [ldap_error](#).

7.48.20 ldap_error() : Retourne le message LDAP de la dernière commande LDAP

string **ldap_error** (resource link_identifiant)

[ldap_error](#) retourne un message d'erreur LDAP.

[ldap_error](#) retourne le message d'erreur lié à la connexion `link_identifiant`. 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](#).

7.48.21 ldap_explode_dn() : Sépare les différents composants d'un DN

```
array ldap_explode_dn ( string dn , int with_attrb )
```

[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_attrb` 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_attrb` la valeur de 0 et sinon, donnez lui la valeur de 1.

7.48.22 ldap_first_attribute() : Return first attribute

```
string ldap_first_attribute ( resource link_identfier , resource result_entry_identfier , int ber_identfier )
```

[ldap_first_attribute](#) retourne le premier attribut de l'entrée `result_entry_identfier`, sur le serveur `link_identfier`, 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_identfier`. Les autres attributs sont lus grâce à la fonction [ldap_next_attribute](#), appelée aussi souvent que nécessaire. Le paramètre `ber_identfier` est l'identifiant de pointeur en mémoire locale. Il est passé par référence. Le même paramètre `ber_identfier` est passé à la fonction [ldap_next_attribute](#), qui modifiera ce pointeur.

Voir aussi [ldap_get_attributes](#).

7.48.23 ldap_first_entry() : Retourne la première entrée

```
resource ldap_first_entry ( resource link_identfier , resource result_identfier )
```

[ldap_first_entry](#) retourne la première entrée du résultat `result_identfier`, pour la connexion `link_identfier`, 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. Les entrées suivantes seront lues grâce à la fonction [ldap_next_entry](#), en appelant cette fonction aussi souvent que nécessaire.

Voir aussi [ldap_get_entries](#).

7.48.24 ldap_first_reference() : Return first reference

```
resource ldap_first_reference ( resource link , resource result )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.48.25 ldap_free_result() : Free result memory

`bool ldap_free_result ( resource result_identifieur )`

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_identifieur`. 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.

7.48.26 ldap_get_attributes() : Lit les attributs d'une entrée

`array ldap_get_attributes ( resource link_identifieur , resource result_entry_identifieur )`

`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 cette entrée 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 <code>ldap_get_attributes</code>

<pre>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 valeur de l'attribut return_value["attribute"][0] = première valeur de l'attribut return_value["attribute"][i] = (i+1)-ème valeur de l'attribut</pre>

Affiche la liste des attributs d'une entrée

<pre><?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]."
"; ?></pre>

Voir aussi `ldap_first_attribute` et `ldap_next_attribute`.

7.48.27 ldap_get_dn() : Lit le DN d'une entrée

string **ldap_get_dn** (resource link_identifiant , resource result_entry_identifiant)

ldap_get_dn lit le DN de l'entrée courante dans le résultat de recherche LDAP `result_entry_identifiant` , ou `FALSE` , en cas d'erreur.

ldap_get_dn sert à lire le DN dans un résultat de recherche.

7.48.28 ldap_get_entries() : Lit toutes les entrées du résultat

array **ldap_get_entries** (resource link_identifiant , resource result_identifiant)

ldap_get_entries retourne toutes les entrées du résultat `result_identifiant` , 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 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 au 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-ème entrée
return_value[n]["attribute"][m] = m-ième valeur de la n-ième entrée
```

Voir aussi ldap_first_entry et ldap_next_entry .

7.48.29 ldap_get_option() : Lit/écrit la valeur courante d'une option

bool **ldap_get_option** (resource link_identifiant , int option , mixed retval)

ldap_get_option modifie la valeur de l'option `option` à la valeur `retval` pour la connexion `link_identifiant` . 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

```
// $ds doit être une ressource de connexion valide
if (ldap_get_option($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](#) .

7.48.30 ldap_get_values_len() : Lit toutes les valeurs binaires d'une entrée

```
array ldap_get_values_len ( resource link_identifiant , resource result_entry_identifiant , 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_identifiant` . L'entrée utilisée et l'entrée courante du résultat de recherche LDAP `result_entry_identifiant` . Le nombre de valeurs retournées est disponibles à l'index 'count' du tableau retourné. Les valeurs sont accessibles individuellement, avec les index numérique du tableau. L'indexation commence à 0.

Cette fonction d'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.

7.48.31 ldap_get_values() : Lit toutes les valeurs d'une entrée LDAP

```
array ldap_get_values ( resource link_identifiant , resource result_entry_identifiant , 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_identifiant` , pour la connexion `link_identifiant` . Le nombre de valeurs retournées est disponibles à l'index 'count' du tableau retourné. Les valeurs sont accessibles individuellement, avec les index numérique du tableau. L'indexation commence à 0.

L'utilisation de la fonction `result_entry_identifiant` requiert un résultat de recherche, et doit donc être précédé 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 [ldap_get_values](#)

```

return_value["count"] = nombre de valeur de l'attribut
return_value[0] = première valeur de l'attribut
return_value[i] = n-ième valeur de l'attribut

```

Liste toutes les valeurs de l'attribut "mail" d'une entrée

```
// $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.<p>";

for ($i=0; $i < $values["count"]; $i++)
    echo $values[$i]. "<br>";
```

7.48.32 ldap_list() : Recherche dans un niveau

resource **ldap_list** (resource link_identifiant , 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 les détails de leur utilisation.

Note

Ces paramètres optionnels ont été ajoutés en 4.0.2 : `attrsonly`, `sizelimit`, `timelimit` et `deref`.

Produit une liste de tous les services d'une société avec ldap_list

```
<?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] ;
}
?>
```

Note

Depuis PHP 4.0.5, il est aussi possible de faire des recherches parallèles. Reportez vous à ldap_search pour plus de détails.

7.48.33 ldap_mod_add() : Ajoute un attribut à l'entrée courante

bool **ldap_mod_add** (resource link_identifiant , 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.

7.48.34 ldap_mod_del() : Efface un attribut à l'entrée courante

```
bool ldap_mod_del ( resource link_identifiant , 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.

7.48.35 ldap_mod_replace() : Remplace un attribut dans l'entrée courante

```
bool ldap_mod_replace ( resource link_identifiant , 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.

7.48.36 ldap_modify() : Modifie une entrée LDAP

```
bool ldap_modify ( resource link_identifiant , 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 détaillées dans ldap_add.

7.48.37 ldap_next_attribute() : Lit l'attribut suivant

```
string ldap_next_attribute ( resource link_identifiant , resource result_entry_identifiant , resource ber_identifiant )
```

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_identifiant`. Il est passé par référence à la fonction. Le premier appel à ldap_next_attribute est fait avec le `result_entry_identifiant` retourné par ldap_first_attribute.

Voir aussi ldap_get_attributes.

7.48.38 ldap_next_entry() : Lit la prochaine entrée

```
resource ldap_next_entry ( resource link_identifiant , resource result_entry_identifiant )
```

ldap_next_entry retourne un identifiant d'entrée, pour la prochaine entrée dans le résultat `result_entry_identifiant`. La première entrée a été lue avec la fonction ldap_first_entry. Si 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`.

7.48.39 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.

7.48.40 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.

7.48.41 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.

7.48.42 ldap_read() : Lit une entrée

```
resource ldap_read ( resource link_identifieur , 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 par exemple `"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 [ldap_search](#) pour plus de détails.

7.48.43 ldap_rename() : Modifie le nom d'une entrée

```
bool ldap_rename ( resource link_identifiant , 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 l'ancienne valeur RDN est conservée comme une valeur non distinguée. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note

[ldap_rename](#) ne fonctionne actuellement qu'avec LDAPv3. Vous pouvez être obligé d'utiliser [ldap_set_option](#) 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.

7.48.44 ldap_search() : Recherche sur le serveur LDAP

```
resource ldap_search ( resource link_identifiant , 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 est donc vivement recommandé.

Le quatrième paramètre est un tableau de chaînes PHP avec les attributs demandé, comme 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 retourné 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 seul 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 durée 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 déréféréncés.
- LDAP_DEREF_SEARCHING – les alias sont déréféréncés durant la recherche, mais pas durant la localisation du dossier de recherche.
- LDAP_DEREF_FINDING – les alias sont déréféréncés durant la localisation du dossier de recherche, mais pas durant la recherche.
- LDAP_DEREF_ALWAYS – les alias sont toujours déréféréncés.

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 booléens 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 le 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);

print $info["count"]." entrées trouvées.<p>";
?>
```

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 retourne ^{FALSE}, alors qu'une recherche en parallèle retourne un identifiant.

7.48.45 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** identifiant le contrôle, une valeur optionnelle **value** , et un flag optionnel de criticité (**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.

Set protocol version

```
<?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";
?>
```

Set server controls

```
// $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](#) .

7.48.46 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 , string callback )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.48.47 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.

7.48.48 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.

7.48.49 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.

7.48.50 ldap_unbind() : Déconnecte d'un serveur LDAP

```
bool ldap_unbind ( resource link_identifiant )
```

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

ldap_unbind assure la déconnexion du serveur LDAP.

7.49 Mail

7.49.1 Introduction

La fonction `mail` permet d'envoyer du mail.

7.49.2 Pré-requis

Pour que la fonction `mail` soit disponible, il faut que `PHP` ait access 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`.

7.49.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.49.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
SMTP	"localhost"	PHP_INI_ALL
smtp_port	"25"	PHP_INI_ALL
sendmail_from	NULL	PHP_INI_ALL
sendmail_path	DEFAULT_SENDMAIL_PATH	PHP_INI_SYSTEM

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`smtp string`

Sous Windows seulement : adresse IP ou nom que PHP doit utiliser pour envoyer du mail avec la fonction `mail`.

`smtp 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 de `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.

7.49.5 Types de ressources

Cette extension ne définit aucune ressource.

7.49.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [ezmlm_hash](#)
- [mail](#)

7.49.8 mail() : Envoie de mail

`bool mail ( string to , string subject , string message , string additional_headers , string additional_parameters )`

`mail` poste automatiquement le message `message` à destination de `to` . Les destinataires multiples doivent être séparés par des virgules. Les emails avec pièces jointes ou contenus particuliers (comme les emails en HTML, par exemple), peuvent être réalisés avec cette fonction. Il faut respecter l'encodage MIME. Pour plus de détails, voyez [un article chez Zend](#) et [les classes PEAR MIME](#) .

Les RFC suivantes peuvent aussi se révéler utiles : [RFC 1896](#) , [RFC 2045](#) , [RFC 2046](#) , [RFC 2047](#) , [RFC 2048](#) et [RFC 2049](#) .

`mail` retourne `TRUE` si le mail est envoyé, et `FALSE` sinon.

Envoie de courrier électronique (mail)
<pre><?php mail("maxou@exemple.com", "Mon Sujet", "Ligne 1\nLigne 2\nLigne 3"); ?></pre>

Le quatrième argument passé sera inséré à la fin de l'en-tête. Typiquement, cela permet d'insérer des en-têtes supplémentaires. Les en-têtes multiples doivent être séparées par des virgules.

Note
Sous Windows 32bits, vous devez utiliser <code>\r\n</code> pour séparer les en-têtes. Notez aussi que les en-têtes <code>cc:</code> et <code>bcc:</code> sont sensibles à la casse et doivent être écrits <code>Cc:</code> et <code>Bcc:</code> sous Win32.

Si le cinquième argument `additional_parameters` est fourni, PHP l'utilisera dans son appel du programme d'envoi de courrier électronique. Ceci est pratique pour passer une valeur correcte à l'en-tête `Return-Path`, avec `sendmail`.

Note

Le cinquième paramètre a été ajouté en PHP 4.0.5.

Envoi de eMail avec des en-têtes supplémentaires

```
<?php
mail("personne@exemple.com", "Le sujet", $message,
    "From: webmaster@{$_SERVER['SERVER_NAME']}\r\n"
    ."Reply-To: webmaster@{$_SERVER['SERVER_NAME']}\r\n"
    ."X-Mailer: PHP/" . phpversion());
?>
```

Avec le cinquième paramètre, vous pouvez ajouter d'autres paramètres de ligne de commande qui seront utilisés par le programme d'envoi de courrier. Dans l'exemple ci-dessous, l'en-tête `Return-Path` est correctement paramétré. Normalement, `sendmail` ajoute automatiquement l'en-tête `X-Authentication-Warning` (paramètre `-f`), car l'utilisateur "serveur web" n'est probablement pas un de ses utilisateurs de confiance ("trusted users"). Pour supprimer cette alerte, ajoutez l'utilisateur du serveur web dans la configuration de `sendmail`.

Envoi de eMail avec des en-têtes supplémentaires et un paramètre de ligne de commande supplémentaire

```
<?php
mail("personne@exemple.com", "Le sujet", $message,
    "From: webmaster@{$_SERVER['SERVER_NAME']}", "-fwebmaster@{$_SERVER['SERVER_NAME']}");
?>
```

Vous pouvez aussi utiliser des techniques simples de concaténations de chaînes pour construire des messages complexes :

Envoi de mail complexe avec `mail`

```
<?php
/* destinataire */
$to = "Mary <mary@example.com>" . ", " . " ; // note the comma
$to .= "Kelly <kelly@example.com>";

/* sujet */
$subject = "Anniversaires à souhaiter en Août";

/* message */
$message = "
<html>
<head>
  <title>Rappel des anniversaires d'Août</title>
</head>
<body>
<p>Voici la liste des anniversaires du mois d'Août!</p>
<table>
  <tr>
    <th>Ami</th><th>Jour</th><th>Mois</th><th>Année</th>
  </tr>
  <tr>
    <td>Guillaume</td><td>3</td><td>Août</td><td>1970</td>
  </tr>
  <tr>
    <td>Sandra</td><td>17</td><td>Août</td><td>1973</td>
  </tr>
</table>
</body>
</html>";

/* Pour envoyer du mail au format HTML, vous pouvez configurer le type Content-type. */
$headers = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html; charset=iso-8859-1\r\n";

/* D'autres en-têtes : errors, From cc's, bcc's, etc */
$headers .= "From: Birthday Reminder <birthday@example.com>\r\n";

$headers .= "Cc: birthdayarchive@example.com\r\n";
$headers .= "Bcc: birthdaycheck@example.com\r\n";

/* et hop, à la poste */
mail($to, $subject, $message, $headers);
?>
```

Note
Assurez-vous qu'il n'y ait aucune nouvelle ligne (ou d'autres espaces ou caractères blancs) dans les paramètres <code>to</code> ou <code>subject</code> , car cela peut avoir des effets secondaires irrationnels.
Note
Le paramètre <code>to</code> ne peut pas être une adresse de la forme "Quelqu'un <quidam@exemple.com>". La commande mail ne comprendra pas ce format.
Voir aussi imap_mail .

7.50 Traitement de email

7.50.1 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 retirée de ^{PHP} depuis la version 4.2.0 et elle est disponible dans PECL .

7.50.2 Installation

Ces fonctions ne sont disponibles que si ^{PHP} a été compilé avec l'option `--with-aspell[=DIR]` .

- [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](#)

7.50.4 mailparse_msg_create() : Retourne un analyseur destiné au traitement d'un message

```
int 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.

7.50.5 mailparse_msg_extract_part_file() : Extrait et décode une section de message

```
string mailparse_msg_extract_part_file ( resource rfc2045 , string filename , string 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.

7.50.6 mailparse_msg_extract_part() : Extrait et décode une section de message. Si la fonction de callback n'est pas spécifiée, le contenu sera envoyé à la sortie standard

```
void mailparse_msg_extract_part ( resource rfc2045 , string msgbody , string 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.

7.50.7 mailparse_msg_free() : Détruit un analyseur créé par [mailparse_msg_create](#)

```
void 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.

7.50.8 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.

7.50.9 mailparse_msg_get_part() : Retourne une ressource sur une section d'un message MIME

```
int 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.

7.50.10 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.

7.50.11 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.

7.50.12 mailparse_msg_parse() : Analyse incrémentalement des données dans un buffer

```
void 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.

7.50.13 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.

7.50.14 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.

7.50.15 mailparse_uudecode_all() : Scanne les données du fichier et extrait tous les fichiers encodés qui s'y trouvent. Retourne un tableau contenant la liste des fichiers.

```
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.

7.51 Mathématiques

7.51.1 Introduction

Ces fonctions ne sont capables de manipuler que des entiers ^{integer}, 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 .

7.51.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.51.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.51.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.51.5 Types de ressources

Cette extension ne définit aucune ressource.

7.51.6 Constantes prédefinies

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 prédéfinies](#)
- [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](#)
- [lcg_value](#)
- [log](#)
- [log10](#)
- [log1p](#)
- [max](#)
- [min](#)
- [mt_getrandmax](#)
- [mt_rand](#)
- [mt_srand](#)
- [octdec](#)
- [pi](#)
- [pow](#)
- [rad2deg](#)
- [rand](#)
- [round](#)
- [sin](#)
- [sinh](#)
- [sqrt](#)
- [srand](#)
- [tan](#)
- [tanh](#)

7.51.8 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 $a == \cos(\cos(a))$ pour toute valeur qui soit dans l'intervalle de validité de acos.

Voir aussi acosh, asin et atan.

7.51.9 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.

7.51.10 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 $a == \sin(\sin(a))$ pour toute valeur qui soit dans l'intervalle de validité de asin.

Voir aussi asinh, acos et atan.

7.51.11 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.

7.51.12 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 $a == \tan(\tan(a))$ pour toute valeur qui soit dans l'intervalle de validité de atan.

Voir aussi atanh, asin et acos.

7.51.13 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 π (inclus).

Voir aussi acos et atan.

7.51.14 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.

7.51.15 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
$binary = base_convert($hexadecimal, 16, 2);
?>
```

affichera

```
101000110111001100110100
```

7.51.16 bindec() : Convertit de binaire en décimal

`int 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.

Voir aussi [decbin](#) .

7.51.17 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 [ceil](#)

```
<?php
echo ceil(4.3); // 5
echo ceil(9.999); // 10
?>
```

Voir aussi [floor](#) et [round](#) .

7.51.18 cos() : cosinus

float **cos** (float arg)

[cos](#) retourne le cosinus de `arg` (`arg` en radians).

Voir aussi [acos](#) , [sin](#) [tan](#) et [deg2rad](#) .

7.51.19 cosh() : Cosinus hyperbolic

float **cosh** (float arg)

[cosh](#) retourne le cosinus hyperbolique de `arg` , définit comme $(\exp(\arg) + \exp(-\arg))/2$.

Voir aussi [cos](#) , [acosh](#) , [sin](#) et [tan](#) .

7.51.20 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).

Voir aussi [bindec](#) .

7.51.21 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 2147483647 en décimal, ce qui donnera "7fffffff".

Voir aussi hexdec.

7.51.22 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 2147483647 en décimal, ce qui donnera "17777777777".

Voir aussi octdec.

7.51.23 deg2rad() : Convertit un nombre de degrés en radians

float **deg2rad** (float number)

deg2rad convertit ^{number} de degrés en radians.

Voir aussi rad2deg.

7.51.24 exp() : Exponentielle

float **exp** (float arg)

exp retourne l'exponentielle de ^{arg}, c'est-à-dire e élevé à la puissance ^{arg}.

Voir aussi pow et log.

7.51.25 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.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.51.26 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.

7.51.27 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

```
$x = 5.7;
$y = 1.3;
$r = fmod($x, $y);
// $r égale 0.5, car 4 * 1.3 + 0.5 = 5.7
```

7.51.28 getrandmax() : Plus grande valeur aléatoire possible

^{int} **getrandmax** (^{void})

getrandmax retourne la plus grande valeur aléatoire possible retournée par rand.

Voir aussi rand, srand mt_rand, mt_srand et mt_getrandmax.

7.51.29 hexdec() : Convertit de hexadécimal en décimal

^{int} **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".

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](#) .

7.51.30 hypot() : Retourne le résultat de $\sqrt{\text{num1}^2 + \text{num2}^2}$

float **hypot** (float num1 , float num2)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.51.31 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](#) .

7.51.32 is_infinite() :

bool **is_infinite** (float val)

is_infinite **TRUE** si *val* est infini (positif ou négatif), comme par exemple 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](#) .

7.51.33 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 par exemple, le résultat de $\text{acos}(1.01)$.

Voir aussi [is_finite](#) et [is_infinite](#) .

7.51.34 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 $(2^{31} - 85) \cdot (2^{31} - 249)$).

7.51.35 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 $base$ 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.

7.51.36 log10() : logarithme en base 10

float **log10** (float arg)

log10 retourne le logarithme en base 10 de arg .

7.51.37 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.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.51.38 max() : La plus grande valeur

mixed **max** (number arg1 , number arg2 , number ...) mixed **max** (array numbers , array ...)

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.

Si au moins une valeur est un nombre à virgule flottante, elles seront toutes traitées comme des nombres à virgule flottante, et un nombre à virgule flottante sera retourné. Si aucune valeur n'est un

nombre à virgule flottante, elles seront traitées comme des entiers, et un entier sera retourné.

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`.

7.51.39 min() : La plus petite valeur

mixed **min** (number arg1 , number arg2 , number ...) mixed **min** (array numbers , array ...)

`min` retourne la plus petite valeur numérique parmi les valeurs passées en paramètres.

Dans la première syntaxe, vous devez fournir au moins deux paramètres à la fonction `min`, qui vous retournera la plus petite de deux valeurs. Vous pouvez comparer un nombre illimité de valeurs. Si l'une d'entre elle est indéfinie, `min` échouera.

Dans la seconde variante, `min` retourne la plus petite valeur trouvée dans le tableau `numbers`.

Si au moins une valeur est un nombre à virgule flottante, elles seront toutes traitées comme des nombres à virgule flottante, et un nombre à virgule flottante sera retourné. Si aucune valeur n'est un nombre à virgule flottante, elles seront traitées comme des entiers, et un entier sera retourné.

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('chaine', array(2, 5, 7), 42); // chaine
?>
```

Voir aussi `max` et `count`.

7.51.40 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](#) [rand](#) , [srand](#) et [getrandmax](#) .

7.51.41 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 va produire des nombres utilisables en cryptographie, et qui est 4 fois plus rapide que la fonction standard libc. La "Homepage of the Mersenne Twister " est <http://www.math.keio.ac.jp/~matumoto/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)` .

Note

Depuis ^{PHP} 4.2.0, il n'est 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](#) .

7.51.42 mt_srand() : Initialise une valeur aléatoire (meilleure méthode)

```
void mt_srand ( int seed )
```

[mt_srand](#) initialise une meilleure valeur aléatoire avec `seed` .

Exemple avec [mt_srand](#)

```
// 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, il n'est 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](#) .

7.51.43 octdec() : Convertit d'octal en décimal

```
int octdec ( string octal_string )
```

octdec retourne une chaîne contenant la représentation décimale du nombre `octal_tring`. Le nombre le plus grand qui puisse être converti est 17777777777 en décimal, ce qui donnera "2147483647".

Voir aussi decoct.

7.51.44 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
?>
```

7.51.45 pow() : Expression exponentielle

number **pow** (number base , number exp)

pow retourne `base` élevé à la puissance `exp`. Si possible, pow retourne un `integer`.

Si le calcul ne peut être fait, une alerte sera affichée et pow retournera `FALSE`.

Quelques exemples avec pow

```
<?php
var_dump( pow(2,8) ); // int(256)
echo pow(-1,20); // 1
echo pow(0, 0); // 1

echo pow(-1, 5.5); // error
?>
```

Attention

En PHP 4.0.6 plus ancien, pow 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), pow retournait `NAN`.

Voir aussi exp et sqrt.

7.51.46 rad2deg() : Convertit de radians en degrés

float **rad2deg** (float number)

rad2deg convertit `number` (supposé en radians) en degrés.

Voir aussi deg2rad.

7.51.47 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)`.

Note

Depuis PHP 4.2.0, il n'est 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`.

7.51.48 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 (sa valeur par défaut).

Attention

PHP ne gère par correctement les chaînes telles que `"12 300,2"`, par défaut. Reportez-vous à [la conversion de chaînes](#).

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
?>
```

Note

Le paramètre `precision` est disponible uniquement en PHP 4.

Voir aussi `ceil` et `floor`.

7.51.49 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
print sin(deg2rad(60)); // 0.866025403 ...
print sin(60);          // -0.304810621 ...
```

```
?>
```

Voir aussi [asin](#) , [cos](#) , [tan](#) et [deg2rad](#) .

7.51.50 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](#) .

7.51.51 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](#) .

7.51.52 srand() : Initialise le générateur de nombres aléatoires

void **srand** (int seed)

[srand](#) initialise le générateur de nombres aléatoires avec $seed$.

Exemple avec [srand](#)

```
// 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, il n'est plus besoin d'initialiser le générateur de nombres aléatoires avec [srand](#) ou [mt_srand](#) car c'est fait automatiquement.

Voir aussi [rand](#) , [getrandmax](#) , [mt_rand](#) , [mt_srand](#) et [mt_getrandmax](#) .

7.51.53 tan() : Tangente

float **tan** (float arg)

tan retourne la tangente de ^{arg} (^{arg} en radians).

Voir aussi atan , sin , cos et deg2rad .

7.51.54 tanh() : Tangente hyperbolique

^{float} **tanh** (^{float} ^{arg})

tanh retourne la tangeante hyperbolique de ^{arg} , définie comme ^{sinh(arg)/cosh(arg)} .

Voir aussi tan , atanh , sin et cos .

7.52 Chaînes de caractères multi-octets

7.52.1 Introduction

De nombreuses langues dont les signes ne peuvent pas être exprimés sur un seul octet. Des codes multi-octets sont utilisés pour pallier à cette insuffisance. `mbstring` est développé pour supporter les caractères japonais. Cependant, de nombreuses fonctions `mbstring` peuvent supporter d'autres jeux de caractères.

Les jeux de caractères multi-octets représentent les caractères sur plusieurs octets consécutifs (d'où leur nom). Certains systèmes d'encodages ont des caractères d'échappement dédiés, pour démarrer/finir une séquence de caractères multi-octets. De ce fait, certains caractères peuvent être détruit lorsqu'une chaîne est coupée en plusieurs morceaux, ou bien conduire à des résultats erronés lorsque le nombre de caractère est compté. Il faut utiliser des fonctions qui supportent ces encodages. Les fonctions `mbstring` supportent les jeux de caractères multi-octets, ainsi que les conversions.

Etant donné que PHP supporte essentiellement le jeu de caractères ISO-8859-1, certains jeux de caractères ne fonctionnent pas bien avec PHP. Par conséquent, il est important de donner une valeur à l'option de configuration `mbstring.internal_encoding` qui permettent à PHP de travailler correctement.

Pré-requis PHP 4

- Encodage par octet
- Les caractères d'un octet dans l'intervalle `00h-7fh` doivent être compatibles avec le code `ASCII`
- Jeux de caractères multi-octets, qui n'utilisent pas l'intervalle `00h-7fh`.

Voici des exemples d'encodage internes :

Jeu de caractères qui fonctionnent avec PHP
Jeux de caractères qui fonctionnent avec PHP : ISO-8859-*, EUC-JP, UTF-8
Jeu de caractères qui NE fonctionnent PAS avec PHP : JIS, SJIS

Les jeux de caractères qui ne fonctionnent pas comme encodage interne à PHP, peuvent toutefois être utilisé avec les fonctions de conversion de `mbstring`.

Note
SJIS ne doit pas être utilisé comme encodage interne, à moins que vous ne soyez familier de l'analyseur/compilateur, et des problèmes liés aux jeux de caractères.

Note
Si vous utilisez une base de données avec PHP, il est recommandé que vous utilisiez le même jeu de caractère pour la base de données et le jeu de caractère interne de PHP, pour améliorer les performances.
Si vous utilisez PostgreSQL, il supporte des jeux de caractères qui peuvent être différents de ceux du client. Reportez vous au manuel de PostgreSQL pour plus de détails.

7.52.2 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` : Active les fonctions `mbstring`. Cette option est nécessaire pour utiliser les fonctions `mbstring`.

Note

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 `--with-mbstring=all` est utilisé, toutes les langues ci-dessus seront configurée, mais la taille de PHP sera maximisée, à cause des tables de conversions pour l'Unicode. Notez que le chinois, coréen et russe sont expérimentalement supportés en PHP 4.3.0.

- `--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 venants 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 régulières, compatibles avec les caractères multi-octets.

7.52.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>mbstring.language</code>	NULL	PHP_INI_ALL
<code>mbstring.detect_order</code>	NULL	PHP_INI_ALL
<code>mbstring.http_input</code>	NULL	PHP_INI_ALL
<code>mbstring.http_output</code>	NULL	PHP_INI_ALL
<code>mbstring.internal_encoding</code>	NULL	PHP_INI_ALL
<code>mbstring.script_encoding</code>	NULL	PHP_INI_ALL
<code>mbstring.substitute_character</code>	NULL	PHP_INI_ALL
<code>mbstring.func_overload</code>	"0"	PHP_INI_SYSTEM
<code>mbstring.encoding_translation</code>	"0"	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini set](#).

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 string`

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 reception 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. [mail](#), [ereg](#), etc... sont remplacées par [mb_send_mail](#), [mb_ereg](#), etc... Les valeurs possibles sont 0, 1, 2, 4 ou toute combinaison des quatres. Par exemple, 7 remplacera tout : 0: Aucun remplacement, 1: Remplace [mail](#), 2: Remplace les fonctions str*(), 4: Remplace les fonctions ereg*().

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 utilisé par les navigateurs.

Si `enctype` est mis à `multipart/form-data` dans les formulaires HTML, `mbstring` ne convertira pas les données de POST. L'utilisateur doit les convertir dans le script, si nécessaire.

En général, les navigateurs sont suffisamment intelligent 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.

Configuration de php.ini pour mbstring

```
; Langage par défaut
mbstring.language      = English; Anglais (par défaut)
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
mbstring.http_input      = UTF-8,SJIS,EUC-JP ; Specifie 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

;; Set default substitute character
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

;; Choisi 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

;; Choisi 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

```

7.52.4 Types de ressources

Cette extension ne définit aucune ressource.

7.52.5 Constantes prédéfinies

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)

```

7.52.6 Entrées/Sorties HTTP

La conversion automatique des entrées/sorties HTTP peuvent aussi convertir des données binaires. Les utilisateurs doivent contrôler les conversions, si des données binaires doivent être utilisées via HTTP.

Si l'option `enctype` d'un formulaire HTML vaut `multipart/form-data`, `mbstring` ne convertira pas les données du POST. Dans ce cas, les chaînes de caractères doivent être convertis manuellement.

• Entrée HTTP

Il n'y a pas de moyen 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 <code>php.ini</code>
<pre>;; Désactive la conversion HTTP mbstring.http_input = pass ;; Désactive la conversion HTTP (PHP 4.3.0 ou plus récent) mbstring.encoding_translation = Off</pre>

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 <code>mbstring</code> diffère de celui de PHP3-i18n. Le jeu de caractère est converti avec un buffer de sortie.

Exemple de configuration de <code>mbstring</code> dans <code>php.ini</code>

```
;; Active la conversion de sortie pour toute 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 <code>mbstring</code>

```
<?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');
?>
```

7.52.7 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 UCS-2LE UTF-16 UTF-16BE UTF-16LE UTF-8 UTF-7
 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 7bit 8bit et UTF7-IMAP.

Depuis PHP 4.3.0, les jeux de caractères suivants sont ajoutés, mais restent expérimentaux : EUC-CN, CP936, HZ, EUC-TW, CP950, BIG-5, EUC-KR, UHC (CP949), ISO-2022-KR, Windows-1251 (CP1251), Windows-1252 (CP1252), CP866 et 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 utilisée, elle est remplacée par "ASCII, JIS, UTF-8, EUC-JP, SJIS".

Voir aussi [mb_detect_order](#).

Note

Un jeu de caractère supporté n'est pas forcément un bon choix comme jeu de caractères interne.

7.52.8 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 par exemple [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 par exemple [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 régulières. 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	mail	mb_send_mail
2	strlen	mb_strlen
2	strpos	mb_strpos
2	 strrpos	mb_strrpos
2	substr	mb_substr
4	ereg	mb_ereg
4	eregi	mb_eregi
4	ereg_replace	mb_ereg_replace
4	eregi_replace	mb_eregi_replace
4	split	mb_split

7.52.9 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 utilisé. 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 encodage de caractères définissent des séquences de début/fin pour les sections multi-octets.
- Les bases de données allouent des tailles de stockages différentes de celles utilisées par PHP, même si le même encodage de caractère est utilisé (par exemple, PostGreSQL).
- Le courrier électronique utilise généralement ISO-2022-JP.
- Les sites web en "i-mode" utilisent Shift_JIS.

7.52.10 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

<ftp://ftp.ora.com/pub/examples/nutshell/ujip/doc/cjk.inf>

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [Entrées/Sorties HTTP](#)
- [Jeux de caractères supportés](#)
- [Exploitation des chaînes multi-octets en PHP](#)
- [Cas des caractères japonais](#)
- [Références](#)
- [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_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](#)

7.52.12 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 à convertir.

from-encoding est l'encodage de la chaîne *str* à l'origine. Il sera détecté parmi plusieurs encodage 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](#).

7.52.13 mb_convert_kana() : Convertit entre les différents "kana"

string **mb_convert_kana** (string str , string option , mixed 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 alphabets "zen-kaku" en "han-kaku".
A	Convertit les nombres et alphabets "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, "KV");
?>
```

7.52.14 mb_convert_variables() : Convertit l'encodage de variables

string **mb_convert_variables** (string to-encoding , mixed from-encoding , mixed vars)

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.

`from-encoding` est une liste d'encodages possibles pour les variables `vars`, fourni sous forme d'un tableau ou d'une liste d'encodage, séparés par des virgules. Si `from-coding` est omis, les encodages fournis dans mb_detect_order sont utilisés.

`vars` est une référence sur une variables à convertir. Les chaînes, tableaux et objets sont aussi supportés.

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);
?>
```

7.52.15 mb_decode_mimeheader() : Décode une en-tête MIME

string **mb_decode_mimeheader** (string str)

mb_decode_mimeheader décode l'en-tête MIME `str`, obtenue dans un courrier électronique.

mb_decode_mimeheader retourne la chaîne décodée, encodée au format interne.

Voir aussi mb_encode_mimeheader.

7.52.16 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.

array 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

```
$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 obtenu en entité numérique
```

Voir aussi mb_encode_numericentity.

7.52.17 mb_detect_encoding() : Détecte un encodage

string **mb_detect_encoding** (string str , mixed encoding-list)

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'encodage, sous forme de tableau, ou bien de chaîne, les valeurs étant séparés 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](#) .

7.52.18 mb_detect_order() : Lit/modifie l'ordre de détection des encodages

array **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 failure.

`encoding-list` est un tableau, ou une liste d'encodages séparés par une virgule. La valeur "auto" est automatiquement remplacé 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 affected 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](#)

7.52.19 mb_encode_mimeheader() : Encode une chaîne pour une en-tête MIME

string **mb_encode_mimeheader** (string str , string charset , string transfer-encoding , string linefeed)

`mb_encode_mimeheader` convertit la chaîne `str` en en-tête MIME, et retourne la chaîne encodée.

`charset` est le nom de l'encodage. Par défaut, c'est `ISO-2022-JP`.

`transfer-encoding` est l'encodage de transfert. Il peut être `"B"` (Base64) ou `"Q"` (Quoted-Printable). Par défaut, c'est `"B"`.

`linefeed` est le marqueur de fin de ligne. Par défaut, c'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;
?>
```

Voir aussi `mb_decode_mimeheader`.

7.52.20 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 encodage interne en les codes numériques HTML, puis retourne cette chaîne.

array 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 obtenu 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 numérique */
$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`.

7.52.21 mb_ereg_match() : Expression régulière POSIX pour les chaînes multi-octets

```
bool mb_ereg_match ( string pattern , string string , string option )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`mb_ereg_match` retourne `TRUE` si `string` vérifie l'expression régulière `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.

Note

Cette fonction est supportée par PHP 4.2.0 et plus récent.

Voir aussi `mb_regex_encoding` et `mb_ereg` .

7.52.22 mb_ereg_replace() : Remplace des segments de chaînes, avec le support des expressions régulières mutli-octet

```
string mb_ereg_replace ( string pattern , string replacement , string string , array option )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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 utilisé dans le masque `pattern` .

Des options de recherches peuvent être configurés 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` .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi `mb_regex_encoding` et `mb_eregi_replace` .

7.52.23 mb_ereg_search_getpos() : Retourne l'offset du début du prochain segment repéré par une expression régulière

array mb_ereg_search_getpos (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.

`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` .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi `mb_regex_encoding` et `mb_ereg_search_setpos` .

7.52.24 mb_ereg_search_getregs() : Lit le dernier segment de chaîne multi-octets qui correspond au masque

array mb_ereg_search_getregs (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.

`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` . Si 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` .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi `mb_regex_encoding` et `mb_ereg_search_init` .

7.52.25 mb_ereg_search_init() : Configure les chaînes et les expressions régulières pour le support des caractères multi-octets

array mb_ereg_search_init (string string , string pattern , string option)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`mb_ereg_search_init` configure `string` et `pattern` pour supporter les expressions régulières 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` .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi `mb_regex_encoding` et `mb_ereg_search_regs` .

7.52.26 mb_ereg_search_pos() : Retourne la position et la longueur du segment de chaîne qui vérifie le masque de l'expression régulière

array `mb_ereg_search_pos` (`string` `pattern` , `string` `option`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`mb_ereg_search_pos` retourne un tableau contenant la position et la taille du segment repéré grâce à l'expression régulière. 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 ne l'est pas, la précédente sera utilisée.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par `mb_regex_encoding` .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi `mb_regex_encoding` et `mb_ereg_search_init` .

7.52.27 mb_ereg_search_regs() : Retourne le segment de chaîne trouvé par une expression régulière multi-octets

array `mb_ereg_search_regs` (`string` `pattern` , `string` `option`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`mb_ereg_search_regs` exécute l'expression régulière `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 élément est le contenu de la première parenthèse capturante, le troisième élément est le contenu de

la troisième parenthèse capturante, etc... La fonction retourne ^{FALSE} en cas d'erreur.

Le jeu de caractères internet ou bien le jeu de caractères spécifié par mb_regex_encoding sera utilisé.

Note
Cette fonction est supportée depuis PHP 4.2.0 et plus récent.

Voir aussi mb_regex_encoding et mb_ereg_search_init .

7.52.28 mb_ereg_search_setpos() : Choisit le point de départ de la recherche par expression régulière

array **mb_ereg_search_setpos** (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.

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.

Note
Cette fonction est supportée depuis PHP 4.2.0 et plus récent.

Voir aussi mb_regex_encoding et mb_ereg_search_init .

7.52.29 mb_ereg_search() : Recherche par expression régulière multi-octets

bool **mb_ereg_search** (string pattern , string option)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

mb_ereg_search retourne ^{TRUE} si la chaîne multi-octets correspond au masque d'expression régulière, 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 .

Note
Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi mb_regex_encoding et mb_ereg_search_init .

7.52.30 mb_ereg() : Recherche par expression régulière avec support des caractères multi-octets

```
int mb_ereg ( string pattern , string string , array regs )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`mb_ereg` utilise l'expression régulière `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 est spécifié par `mb_regex_encoding` .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi `mb_regex_encoding` et `mb_eregi`

7.52.31 mb_eregi_replace() : Expression régulière avec support des caractères multi-octets, sans tenir compte de la casse

```
string mb_eregi_replace ( string pattern , string replace , 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.

`mb_eregi_replace` analyse la chaîne `string` avec le masque d'expression régulière `pattern` , puis remplace le texte trouvé par `replacement` . 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 jeu de caractères interne, ou le jeu de caractères est spécifié par `mb_regex_encoding` .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi `mb_regex_encoding` et `mb_ereg_replace` .

7.52.32 mb_eregi() : Regular expression match ignoring case with multibyte support

```
int mb_eregi ( string pattern , string string , array regs )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`mb_eregi` utilise l'expression régulière `pattern` avec le support des caractères multi-octets sur la chaîne `string` . `mb_eregi` ignore la casse dans ses recherches. `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 est spécifié par `mb_regex_encoding` .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi `mb_regex_encoding` et `mb_ereg` .

7.52.33 mb_get_info() : Lit la configuration interne de l'extension mbstring

`string mb_get_info ( 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.

`mb_get_info` lit la configuration interne de l'extension mbstring internal setting.

Si `type` n'est pas spécifié, ou si il vaut "all", un tableau contenant les éléments "internal_encoding", "http_output", "http_input" et "func_overload" sera retourné.

Si `type` est spécifié, et si 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` .

7.52.34 mb_http_input() : Détecte le type d'encodage d'un caractère HTTP

`string 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. 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](#)

7.52.35 mb_http_output() : Lit/modifie l'encodage d'affichage

string **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](#)

7.52.36 mb_internal_encoding() : Lit/modifie l'encodage interne

string **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 <pre><?php /* Utilise l'encodage interne UTF-8 */ mb_internal_encoding("UTF-8"); /* Affiche l'encodage interne courant */ echo mb_internal_encoding(); ?></pre>

Voir aussi [mb_http_input](#), [mb_http_output](#) et [mb_detect_order](#)

7.52.37 mb_language() : Lit/modifie le langage courant

string **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 valies sont : "Japanese" (japonais), "ja" (japonais), "English" (anglais), "en" (anglais) and "uni" (UTF-8). [mb_send_mail](#) utilise cette option pour encoder les emails.

Le langage et sa configuration valle `ISO-2022-JP/Base64` poru le japonais, `UTF-8/Base64` pour l'UTF-8 et `ISO-8859-1/quoted printable` pour l'anglais.

Si `language` est fourni et 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'avait été configuré, `mb_language` retourne `FALSE`.

Voir aussi `mb_send_mail`.

7.52.38 mb_output_handler() : Fonction de traitement des affichages web

`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 Web, dans l'encodage paramétré avec `mb_http_output`.

`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 que des images issues d'un script PHP, vous devez spécifier l'encodage spécial "pass", avec la fonction `mb_http_output`.

Voir aussi `ob_start`.

7.52.39 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, converti 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`.

7.52.40 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));
?>
```

7.52.41 mb_regex_encoding() : Retourne le jeu de caractères courant pour les expressions régulières

string **mb_regex_encoding** (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.

mb_regex_encoding retourne le jeu de caractères courant pour les expressions régulières, 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 régulières. Par défaut, cette option est identique au jeu de caractères interne.

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi mb_internal_encoding et mb_ereg

7.52.42 mb_regex_set_options() : Lit et modifie les options des fonctions d'expression régulières à support de caractères multi-octets

string **mb_regex_set_options** (string 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.

mb_regex_set_options configure les options par défaut avec les nouvelles valeurs contenues dans ^{options}, pour les fonctions d'expressions régulières 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.

Note

Cette fonction est supportée depuis PHP 4.3.0.

Voir aussi [mb_split](#) , [mb_ereg](#) et [mb_eregi](#)

7.52.43 mb_send_mail() : Envoie un mail encodé ISO-2022-JP (mail japonais)

```
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 ISO-2022-JP. [mb_send_mail](#) est une version adaptée de [mail](#) .

`to` est l'adresse de destination du mail. Les adresses multiples peuvent être spécifiées en les séparant par des virgules.

`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`).

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [mail](#) , [mb_encode_mimeheader](#) et [mb_language](#) .

7.52.44 mb_split() : Scinde une chaîne en tableau avec une expression régulière multi-octets

```
array mb_split ( string pattern , string string , int limit )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[mb_split](#) scinde la chaîne multi-octets `string` en utilisant l'expression régulière `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](#) .

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi [mb_regex_encoding](#) et [mb_ereg](#) .

7.52.45 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 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.

mb_strcut soustrait la partie de la chaîne *str* qui compte *length* caractères.

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 .

7.52.46 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).

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 .

7.52.47 mb_strlen() : Retourne la taille d'une chaîne

string **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.

Voir aussi mb_internal_encoding , strlen .

7.52.48 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, il sera utilisé à 0 (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_strpos`, `mb_internal_encoding` et `strpos`.

7.52.49 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`.

7.52.50 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` ; si 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 modifiée par les configurations locales, et il peut convertir tout les caractères qui sont considéré comme alphabétiques comme le c cédille (Ç).

Exemple avec `mb_strtolower`.

```
$str = "Marie A Un Petit Agneau Et Elle L'Aime BEAUCOUP.";
$str = mb_strtolower($str);
print $str;
// Affiche : marie a un petit agneau et elle l'aime beaucoup
```

Voir aussi `strtolower`, `mb_strtoupper` et `mb_convert_case`.

7.52.51 `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` ; si 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 modifiée par les configurations locales, et il peut convertir tout les caractères qui sont considérés comme alphabétiques comme le c cédille (ç).

Exemple avec `mb_strtoupper`.

```
$str = "Marie A Un Petit Agneau Et Elle L'Aime BEAUCOUP.";
$str = mb_strtoupper($str);
print $str;
// Affiche : MARIE A UN PETIT AGNEAU ET ELLE L'AIME BEAUCOUP.
```

Voir aussi `strtoupper`, `mb_strtolower` et `mb_convert_case`.

7.52.52 `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-octet 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_strlen` et `mb_internal_encoding`.

7.52.53 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_detect_encoding` et `mb_send_mail`.

`substrchar` 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 `substrchar` est fourni, `mb_substitute_character` retourne `TRUE` en cas de succès, et `FALSE` en cas d'erreur. Si `substrchar` 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();
?>
```

7.52.54 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
print mb_substr_count("Ceci est un test", "es");
// affiche 2
?>
```

Voir aussi `substr_count`, `mb_strpos` et `mb_substr`.

7.52.55 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.

`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`.

7.53 MCAL

7.53.1 Introduction

MCAL signifie 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 y a des accesseurs bien pratiques pour créer, initialiser et lire des objets événements.

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 plate-formes Windows.

7.53.2 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.

7.53.3 Installation

Après avoir installé la bibliothèque mcal, vous devez compiler ^{PHP} avec l'option `--with-mcal[=DIR]`.

7.53.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.53.5 Types de ressources

Cette extension ne définit aucune ressource.

7.53.6 Constantes prédefinies

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 prédefinies](#)
- [mcal_append_event](#)
- [mcal_close](#)
- [mcal_create_calendar](#)
- [mcal_date_compare](#)
- [mcal_date_valid](#)
- [mcal_day_of_week](#)
- [mcal_day_of_year](#)

- mcaldays_in_month
- mcaldelate_calendar
- mcaldelate_event
- mcald_event_add_attribute
- mcald_event_init
- mcald_event_set_alarm
- mcald_event_set_category
- mcald_event_set_class
- mcald_event_set_description
- mcald_event_set_end
- mcald_event_set_recur_daily
- mcald_event_set_recur_monthly_mday
- mcald_event_set_recur_monthly_wday
- mcald_event_set_recur_none
- mcald_event_set_recur_weekly
- mcald_event_set_recur_yearly
- mcald_event_set_start
- mcald_event_set_title
- mcald_expunge
- mcald_fetch_current_stream_event
- mcald_fetch_event
- mcald_is_leap_year
- mcald_list_alarms
- mcald_list_events
- mcald_next_recurrence
- mcald_open
- mcald_popen
- mcald_rename_calendar
- mcald_reopen
- mcald_snooze
- mcald_store_event
- mcald_time_valid
- mcald_week_of_year

7.53.8 mcal_close() : Ferme une connexion MCAL

```
int mccl close ( int mccl_stream, int flags )
```

mccl_close ferme la connexion `mccl_stream`.

7.53.9 mcal_create_calendar() : Crée un nouveau calendrier

```
string mcal_create_calendar (int stream, string calendar)
```

mcals create_calendar crée un nouveau calendrier nommé `calendar`, sur le serveur MCAL `stream`.

7.53.10 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 )
```

mcaldatecompare compare les deux dates données, et retourne <0, 0, > 0 si a<b, a==b, a>b respectivement.

7.53.11 mcaldatevalid() : Valide une date.

```
int mcaldatevalid ( int year , int month , int day )
```

mcaldatevalid retourne ^{TRUE} si la date (constituée par l'année ^{year}, le mois ^{month} et la date ^{day}) est valide, et ^{FALSE} sinon.

7.53.12 mcaldayofweek() : Le jour de la semaine.

```
int mcaldayofweek ( int year , int month , int day )
```

mcaldayofweek retourne le jour de la semaine, pour la date constituée par l'année ^{year}, le mois ^{month} et la date ^{day}.

7.53.13 mcaldayofyear() : Le jour de l'année.

```
int mcaldayofyear ( int year , int month , int day )
```

mcaldayofyear retourne le numéro de jour dans l'année pour la date constituée par l'année ^{year}, le mois ^{month} et la date ^{day}.

7.53.14 mcaldaysinmonth() : Retourne le nombre de jour dans un mois

```
int mcaldaysinmonth ( int month , int leap_year )
```

mcaldaysinmonth retourne le nombre de jour du mois ^{month}, et prend en compte le fait que l'année est bissextile avec le paramètre ^{leap_year}.

7.53.15 mcaldeletecalendar() : Efface un calendrier

```
string mcaldeletecalendar ( int stream , string calendar )
```

mcaldeletecalendar efface le calendrier ^{calendar} sur le serveur ^{stream}.

7.53.16 mcaldeleteevent() : Efface un événement dans un calendrier MCAL.

```
int mcaldeleteevent ( int mcal_stream , int event_id )
```

mcaldeleteevent efface l'événement d'identifiant ^{uid}.

mcaldeleteevent retourne ^{TRUE}.

7.53.17 `mcald_event_add_attribute()` : Ajoute un attribut et une valeur à la structure globale

```
void mcald_event_add_attribute ( int stream , string attribute , string value )
```

`mcald_event_add_attribute` ajoute l'attribut `attribute` à la structure globale, avec la valeur `value` .

7.53.18 `mcald_event_init()` : Initialise la structure globale d'un flôt MCAL

```
int mcald_event_init ( int stream )
```

`mcald_event_init` initialise la structure globale d'un flôt mcal. Cela remet tous les éléments de la structure à 0, ou à leur valeur par défaut.

`mcald_event_init` retourne `TRUE` .

7.53.19 `mcald_event_set_alarm()` : Fixe l'alarme de la structure globale

```
int mcald_event_set_alarm ( int stream , int alarm )
```

`mcald_event_set_alarm` fixe l'alarme de la structure globale, à un nombre de minutes avant déclenchement.

`mcald_event_set_alarm` retourne `TRUE` .

7.53.20 `mcald_event_set_category()` : Fixe la catégorie de la structure globale

```
int mcald_event_set_category ( int stream , string category )
```

`mcald_event_set_category` fixe la catégorie de la structure globale à la valeur de `category` .

`mcald_event_set_category` retourne `TRUE` .

7.53.21 `mcald_event_set_class()` : Fixe la classe de la structure globale

```
int mcald_event_set_class ( int stream , int class )
```

`mcald_event_set_class` fixe la classe de la structure globale. La classe vaut 0 pour si elle est publique, et 1 si elle est privée.

`mcald_event_set_class` retourne `TRUE` .

7.53.22 `mcald_event_set_description()` : Fixe la description de la structure globale

```
int mcald_event_set_description ( int stream , string description )
```


mcald_event_set_description fixe la catégorie de la structure globale à la valeur de `description` .

mcald_event_set_description retourne `TRUE` .

7.53.23 `mcald_event_set_end()` : Fixe la date de fin de la structure globale.

```
int mcald_event_set_end ( int stream , int year , int month , int day , int hour , int min , int sec )
```

mcald_event_set_end fixe la date de fin de la structure globale.

mcald_event_set_end retourne `TRUE` .

7.53.24 `mcald_event_set_recur_daily()` : Fixe la récurrence quotidienne

```
int mcald_event_set_recur_daily ( int stream , int year , int month , int day , int interval )
```

mcald_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).

7.53.25 `mcald_event_set_recur_monthly_mday()` : Fixe la récurrence

```
int mcald_event_set_recur_monthly_mday ( int stream , int year , int month , int day , int interval )
```

mcald_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).

7.53.26 `mcald_event_set_recur_monthly_wday()` : Fixe la récurrence mensuelle.

```
int mcald_event_set_recur_monthly_wday ( int stream , int year , int month , int day , int interval )
```

mcald_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).

7.53.27 `mcald_event_set_recur_none()` : Supprime la récurrence de la structure globale

```
int mcald_event_set_recur_none ( int stream )
```

mcald_event_set_recur_none supprime la récurrence de la structure globale (event->recur_type est mis à `MCAL_RECUR_NONE`).

7.53.28 `mcald_event_set_recur_weekly()` : Fixe la récurrence hebdomadaire

```
int mcald_event_set_recur_weekly ( int stream , int year , int month , int day , int interval , int weekdays )
```

`mcald_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).

7.53.29 `mcald_event_set_recur_yearly()` : Fixe la récurrence annuelle

```
int mcald_event_set_recur_yearly ( int stream , int year , int month , int day , int interval )
```

`mcald_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).

7.53.30 `mcald_event_set_start()` : Fixe la date de début de la structure globale

```
int mcald_event_set_start ( int stream , int year , int month , int day , int hour , int min , int sec )
```

`mcald_event_set_start` fixe la date de début de la structure globale.

`mcald_event_set_start` retourne `TRUE` .

7.53.31 `mcald_event_set_title()` : Fixe le titre de la structure globale

```
int mcald_event_set_title ( int stream , string title )
```

`mcald_event_set_title` fixe le titre de la structure globale à la valeur de `title` .

`mcald_event_set_title` retourne `TRUE` .

7.53.32 `mcald_expunge()` : Supprime tous les événements marqués pour l'effacement

```
int mcald_expunge ( int stream )
```

`mcald_expunge` supprime tous les événements marqués pour l'effacement sur la connexion `stream` .

7.53.33 `mcald_fetch_current_stream_event()` : Retourne un objet contenant la structure de date

```
object mcald_fetch_current_stream_event ( int stream )
```

`mcald_fetch_current_stream_event` retourne la structure de la date du flôt courant sous la forme d'un objet, qui contient :

- int id – ID de l'événement.
- int public – ^{TRUE} si l'événement est public, ^{FALSE} si 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_enddate – 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

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 jours (i.e., le troisième samedi du mois)
- 5 – Indique que l'événement se répète tous les ans

7.53.34 mcal_fetch_event() : Recherche un événement dans le calendrier

object **mcal_fetch_event** (int mcal_stream , int event_id , int options)

mcal_fetch_event recherche un événement dans le calendrier spécifié par ^{id} .

mcal_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} si 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_enddate – 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

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 jours (i.e., le troisième samedi du mois)
- 5 – Indique que l'événement se répète tous les ans

7.53.35 `mcalf_is_leap_year()` : Vérifie que l'année est bissextile.

```
int mcal_is_leap_year ( int year )
```

`mcalf_is_leap_year` retourne 1 si l'année `year` est bissextile, et 0 sinon.

7.53.36 `mcalf_list_alarms()` : Retourne une liste d'événements qui ont une alarme prévue à une date

```
array mcal_list_alarms ( int mcal_stream , int begin_year , int begin_month , int begin_day , int end_year , int end_month , int end_day )
```

`mcalf_list_events` retourne un tableau d'identifiants d'événements, qui ont une alarme de prévue à la date `alarm_date`. Si seul le flôt MCAL est donné, la date de début et de fin de la structure globale sera utilisée.

`mcalf_list_events` prend une date, et retourne un tableau d'identifiants.

7.53.37 `mcalf_list_events()` : Retourne une liste d'événement entre deux dates

```
array mcal_list_events ( int mcal_stream , object begin_date , object end_date )
```

`mcalf_list_events` retourne un tableau d'identifiants d'événements, compris entre deux dates.

`mcalf_list_events` prend une date de début et une date de fin. Un tableau d'identifiants est retourné.

7.53.38 `mcalf_next_recurrence()` : Retourne la prochaine occurrence d'un événement

```
int mcalf_next_recurrence ( int stream , int weekstart , array next )
```

`mcalf_next_recurrence` retourne un objet contenant la prochaine date de l'événement, ou la date de l'événement suivant la date. `mcalf_next_recurrence` retourne un objet date vide si l'événement n'a pas de réoccurrence, ou si quelque chose est invalide. Utilisez `weekstart` pour déterminer le premier jour.

7.53.39 `mcalf_open()` : Ouvre une connexion MCAL

```
int mcalf_open ( string calendar , string username , string password , int options )
```

`mcalf_open` retourne un flût MCAL en cas de succès, et `FALSE` en cas d'erreur.

`mcalf_open` ouvre une connexion MCAL au serveur `calendar`. Si `options` est spécifié, passe aussi `options` à la boîte aux lettres (???). La structure interne du flût MCAL est initialisée à la connexion.

7.53.40 `mcalf_popen()` : Ouvre une connexion persistante MCAL

```
int mcalf_popen ( string calendar , string username , string password , int options )
```

`mcalf_popen` retourne un flût MCAL en cas de succès, et `FALSE` sinon.

`mcalf_popen` ouvre une connexion MCAL au serveur de calendrier `calendar`. Si les options `options` sont spécifiées, elles sont aussi passé à cette boîte aux lettres (???). La structure interne du flût est aussi initialisée.

7.53.41 `mcalf_rename_calendar()` : Renomme un calendrier

```
string mcalf_rename_calendar ( int stream , string old_name , string new_name )
```

`mcalf_rename_calendar` renomme le calendrier `old_name` en `new_name`.

7.53.42 `mcalf_reopen()` : Réouvre une connexion MCAL

```
int mcalf_reopen ( string calendar , int options )
```

`mcalf_reopen` réouvre une connexion MCAL.

`mcalf_reopen` réouvre une connexion MCAL avec le serveur `calendar`. Si les options `options` sont spécifiées, elles sont aussi passé à cette boîte aux lettres.

7.53.43 `mcalf_snooze()` : Eteind l'alarme d'un événement

```
bool mcalf_snooze ( int stream_id , int event_id )
```

`mcalf_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.

7.53.44 `mcalf_store_event()` : Modifie un événement dans un calendrier MCAL

```
int mcal_store_event ( int mcal_stream )
```

`mcalf_store_event` enregistre l'événement global dans le calendrier MCAL `mcalf_stream`.

`mcalf_store_event` retourne l'identifiant de l'événement modifié en cas de succès, et `FALSE` en cas d'erreur.

7.53.45 `mcalf_time_valid()` : Valide une heure.

```
int mcal_time_valid ( int hour , int minutes , int seconds )
```

`mcalf_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.

7.53.46 `mcalf_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 )
```

7.54 Chiffage mcrypt

7.54.1 Introduction

Ces fonctions permettent d'accéder à la librairie mcrypt, qui dispose d'une grande variété d'algorithmes de chiffage, tels que 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, elle accepte aussi RC6 et IDEA qui sont considérés comme "non-libres".

7.54.2 Pré-requis

Ces fonctions utilisent `mcrypt`. Pour utiliser cette librairie, téléchargez le fichier `libmcrypt-x.x.tar.gz` sur [ce site](#) et suivez les instructions d'installation fournies. Les utilisateurs Windows trouveront toutes librairies nécessaires [ici](#).

Si vous compilez PHP avec la librairie `libmcrypt 2.4.x`, les algorithmes suivants sont supportés : CAST, LOKI97, RIJNDAEL, SAFERPLUS, SERPENT ainsi que les chiffrements suivants : ENIGMA (chiffage), PANAMA, RC4 et WAKE. Avec `libmcrypt 2.4.x` un autre mode de chiffrement est disponible : nOFB.

7.54.3 Installation

Vous devez compiler `PHP` avec l'option `--with-mcrypt` pour activer cette extension.

7.54.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Changeable
<code>mcrypt.algorithms_dir</code>	NULL	PHP_INI_ALL
<code>mcrypt.modes_dir</code>	NULL	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

7.54.5 Types de ressources

Cette extension ne définit aucune ressource.

7.54.6 Constantes prédefinies

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 chiffage (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).

- ECB (electronic codebook) ECB (electronic codebook) est prévu pour des données aléatoires, telles que des clés. Etant donné que les données sont peu nombreuses et aléatoires, les inconvénients de l'ECB ont ici un effet négatif favorable.
- CBC (cipher block chaining) est spécialement pratique avec les fichiers dont la sécurité ECB n'est pas suffisante.
- CFB (cipher feedback) est la meilleure méthode pour chiffrer des flots d'octets, quand les octets doivent être encryptés un par un.
- OFB (output feedback) est comparable à CFB, mais peut être utilisé lorsque des erreurs ne doivent pas être propagées.
- nOFB (output feedback, in nbit) est comparable à OFB, mais plus sûr, car il opère avec la taille de blocs de l'algorithme.
- STREAM est un mode supplémentaire, pour permettre l'utilisation d'algorithmes tels que 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)

```

7.54.7 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 est avec l'API mcrypt-2.2.x API est que vous pouvez accéder au mode de chiffrement depuis PHP avec la constante MCRYPT_ciphername. Avec la librairie 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) by Schneier (ISBN 0-471-11709-9) pour plus de détails sur le sujet).

7.54.8 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 cle est ultra-secrete";
$input = "Rencontrons-nous dans notre place secrete a 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.

Encryption d'une valeur avec TripleDES sous 2.4.x en mode ECB

```
<?php
$key = "Ceci est une vraie cle secrete";
$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`.

- [Introduction](#)
- [Pré-requis](#)

- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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](#)
- [mcrypt_module_is_block_mode](#)
- [mcrypt_module_open](#)
- [mcrypt_module_self_test](#)
- [mcrypt_ofb](#)
- [mdecrypt_generic](#)

7.54.10 mcrypt_cfb() : Chiffre/déchiffre des données en mode CFB

```
string mcrypt_cfb ( int cipher, string key, string data, int mode, string iv ) string mcrypt_cfb ( string
cipher, string key, string data, int mode, string iv )
```

La première syntaxe utilise libmcrypt 2.2.x, et la seconde utilise libmcrypt 2.4.x.

`mcrypt_cfb` ne doit plus être utilisée. Vous pouvez la remplacer par `mcrypt_generic` et `mdecrypt_generic`.

7.54.11 mcrypt_create_iv() : Crée un vecteur d'initialisation à partir d'une source aléatoire.

```
string mcrypt_create_iv ( int size , int source )
```

`mcrypt_create_iv` sert à créer un IV (vecteur d'initialisation).

`mcrypt_create_iv` prend deux arguments, `size` détermine la taille de IV, `source` 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`). Si vous utilisez `MCRYPT_RAND`, assurez-vous de bien appeler `srand` pour initialiser le générateur de nombres aléatoires.

Exemple avec `mcrypt_create_iv`

```
<?php
    $size = mcrypt_get_iv_size (MCRYPT_CAST_256, MCRYPT_MODE_CFB);
    $iv = mcrypt_create_iv ($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).

7.54.12 mcrypt_decrypt() : Déchiffre un texte

```
string mcrypt_decrypt ( string cipher , string key , string data , string mode , string iv )
```

`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 le fournissez par le VI, alors qu'il est nécessaire, la fonction affichera une alerte, et utilise un VI composé de caractères `'\0'`.

7.54.13 `mcrypt_ecb()` : Chiffre/déchiffre des données en mode ECB

```
string mcrypt_ecb ( int cipher , string key , string data , int mode ) string mcrypt_ecb ( string cipher , string
key , string data , int mode , string iv )
```

La première syntaxe utilise libmcrypt 2.2.x, et la seconde utilise libmcrypt 2.4.x.

`mcrypt_ecb` ne doit plus être utilisée. Vous pouvez la remplacer par `mcrypt_generic` et `mdecrypt_generic`.

7.54.14 `mcrypt_enc_get_algorithms_name()` : Retourne le nom de l'algorithme

```
string mcrypt_enc_get_algorithms_name ( resource td )
```

`mcrypt_enc_get_algorithms_name` retourne le nom de l'algorithme désigné par `td`.

Exemple avec `mcrypt_enc_get_algorithms_name`

```
<?php
$td = mcrypt_module_open (MCRYPT_CAST_256, '', MCRYPT_MODE_CFB, '');
echo mcrypt_enc_get_algorithms_name($td). "\n";

$td = mcrypt_module_open ('cast-256', '', MCRYPT_MODE_CFB, '');
echo mcrypt_enc_get_algorithms_name($td). "\n";
?>
```

Affiche :
CAST-256
CAST-256

7.54.15 `mcrypt_enc_get_block_size()` : Retourne la taille de blocs d'un algorithme

```
int mcrypt_enc_get_block_size ( resource td )
```

`mcrypt_enc_get_block_size` retourne la taille de blocs d'un algorithme, en octets.

7.54.16 `mcrypt_enc_get_iv_size()` : Retourne la taille du VI d'un algorithme

```
int mcrypt_enc_get_iv_size ( resource td )
```

`mcrypt_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 stream.

7.54.17 `mcrypt_enc_get_key_size()` : Retourne la taille maximale de la clé pour un mode

```
int mcrypt_enc_get_key_size ( resource td )
```

`mcrypt_enc_get_key_size` retourne la taille maximale de clé acceptée par le mode désigné par `td`, en octets.

7.54.18 mcrypt_enc_get_modes_name() : Retourne le nom du mode

string **mcrypt_enc_get_modes_name** (resource td)

mcrypt_enc_get_modes_name retourne le nom du mode désigné par ^{td} .

Exemple avec mcrypt_enc_get_modes_name

```
<?php
    $td = mcrypt_module_open (MCRYPT_CAST_256, '', MCRYPT_MODE_CFB, '');
    echo mcrypt_enc_get_modes_name($td). "\n";

    $td = mcrypt_module_open ('cast-256', '', 'ecb', '');
    echo mcrypt_enc_get_modes_name($td). "\n";
?>
```

Affiche :
CFB
ECB

7.54.19 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));
?>
```

Affiche :

```
array(3) {
    [0]=>
    int(16)
    [1]=>
    int(24)
    [2]=>
    int(32)
}
```

7.54.20 mcrypt_enc_is_block_algorithm_mode() : Teste le chiffage par blocs d'un mode

bool **mcrypt_enc_is_block_algorithm_mode** (resource td)

mcrypt_enc_is_block_algorithm_mode retourne 1 si ce mode utilise des algorithmes par blocs, et 0 sinon. (i.e. 0 pour stream, et 1 pour cbc, cfb, ofb).

7.54.21 mcrypt_enc_is_block_algorithm() : Teste le chiffage par blocs d'un algorithme

bool **mcrypt_enc_is_block_algorithm** (resource td)

`mcrypt_enc_is_block_algorithm` retourne 1 si l'algorithme utilisé est un algorithme par blocs, et 0 si c'est un algorithme par flôt.

7.54.22 `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 1 si le mode retourne des blocs d'octets, ou bien 0 s'il retourne des octets (par flôt). (i.e. 1 pour cbc et ecb, et 0 pour cfb et stream).

7.54.23 `mcrypt_enc_self_test()` : Teste un module ouvert

`bool 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 0, sinon, 1.

7.54.24 `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 le fournissez par 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";
$cripttext = mcrypt_encrypt (MCRYPT_RIJNDAEL_256, $key, $text, MCRYPT_MODE_ECB, $iv);
echo strlen ($cripttext)."\n";
?>
```

L'exemple ci-dessus affichera :

42

Voir aussi [mdecrypt_module_open](#) .

7.54.25 mdecrypt_generic_deinit() : Prépare le module pour le déchargement

```
bool mdecrypt_generic_deinit ( resource td )
```

[mdecrypt_generic_deinit](#) prépare le module de chiffrement `td` pour le déchargement. Tous les buffers sont vidées, mais le module n'est pas déchargé. Vous devez appeler [mdecrypt_module_close](#) vous même (mais PHP le fera pour vous a la fin du script). [mdecrypt_generic_deinit](#) retourne `FALSE` en cas d'erreur, et `TRUE` sinon.

Pour un exemple, voir [mdecrypt_module_open](#) et l'entrée sur [mdecrypt_generic_init](#) .

7.54.26 mdecrypt_generic_end() : Termine un chiffrage

```
bool mdecrypt_generic_end ( resource td )
```

[mdecrypt_generic_end](#) termine le chiffrage désigné par le pointeur `td` . En fait, elle supprime tous les buffers, et ferme les modules utilisés. Elle retourne `FALSE` en cas d'erreur, et `TRUE` sinon.

7.54.27 mdecrypt_generic_init() : Initialise tous les buffers nécessaires

```
int mdecrypt_generic_init ( resource td , string key , string iv )
```

La taille maximale de la clé doit être celle retournée par [mdecrypt_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 [mdecrypt_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 chiffrage et le déchiffrement. Si vous ne voulez pas l'utiliser, remplissez-le de zéros, mais ce n'est pas recommandé. La fonction retourne (-1) en cas d'erreur.

Vous devez appeler [mdecrypt_generic_init](#) avant chaque appel à [mdecrypt_generic](#) ou [mdecrypt_generic](#) .

7.54.28 mdecrypt_generic() : Chiffre

```
string mdecrypt_generic ( resource td , string data )
```

[mdecrypt_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.

Le gestionnaire de chiffrement `td` doit être initialisée 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](#) .

7.54.29 mcrypt_get_block_size() : Retourne la taille de blocs d'un chiffrement

`int mcrypt_get_block_size ( int cipher ) int mcrypt_get_block_size ( string cipher , string module )`

`mcrypt_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 librairie libmcrypt 2.2.x, le second lorsque ^{PHP} est compilé avec libmcrypt 2.4.x ou 2.5.x.

Cet exemple montre comment utiliser cette fonction lorsque ^{PHP} est compilé avec libmcrypt 2.4.x et 2.5.x.

Exemple avec `mcrypt_get_block_size`

```
<?php
    echo mcrypt_get_block_size ('tripledes', 'ecb');
?>
```

Affiche :

8

Voir aussi `mcrypt_get_key_size` et `mcrypt_encrypt` .

7.54.30 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` , si ce chiffrement n'existe pas.

Exemple avec `mcrypt_get_cipher_name`

```
<?php
$cipher = MCRYPT_TripleDES;
print mcrypt_get_cipher_name($cipher);
?>
```

L'exemple ci-dessus va afficher :

3DES

7.54.31 mcrypt_get_iv_size() : Retourne la taille du VI utilisé par un couple chiffrement/mode

`int mcrypt_get_iv_size ( resource td ) int mcrypt_get_iv_size ( string cipher , string mode )`

La première syntaxe utilise libmcrypt 2.2.x, et la seconde utilise libmcrypt 2.4.x.

`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` qui peut valoir : "ecb", "cbc", "cfb", "ofb", "nofb" ou "stream".

Cet exemple illustre l'utilisation de cette fonction avec les deux bibliothèques libmcrypt 2.4.x et 2.5.x.

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` et `mcrypt_create_iv`.

7.54.32 `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 )`

`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.

Exemple avec `mcrypt_get_block_size`

```
<?php
    echo mcrypt_get_key_size ('tripledes', 'ecb');
?>
```

Affiche :
24

Voir aussi `mcrypt_get_block_size` et `mcrypt_encrypt`.

7.54.33 `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/libmccrypt " .

7.54.34 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
$modes = mcrypt_list_modes ();
foreach ($modes as $mode) {
    echo $mode."<BR>";
}
?>
```

L'exemple ci-dessus va afficher tous les modes supportés dans le dossier " /usr/local/lib/libmccrypt " .

7.54.35 mcrypt_module_close() : Décharge le module de chiffrement

bool **mcrypt_module_close** (resource td)

mcrypt_module_close décharge le module `td` .

Voir mcrypt_module_open .

7.54.36 mcrypt_module_get_algo_block_size() : Retourne la taille de blocs d'un algorithme

int **mcrypt_module_get_algo_block_size** (string algorithm , string lib_dir)

mcrypt_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.

7.54.37 mcrypt_module_get_algo_key_size() : Retourne la taille maximale de clé

int **mcrypt_module_get_algo_key_size** (string algorithm , string lib_dir)

mcrypt_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.

7.54.38 `mcrypt_module_get_supported_key_sizes()` : Retourne un tableau contenant les tailles de clés supportée par un algorithme de chiffrement

```
array mcrypt_module_get_supported_key_sizes ( string algorithm , string lib_dir )
```

`mcrypt_module_get_supported_key_sizes` retourne un tableau contenant les tailles de clés supportée par l'algorithme de chiffrement `algorithm`. Si il retourne un tableau vide, alors toutes les clés entre 1 et `mcrypt_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 `mcrypt_enc_get_supported_key_sizes` qui est utilisés pour ouvrir les modules de chiffrement.

7.54.39 `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.

7.54.40 `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 bloc, sinon retourne 0. Le paramètre optionnel `lib_dir` contient le chemin jusqu'au module de l'algorithme sur le système.

7.54.41 `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 bien un flôt 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.

7.54.42 `mcrypt_module_open()` : Ouvre le module de l'algorithme et le 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_ciphername`. La librairie 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 d'encryption, ou bien `FALSE` en cas d'erreur.

`algorithm_directory` et `mode_directory` servent à repérer les modules d'encryption. 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 librairie 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");
?>
```

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`.

Using `mcrypt_module_open` in encryption

```
<?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`.

7.54.43 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.

7.54.44 `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 utilise libmcrypt 2.4.x.

`mcrypt_ofb` ne doit plus être utilisée. Vous pouvez la remplacer par `mcrypt_generic` et `mdecrypt_generic`.

7.54.45 `mdecrypt_generic()` : Déchiffre

`string mdecrypt_generic ( resource td , string data )`

`mdecrypt_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 `mdecrypt_generic`

```
<?php
/* Données */
$key = 'Ceci est une très longue clé de chiffrement, voire 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 buffers pour le déchiffrement */
    mcrypt_generic_init ($td, $key, $iv);
    $p_t = mdecrypt_generic ($td, $c_t);

    /* Nettoie */
    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 `mcrypt_generic_init` avant que nous ne déchiffrions les données.

Le gestionnaire de déchiffrement doit toujours être initialisé par la fonction `mcrypt_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 `mcrypt_generic_deinit`. Voir `mcrypt_module_open` pour un exemple.

Voir aussi `mcrypt_generic`, `mcrypt_generic_init` et `mcrypt_generic_deinit`.

7.55 MCVE (Paielement)

7.55.1 Introduction

Ces fonctions font l'interface avec l'API MCVE (libmcve), et permettent l'accès direct aux fonctions MCVE depuis les scripts PHP. MCVE est la solution de paiement en ligne de Main Street Softworks pour Linux / Unix (<http://www.mainstreetsoftworks.com/>) qui permet de traiter des cartes de crédits. 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 de PHP, vous pouvez traiter des cartes de crédit directement depuis vos scripts PHP. La référence suivante va vous décrire le processus.

Note

MCVE est la solution de remplacement pour le module de RedHat CCVS. Ils ont conclu un accord avec RedHat à la fin de l'année 2001 pour migrer tous leur clients vers la plateforme MCVE.

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.55.2 Installation

Pour activer le support MCVE de ^{PHP} commencez par vérifier si la librairie LibMCVE est installée. Vous avez alors à configurer PHP avec l'option `--with-mcve`. Si vous utilisez cette option sans autre détails, PHP essaiera de trouver la librairie LibMCVE dans son dossier standard (`/usr/local`). Si MCVE est dans un dossier non-standard, utilisez la forme suivante : `--with-mcve=$mcve_path`, où `$mcve_path` est le chemin menant à votre installation MCVE. Notez que MCVE requiert que `$mcve_path/lib` et `$mcve_path/include` existent, et qu'ils incluent `mcve.h` dans le dossier `include`, et `libmcve.so` et/ou `libmcve.a` dans le dossier `lib`.

Comme MCVE 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 avec PHP, connectez vous au site testbox.mcve.com sur le port 8333 pour le mode IP, ou le port 8444 pour le mode SSL. Utilisez 'vitale' comme nom d'utilisateur et 'test' comme mot de passe. Plus de détails sur ce système de test sont disponibles à <http://www.mcve.com/>.

7.55.3 Voir aussi

Plus de documentation sur l'API PHP de MCVE est disponible à <http://www.mainstreetsoftworks.com/docs/phpapi.pdf>. La documentation éditée par Main Street est complète et est la source primaire d'informations.

7.55.4 Constantes prédefinies

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.

```
MC_TRANTYPE ( integer )
MC_USERNAME ( integer )
```

```

MC_PASSWORD ( integer )
MC_ACCOUNT ( integer )
MC_TRACKDATA ( integer )
MC_EXPDATE ( integer )
MC_STREET ( integer )
MC_ZIP ( integer )
MC_CV ( integer )
MC_COMMENTS ( integer )
MC_CLERKID ( integer )
MC_STATIONID ( integer )
MC_APPRCODE ( integer )
MC_AMOUNT ( integer )
MC_PTRANUM ( integer )
MC_TTID ( integer )
MC_USER ( integer )
MC_PWD ( integer )
MC_ACCT ( integer )
MC_BDATE ( integer )
MC_EDATE ( integer )
MC_BATCH ( integer )
MC_FILE ( integer )
MC_ADMIN ( integer )
MC_AUDITTYPE ( integer )
MC_CUSTOM ( integer )
MC_EXAMOUNT ( integer )
MC_EXCHARGES ( integer )
MC_RATE ( integer )
MC_REENTERNAME ( integer )
MC_RETURNCITY ( integer )
MC_RETURNSTATE ( integer )
MC_RETURNLOCATION ( integer )
MC_PRIORITY ( integer )
MC_INQUIRY ( integer )
MC_CARDTYPES ( integer )
MC_SUB ( integer )
MC_MARKER ( integer )
MC_DEVICETYPE ( integer )
MC_ERRORCODE ( integer )
MC_NEWBATCH ( integer )
MC_CURR ( integer )
MC_DESCMERCH ( integer )
MC_DESCLOC ( integer )
MC_ORIGTYPE ( integer )
MC_PIN ( integer )
MC_VOIDORIGTYPE ( integer )
MC_TIMESTAMP ( integer )
MC_PRIO_HIGH ( integer )
MC_PRIO_NORMAL ( integer )
MC_PRIO_LOW ( integer )
MC_USER_PROC ( integer )
MC_USER_USER ( integer )
MC_USER_PWD ( integer )
MC_USER_INDCODE ( integer )
MC_USER_MERCHID ( integer )

```

```

MC_USER_BANKID ( integer )
MC_USER_TERMID ( integer )
MC_USER_CLIENTNUM ( integer )
MC_USER_STOREID ( integer )
MC_USER_AGENTID ( integer )
MC_USER_CHAINID ( integer )
MC_USER_ZIPCODE ( integer )
MC_USER_TIMEZONE ( integer )
MC_USER_MERCHCAT ( integer )
MC_USER_MERNAME ( integer )
MC_USER_MERCHLOC ( integer )
MC_USER_STATECODE ( integer )
MC_USER_PHONE ( integer )
MC_USER_SUB ( integer )
MC_USER_CARDTYPES ( integer )
MC_USER_MODE ( integer )
MC_USER_VNUMBER ( integer )
MC_USER_ROUTINGID ( integer )
MC_USER_PPPROPERTY ( integer )
MC_USER_PID ( integer )
MC_USER_PIDPWD ( integer )
MC_USER_SMID ( integer )
MC_USER_SMIDPWD ( integer )
MC_USER_USDDIV ( integer )
MC_USER_AUDDIV ( integer )
MC_USER_DKKDIV ( integer )
MC_USER_GBPDIV ( integer )
MC_USER_HKDDIV ( integer )
MC_USER_JPYDIV ( integer )
MC_USER_NZDDIV ( integer )
MC_USER_NOKDIV ( integer )
MC_USER_SGDDIV ( integer )
MC_USER_ZARDIV ( integer )
MC_USER_SEKDIV ( integer )
MC_USER_CHFDIV ( integer )
MC_USER_CADDIV ( integer )
MC_USER_DIVNUM ( integer )
MC_CARD_VISA ( integer )
MC_CARD_MC ( integer )
MC_CARD_AMEX ( integer )
MC_CARD_DISC ( integer )
MC_CARD_JCB ( integer )
MC_CARD_CB ( integer )
MC_CARD_DC ( integer )
MC_CARD_GIFT ( integer )
MC_CARD_OTHER ( integer )
MC_CARD_ALL ( integer )
MC_MODE_AUTH ( integer )
MC_MODE_SETTLE ( integer )
MC_MODE_BOTH ( integer )
MC_MODE_ALL ( integer )
MC_EXCHARGES_REST ( integer )
MC_EXCHARGES_GIFT ( integer )
MC_EXCHARGES_MINI ( integer )

```



```

MC_EXCHARGES_TELE ( integer )
MC_EXCHARGES_OTHER ( integer )
MC_EXCHARGES_LAUND ( integer )
MC_EXCHARGES_NONE ( integer )
MC_EXCHARGES_GAS ( integer )
MC_EXCHARGES_MILE ( integer )
MC_EXCHARGES_LATE ( integer )
MC_EXCHARGES_1WAY ( integer )
MC_EXCHARGES_VIOL ( integer )
MC_TRAN_SALE ( integer )
MC_TRAN_REDEMPTION ( integer )
MC_TRAN_PREAUTH ( integer )
MC_TRAN_VOID ( integer )
MC_TRAN_PREAUTHCOMPLETE ( integer )
MC_TRAN_FORCE ( integer )
MC_TRAN_OVERRIDE ( integer )
MC_TRAN_RETURN ( integer )
MC_TRAN_RELOAD ( integer )
MC_TRAN_CREDIT ( integer )
MC_TRAN_SETTLE ( integer )
MC_TRAN_INCREMENTAL ( integer )
MC_TRAN_REVERSAL ( integer )
MC_TRAN_ACTIVATE ( integer )
MC_TRAN_BALANCEINQ ( integer )
MC_TRAN_CASHOUT ( integer )
MC_TRAN_TOREVERSAL ( integer )
MC_TRAN_SETTLERFR ( integer )
MC_TRAN_ISSUE ( integer )
MC_TRAN_TIP ( integer )
MC_TRAN_MERCHRETURN ( integer )
MC_TRAN_IVRREQ ( integer )
MC_TRAN_IVRRESP ( integer )
MC_TRAN_ADMIN ( integer )
MC_TRAN_PING ( integer )
MC_TRAN_CHKPWD ( integer )
MC_TRAN_CHNGPWD ( integer )
MC_TRAN_LISTSTATS ( integer )
MC_TRAN_LISTUSERS ( integer )
MC_TRAN_GETUSERINFO ( integer )
MC_TRAN_ADDUSER ( integer )
MC_TRAN_EDITUSER ( integer )
MC_TRAN_DELUSER ( integer )
MC_TRAN_ENABLEUSER ( integer )
MC_TRAN_DISABLEUSER ( integer )
MC_TRAN_IMPORT ( integer )
MC_TRAN_EXPORT ( integer )
MC_TRAN_ERRORLOG ( integer )
MC_TRAN_CLEARERRORLOG ( integer )
MC_TRAN_GETSUBACCTS ( integer )
MC_ADMIN_GUT ( integer )
MC_ADMIN_GL ( integer )
MC_ADMIN_GFT ( integer )
MC_ADMIN_BT ( integer )
MC_ADMIN_UB ( integer )

```

```

MC_ADMIN_QC ( integer )
MC_ADMIN_RS ( integer )
MC_ADMIN_CTH ( integer )
MC_ADMIN_CFH ( integer )
MC_ADMIN_FORCESETTLE ( integer )
MC_ADMIN_SETBATCHNUM ( integer )
MC_ADMIN_RENUMBERBATCH ( integer )
MC_ADMIN_FIELDEDIT ( integer )
MC_ADMIN_CLOSEBATCH ( integer )
MCVE_UNUSED ( integer )
MCVE_NEW ( integer )
MCVE_PENDING ( integer )
MCVE_DONE ( integer )
MCVE_GOOD ( integer )
MCVE_BAD ( integer )
MCVE_STREET ( integer )
MCVE_ZIP ( integer )
MCVE_UNKNOWN ( integer )
MCVE_ERROR ( integer )
MCVE_FAIL ( integer )
MCVE_SUCCESS ( integer )
MCVE_AUTH ( integer )
MCVE_DENY ( integer )
MCVE_CALL ( integer )
MCVE_DUPL ( integer )
MCVE_PKUP ( integer )
MCVE_RETRY ( integer )
MCVE_SETUP ( integer )
MCVE_TIMEOUT ( integer )
MCVE_SALE ( integer )
MCVE_PREAUTH ( integer )
MCVE_FORCE ( integer )
MCVE_OVERRIDE ( integer )
MCVE_RETURN ( integer )
MCVE_SETTLE ( integer )
MCVE_PROC ( integer )
MCVE_USER ( integer )
MCVE_PWD ( integer )
MCVE_INDCODE ( integer )
MCVE_MERCHID ( integer )
MCVE_BANKID ( integer )
MCVE_TERMID ( integer )
MCVE_CLIENTNUM ( integer )
MCVE_STOREID ( integer )
MCVE_AGENTID ( integer )
MCVE_CHAINID ( integer )
MCVE_ZIPCODE ( integer )
MCVE_TIMEZONE ( integer )
MCVE_MERCHCAT ( integer )
MCVE_MERNAME ( integer )
MCVE_MERCHLOC ( integer )
MCVE_STATECODE ( integer )
MCVE_SERVICEPHONE ( integer )

```

- [Introduction](#)
- [Installation](#)
- [Voir aussi](#)
- [Constantes prédefinies](#)
- [mcve_adduser](#)
- [mcve_adduserarg](#)
- [mcve_bt](#)
- [mcve_checkstatus](#)
- [mcve_chkpwd](#)
- [mcve_chngpwd](#)
- [mcve_completeauthorizations](#)
- [mcve_connect](#)
- [mcve_connectionerror](#)
- [mcve_deleterespone](#)
- [mcve_deletetrans](#)
- [mcve_deleteusersetup](#)
- [mcve_deluser](#)
- [mcve_destroyconn](#)
- [mcve_destroyengine](#)
- [mcve_disableuser](#)
- [mcve_edituser](#)
- [mcve_enableuser](#)
- [mcve_force](#)
- [mcve_getcell](#)
- [mcve_getcellbynum](#)
- [mcve_getcommadelimited](#)
- [mcve_getheader](#)
- [mcve_getuserarg](#)
- [mcve_getuserparam](#)
- [mcve_gft](#)
- [mcve_gl](#)
- [mcve_gut](#)
- [mcve_initconn](#)
- [mcve_initengine](#)
- [mcve_initusersetup](#)
- [mcve_iscommadelimited](#)
- [mcve_liststats](#)
- [mcve_listusers](#)
- [mcve_maxconntimeout](#)
- [mcve_monitor](#)
- [mcve_numcolumns](#)
- [mcve_numrows](#)
- [mcve_override](#)
- [mcve_parsecommadelimited](#)
- [mcve_ping](#)
- [mcve_preauth](#)
- [mcve_preauthcompletion](#)
- [mcve_qc](#)
- [mcve_responseparam](#)
- [mcve_return](#)
- [mcve_returncode](#)
- [mcve_returnstatus](#)
- [mcve_sale](#)
- [mcve_setblocking](#)

- [mcve_setdropfile](#)
- [mcve_setip](#)
- [mcve_setssl_files](#)
- [mcve_setssl](#)
- [mcve_settimeout](#)
- [mcve_settle](#)
- [mcve_text_avs](#)
- [mcve_text_code](#)
- [mcve_text_cv](#)
- [mcve_transactionauth](#)
- [mcve_transactionavs](#)
- [mcve_transactionbatch](#)
- [mcve_transactioncv](#)
- [mcve_transactionid](#)
- [mcve_transactionitem](#)
- [mcve_transactionssent](#)
- [mcve_transactiontext](#)
- [mcve_transinqueue](#)
- [mcve_transnew](#)
- [mcve_transparam](#)
- [mcve_transsend](#)
- [mcve_ub](#)
- [mcve_uwait](#)
- [mcve_verifyconnection](#)
- [mcve_verifysslcert](#)
- [mcve_void](#)

7.55.6 mcve_adduserarg() : Ajoute une valeur dans une structure de configuration d'utilisateur

```
int mcve_adduserarg ( resource usersetup , int argtype , string argval )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.7 mcve_bt() : Lit le total des paiements non finalisés en mode batch

```
int mcve_bt ( resource conn , string username , string password )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.8 mcve_checkstatus() : Vérifie si une transaction a été complétée

```
int mcve_checkstatus ( resource conn , int identifier )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.9 mcve_chkpwd() : Verifie le mot de passe

```
int mcve_chkpwd ( resource conn , string username , string password )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.10 mcve_chngpwd() : Modifie le mot de passe administrateur

```
int mcve_chngpwd ( resource conn , string admin_password , string new_password )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.11 mcve_completeauthorizations() : Retourne le nombre de transactions dans la queue d'autorisation, ainsi que la liste des identifiants

```
int mcve_completeauthorizations ( resource conn , int &array )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.12 mcve_connect() : Etablit la connexion avec le système MCVE

```
int mcve_connect ( resource conn )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.13 mcve_connectionerror() : Lit un message représentant la raison de l'échec de la connexion

```
string mcve_connectionerror ( resource conn )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.14 mcve_deleterespone() : Efface une transaction spécifiée dans une structure MCVE_CONN

```
bool mcve_deleterespone ( resource conn , int identifieur )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.15 mcve_deletetrans() : Efface une transaction spécifiée dans une structure MCVE_CONN

```
bool mcve_deletetrans ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.16 mcve_deleteusersetup() : Supprime une structure usersetup

```
void mcve_deleteusersetup ( resource usersetup )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.17 mcve_deluser() : Efface un compte d'utilisateur MCVE

```
int mcve_deluser ( resource conn , string admin_password , string username )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.18 mcve_destroyconn() : Termine une connexion MCVE_CONN

```
void mcve_destroyconn ( resource conn )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.19 mcve_destroyengine() : Libère la mémoire associée à la connectivité IP/SSL

```
void mcve_destroyengine ( void )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.20 mcve_disableuser() : Désactive un utilisateur MCVE actif

```
int mcve_disableuser ( resource conn , string admin_password , string username )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.21 mcve_edituser() : Modifie la structure usersetup d'un utilisateur

```
int mcve_edituser ( resource conn , string admin_password , int usersetup )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.22 mcve_enableuser() : Active un utilisateur MCVE inactif

```
int mcve_enableuser ( resource conn , string admin_password , string username )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

**7.55.23 mcve_force() : Envoie une commande FORCE à MCVE.
(généralement une autorisation par téléphone)**

```
int mcve_force ( resource conn , string username , string password , string trackdata , string account , string expdate ,  
float amount , string authcode , string comments , string clerkid , string stationid , int ptrannum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

**7.55.24 mcve_getcell() : Lit une cellule spécifique dans une réponse au
format CSV, en utilisant le nom de colonne**

```
string mcve_getcell ( resource conn , int identifier , string column , int row )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

**7.55.25 mcve_getcellbynum() : Lit une cellule spécifique dans une
réponse au format CSV, en utilisant le numéro de colonne**

```
string mcve_getcellbynum ( resource conn , int identifier , int column , int row )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

**7.55.26 mcve_getcommadelimited() : Lit une ligne brute, au format CSV
retournée par MCVE**

```
string mcve_getcommadelimited ( resource conn , int identifier )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.27 mcve_getheader() : Lit le nom d'une colonne dans une réponse au format CSV

```
string mcve_getheader ( resource conn , int identifier , int column_num )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.28 mcve_getuserarg() : Lit une valeur dans une structure usersetup

```
string mcve_getuserarg ( resource usersetup , int argtype )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.29 mcve_getuserparam() : Lit un paramètre de réponse utilisateur

```
string mcve_getuserparam ( resource conn , long identifier , int key )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.30 mcve_gft() : Audite MCVE pour les transactions échouées

```
int mcve_gft ( resource conn , string username , string password , int type , string account , string clerkid , string stationid , string comments , int ptrannum , string startdate , string enddate )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.31 mcve_gl() : Audite MCVE pour les transactions finalisées

```
int mcve_gl ( int conn , string username , string password , int type , string account , string batch , string clerkid , string stationid , string comments , int ptrannum , string startdate , string enddate )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.32 mcve_gut() : Audite MCVE pour les transactions non conclues

```
int mcve_gut ( resource conn , string username , string password , int type , string account , string clerkid , string stationid , string comments , int ptrannum , string startdate , string enddate )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.33 mcve_initconn() : Crée et initialise une structure MCVE_CONN

resource mcve_initconn (void)

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.34 mcve_initengine() : Prépare le client pour la connexion IP/SSL

int mcve_initengine (string location)

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.35 mcve_initusersetup() : Initialise une structure pour stocker des données utilisateurs

resource mcve_initusersetup (void)

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.36 mcve_iscommadelimited() : Vérifie qu'une réponse est au format MVCE

int mcve_iscommadelimited (resource conn , int identifier)

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.37 mcve_liststats() : Liste les statistiques de tous les utilisateurs MCVE

int mcve_liststats (resource conn , string admin_password)

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.38 mcve_listusers() : Liste de tous les utilisateurs du système MCVE

int mcve_listusers (resource conn , string admin_password)

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.39 mcve_maxconntimeout() : La durée maximum de tentative de connexion à MCVE

```
bool mcve_maxconntimeout ( resource conn , int secs )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.40 mcve_monitor() : Communique avec MCVE (envoie et reçoit des données) sans blocage

```
int mcve_monitor ( resource conn )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.41 mcve_numcolumns() : Nombre de colonnes retournées dans une ligne au format CSV

```
int mcve_numcolumns ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.42 mcve_numrows() : Nombre de lignes retournées dans une réponse au format CSV

```
int mcve_numrows ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.43 mcve_override() : Envoie une commande OVERRIDE à MCVE

```
int mcve_override ( resource conn , string username , string password , string trackdata , string account , string  
expdate , float amount , string street , string zip , string cv , string comments , string clerkid , string stationid , int  
ptrannum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.44 mcve_parsecommadelimited() : Analyse une au format CSV pour que mcve_getcell et consorts fonctionnent

```
int mcve_parsecommadelimited ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.45 mcve_ping() : Envoie une requête de ping à MCVE

```
int mcve_ping ( resource conn )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.46 mcve_preauth() : Envoie une commande PREAUTHORIZATION à MCVE

```
int mcve_preauth ( resource conn , string username , string password , string trackdata , string account , string
expdate , float amount , string street , string zip , string cv , string comments , string clerkid , string stationid , int
ptrannum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.47 mcve_preauthcompletion() : Complète une PREAUTHORIZATION et la prépare pour la finalisation

```
int mcve_preauthcompletion ( resource conn , string username , string password , float finalamount , int sid , int
ptrannum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.48 mcve_qc() : Audite MCVE pour une liste de transactions dans la queue de sortie

```
int mcve_qc ( resource conn , string username , string password , string clerkid , string stationid , string comments ,
int ptrannum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.49 mcve_responseparam() : Lit un paramètre de réponse personnalisé

```
string mcve_responseparam ( resource conn , long identifiant , string key )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.50 mcve_return() : Emet une commande RETURN ou CREDIT vers MCVE

```
int mcve_return ( int conn , string username , string password , string trackdata , string account , string expdate ,
float amount , string comments , string clerkid , string stationid , int ptrannum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.51 mcve_returncode() : Lit le code retourné exact de la transaction

```
int mcve_returncode ( resource conn , int identifier )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.52 mcve_returnstatus() : Vérifie si la transaction a réussi

```
int mcve_returnstatus ( resource conn , int identifier )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.53 mcve_sale() : Envoie une commande de vente (SALE) à MCVE

```
int mcve_sale ( resource conn , string username , string password , string trackdata , string account , string expdate ,
float amount , string street , string zip , string cv , string comments , string clerkid , string stationid , int ptrannum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.54 mcve_setblocking() : Configure la connexion en mode bloquant/non bloquant

```
int mcve_setblocking ( resource conn , int tf )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.55 mcve_setdropfile() : Configure la connexion en mode Drop-File

```
int mcve_setdropfile ( resource conn , string directory )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.56 mcve_setip() : Configure la connexion en mode IP

```
int mcve_setip ( resource conn , string host , int port )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.57 mcve_setssl_files() : Spécifie le fichier de clé de certificats si le serveur requiert la vérification des certificats du client

```
int mcve_setssl_files ( string sslkeyfile , string sslcertfile )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.58 mcve_setssl() : Configure la connexion en mode SSL

```
int mcve_setssl ( resource conn , string host , int port )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.59 mcve_settimeout() : Configure le temps maximum d'une transaction (par transaction)

```
int mcve_settimeout ( resource conn , int seconds )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.60 mcve_settle() : Emet une résolution pour faire un dépôt batch

```
int mcve_settle ( resource conn , string username , string password , string batch )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.61 mcve_text_avs() : Lit une représentation textuelle de return_avs

```
string mcve_text_avs ( string code )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.62 mcve_text_code() : Lit une représentation textuelle de return_code

```
string mcve_text_code ( string code )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.63 mcve_text_cv() : Lit une représentation textuelle de return_cv

```
string mcve_text_cv ( int code )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.64 mcve_transactionauth() : Lit le numéro d'autorisation de la transaction (alpha-numérique)

```
string mcve_transactionauth ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.65 mcve_transactionavs() : Lit l'adresse de vérification (Address Verification) dans le statut retourné

```
int mcve_transactionavs ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.66 mcve_transactionbatch() : Lit le numéro de batch associé à la transaction

```
int mcve_transactionbatch ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.67 mcve_transactioncv() : Lit le statut de CVC2/CVV2/CID

```
int mcve_transactioncv ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.68 mcve_transactionid() : Lit l'identifiant système unique d'une transaction

```
int mcve_transactionid ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.69 mcve_transactionitem() : Lit le numéro d'ITEM d'une transaction dans un batch

```
int mcve_transactionitem ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.70 mcve_transactionssent() : Vérifie si le buffer de sortie est vide ou pas

```
int mcve_transactionssent ( resource conn )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.71 mcve_transactiontext() : Lit le message retourné par MCVE ou par l'institution de traitement

```
string mcve_transactiontext ( resource conn , int identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.72 mcve_transinqueue() : Nombre de transactions dans la queue client

```
int mcve_transinqueue ( resource conn )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.73 mcve_transnew() : Démarre une nouvelle transaction

```
int mcve_transnew ( resource conn )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.74 mcve_transparam() : Ajoute un paramètre à une transaction

```
int mcve_transparam ( resource conn , long identifiant , int key )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.75 mcve_transsend() : Finalise et envoie une transaction

```
int mcve_transsend ( resource conn , long identifiant )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.76 mcve_ub() : Lit une liste des paiements en attente en mode batch

```
int mcve_ub ( resource conn , string username , string password )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.77 mcve_uwait() : Attend x microsecondes

```
int mcve_uwait ( long microsecs )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.78 mcve_verifyconnection() : Configure MCVE pour qu'il pingue ou pas la connexion, afin d'en vérifier l'existence

```
bool mcve_verifyconnection ( resource conn , int tf )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.79 mcve_verifysslcert() : Configure MCVE pour qu'il vérifie ou pas le certificat SSL

```
bool mcve_verifysslcert ( resource conn , int tf )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.55.80 mcve_void() : Annule (rend VOID) une transaction dans la queue d'attente

```
int mcve_void ( resource conn , string username , string password , int sid , int ptrannum )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.56 Hash

7.56.1 Introduction

Ces fonctions ont été prévues pour fonctionner avec [mhash](#) .

Cet ensemble de fonctions représente une interface avec la librairie mhash. mhash accepte un grand nombre d'algorithmes différents, tels que MD5, SHA1, GOST, 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.

7.56.2 Pré-requis

Pour l'utiliser, téléchargez les distributions de mhash depuis le site [web ici](#) et suivez les instructions d'installation incluses.

7.56.3 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 librairie MHASH.

7.56.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.56.5 Types de ressources

Cette extension ne définit aucune ressource.

7.56.6 Constantes prédefinies

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 doc retarde.

- MHASH_MD5
- MHASH_SHA1
- MHASH_HAVAL256
- MHASH_HAVAL192
- MHASH_HAVAL160
- MHASH_HAVAL128
- MHASH_RIPEMD160
- MHASH_GOST
- MHASH_TIGER
- MHASH_CRC32
- MHASH_CRC32B

7.56.7 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, $input);
print "Le hash vaut ".bin2hex ($hash)."<br />\n";
$hash = mhash (MHASH_MD5, $input, "Jefe");
print "Le hmac vaut ".bin2hex ($hash)."<br />\n";
?>
```

Cet exemple va afficher :

```
Le hmac vaut d03cb659cbf9192dcd066272249f8412
Le hmac vaut 750c783e6ab0b503eaa86e310a5db738
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [mhash_count](#)
- [mhash_get_block_size](#)
- [mhash_get_hash_name](#)
- [mhash_keygen_s2k](#)
- [mhash](#)

7.56.9 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.

7.56.10 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 bien ^{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

7.56.11 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 de créer des clés différentes. Ce grain de sel réservira 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 conseillé pour faire des clés cryptographique, 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 (voir moins). Il est vivement recommandé d'appliquer une fonction de transformation (comme celle-ci), à un mot de passe utilisateur.

7.56.12 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 compatible avec le mode HMAC. En cas d'erreur, la fonction retourne ^{FALSE}.

7.57 Mimetype

7.57.1 Introduction

Ces fonctions essaient de définir le type de contenu et l'encodage d'un fichier en recherchant certaines séquences **magiques**, à certaines positions dans le fichier. Même si ce n'est pas une approche imparable, les heuristiques réalisent un bon travail.

Cette extension est dérivée du module Apache `mod_mime_magic`, qui est lui-même basé sur la commande `file`, entretenue par Ian F. Darwin. Voyez le code source pour plus de détails sur les heuristiques et les informations de copyright.

7.57.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.57.3 Installation

Vous devez compiler `PHP` avec l'option `--enable-mime-magic` pour obtenir le support des fonctions `mime-type`. L'extension nécessite une copie de `magic.mime` tel que distribué avec la commande `file`. Ce fichier fait partie des distributions Linux les plus récentes, et il est généralement stocké dans le dossier `/usr/share/misc`.

Note
L'option de configuration a été changée de <code>--enable-mime-magic</code> en <code>--with-mime-magic</code> depuis <code>PHP 4.3.2</code>

Note
Cette extension n'est pas capable d'utiliser le fichier <code>magic</code> complet, qui est généralement livré avec les distributions Linux, et il est supposé fonctionner avec les plus récentes versions de la commande <code>file</code> .

Note
Note aux utilisateurs Windows
Afin de pouvoir utiliser ce module en environnement Windows, vous devez activer le chemin jusqu'au fichier <code>magic.mime</code> de votre <code>php.ini</code> .
Configuration Windws de <code>magic.mime</code>
<code>mime_magic.magicfile = "\$PHP_INSTALL_DIR\magic.mime"</code>
N'oubliez pas de remplacer la valeur de <code>\$PHP_INSTALL_DIR</code> par votre chemin jusqu'à l'exécutable PHP, dans l'exemple ci-dessus. Par exemple, <code>c:\php</code> .

7.57.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>mime_magic.magicfile</code>	<code>"/usr/share/misc/magic.mime"</code>	<code>PHP_INI_SYSTEM</code>

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [ini_set](#) .

7.57.5 Types de ressources

Cette extension ne définit aucune ressource.

7.57.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [mime_content_type](#)

7.58 Microsoft SQL Server

7.58.1 Introduction

Ces fonctions vous permettent d'accéder aux serveurs de données MS SQL.

Note

L'extension MSSQL n'est disponible que sous Windows 32 bits. Vous pouvez utiliser l'extension Sybase extension pour vous connecter à un serveur MSSQL, depuis d'autres plate-formes.

7.58.2 Pré-requis

Vous avez besoin d'un client MS SQL Client Tools sur votre système, accessible à PHP. Les Client Tools peuvent être installé 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 plate-formes Unix/Linux.

Pour utiliser l'extension MSSQL sur Unix/Linux, vous devez commencer par compiler et installer la librairie FreeTDS. Le code source et les instructions d'installation sont disponibles sur les site de FreeTDS : <http://www.freetds.org/>

7.58.3 Installation

L'extension MSSQL est activée par la ligne `extension=php_mssql.dll` dans le fichier `php.ini`.

7.58.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<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
<code>mssql.timeout</code>	"60"	PHP_INI_ALL
<code>mssql.textsize</code>	"-1"	PHP_INI_ALL
<code>mssql.textlimit</code>	"-1"	PHP_INI_ALL
<code>mssql.batchsize</code>	"0"	PHP_INI_ALL
<code>mssql.datetimeconvert</code>	"1"	PHP_INI_ALL
<code>mssql.secure_connection</code>	"0"	PHP_INI_SYSTEM
<code>mssql.max_procs</code>	"25"	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

7.58.5 Types de ressources

7.58.6 Constantes prédefinies

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 prédefinies](#)
- [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](#)

7.58.8 [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 associé 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](#).

7.58.9 [mssql_connect\(\)](#) : Ouvre une connexion à un serveur MS SQL server

```
int mssql\_connect ( string servername , string username , string password )
```

[mssql_connect](#) retourne un identifiant positif 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 retourné.

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](#).

7.58.10 [mssql_data_seek\(\)](#) : Déplace le pointeur interne de ligne

```
bool mssql\_data\_seek ( resource result_identifiant , 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_identifiant`, jusqu'à la ligne `row_number`. Le prochain appel à [mssql_fetch_row](#) retournera cette ligne.

Voir aussi [mssql_data_seek](#).

7.58.11 `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 qu'elle retourne une valeur, ils seront disponibles après l'appel à la fonction `mssql_execute`, à moins que la procédure stockée ne retourne plusieurs résultats. Dans ce cas, utilisez la fonction `mssql_next_result` 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`.

7.58.12 `mssql_fetch_array()` : Lit une ligne de résultat MSSQL 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 clé.

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**.

Pour plus de détails, voyez `mssql_fetch_row`.

7.58.13 `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.

7.58.14 `mssql_fetch_batch()` : Retourne le prochain lot de lignes MS SQL Server

int `mssql_fetch_batch` (string result_index)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.58.15 `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

Voir aussi `mssql_field_seek` .

7.58.16 `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` si 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 champs, 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 .

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` .

7.58.17 `mssql_fetch_row()` : Lit une ligne de résultat MSSQL 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` si il n'y a plus de lignes à lire.

[mssql_fetch_row](#) lit une ligne dans le résultat [result](#) et place les valeurs dans un tableau. Chaque valeur est enregistré 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 lignes.

Voir aussi [mssql_fetch_array](#), [mssql_fetch_object](#), [mssql_data_seek](#), [mssql_fetch_lengths](#) et [mssql_result](#).

7.58.18 [mssql_field_length\(\)](#) : Lit la longueur d'un champ MS SQL Server

```
int mssql\_field\_length ( resource result , int offset )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.58.19 [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.

7.58.20 [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. Le prochain appel à [mssql_fetch_field](#) qui ne précisera pas de numéro de champ, le champ fixé par [mssql_field_seek](#) sera retournée.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [mssql_fetch_field](#).

7.58.21 [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.

7.58.22 [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 libérés par [mssql_free_result](#).

7.58.23 `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 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`.

7.58.24 `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.

7.58.25 `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.

7.58.26 `mssql_init()` : Initialise une procédure stockée MS SQL Server locale ou distante

```
int 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`

7.58.27 `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.

7.58.28 `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.

7.58.29 `mssql_next_result()` : Déplace le pointeur interne MS SQL Server au le 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);
?>
```

7.58.30 `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`.

7.58.31 `mssql_num_rows()` : Retourne le nombre de lignes dans un résultat MSSQL

```
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`.

7.58.32 `mssql_pconnect()` : Ouvre une connexion persistante à un serveur MS SQL.

```
int 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'en 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'.

7.58.33 `mssql_query()` : Envoie une requête SQL au serveur MSSQL

```
resource mssql_query ( string query , resource link_identifiant , int batch_size )
```

`mssql_query` retourne un identifiant positif de résultat en cas de succès, ou `FALSE` sinon.

`mssql_query` envoie la requête au serveur courant, associé à l'identifiant `link_identifiant` (ou la base par défaut, s'il est omis). Si aucun lien n'est ouvert, `mssql_query` essaiera d'en ouvrir une, en appelant `mssql_connect`.

Voir aussi `mssql_select_db` et `mssql_connect`.

7.58.34 `mssql_result()` : Lit les données d'un résultat

```
string mssql_result ( resource result , int i , 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, ou encore, le nom de la table + "." + le nom du champ, ou encore 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`.

7.58.35 `mssql_rows_affected()` : Retourne le nombre de ligne 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.

7.58.36 `mssql_select_db()` : Sélectionne la base de données MS SQL

```
bool mssql_select_db ( string database_name , resource link_identifier )
```

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 connexion 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.

Exemple avec <code>mssql_select_db</code>

<pre><?php \$conn = mssql_connect('MYSQLSERVER', 'sa', 'password'); mssql_select_db('my data-base', \$conn); ?></pre>

Voir aussi `mssql_connect`, `mssql_pconnect` et `mssql_query`.

7.59 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.

7.59.1 Introduction

Ming n'est pas un acronyme.

Ming est une librairie 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 fournis 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 ajouté en PHP 4.0.5.

7.59.2 Pré-requis

Pour utiliser Ming avec PHP, vous devez d'abord installer la librairie Ming. Le code source et les instructions d'installation sont disponible 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.

7.59.3 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.

7.59.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.59.5 Types de ressources

7.59.6 Constantes prédéfinies

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.

```
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*)

7.59.7 Classes prédefinies

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 prédefinies](#)
- [Classes prédefinies](#)
- [ming_setcubicthreshold](#)
- [ming_setscale](#)
- [ming_useswfversion](#)
- [swfaction](#)
- [swfbitmap->getheight](#)
- [swfbitmap->getwidth](#)
- [swfbitmap](#)
- [swfbutton_keypress](#)
- [swfbutton->addaction](#)
- [swfbutton->addshape](#)
- [swfbutton->setaction](#)
- [swfbutton->setdown](#)
- [swfbutton->sethit](#)
- [swfbutton->setover](#)
- [swfbutton->setup](#)
- [swfbutton](#)
- [swfdisplayitem->addcolor](#)
- [swfdisplayitem->move](#)
- [swfdisplayitem->moveto](#)
- [swfdisplayitem->multicolor](#)
- [swfdisplayitem->remove](#)
- [swfdisplayitem->rotate](#)
- [swfdisplayitem->rotateto](#)
- [swfdisplayitem->scale](#)
- [swfdisplayitem->scaletto](#)
- [swfdisplayitem->setdepth](#)
- [swfdisplayitem->setname](#)
- [swfdisplayitem->setratio](#)
- [swfdisplayitem->skewx](#)

- [swfdisplayitem->skewxto](#)
- [swfdisplayitem->skewy](#)
- [swfdisplayitem->skewyto](#)
- [swfdisplayitem](#)
- [swffill->moveto](#)
- [swffill->rotateto](#)
- [swffill->scaletto](#)
- [swffill->skewxto](#)
- [swffill->skewyto](#)
- [swffill](#)
- [swffont->getwidth](#)
- [swffont](#)
- [swfgradient->addentry](#)
- [swfgradient](#)
- [swfmorph->getshape1](#)
- [swfmorph->getshape2](#)
- [swfmorph](#)
- [swfmovie->add](#)
- [swfmovie->nextframe](#)
- [swfmovie->output](#)
- [swfmovie->remove](#)
- [swfmovie->save](#)
- [swfmovie->setbackground](#)
- [swfmovie->setdimension](#)
- [swfmovie->setframes](#)
- [swfmovie->setrate](#)
- [swfmovie->streammp3](#)
- [swfmovie](#)
- [swfshape->addfill](#)
- [swfshape->drawcurve](#)
- [swfshape->drawcurveto](#)
- [swfshape->drawline](#)
- [swfshape->drawlineto](#)
- [swfshape->movepen](#)
- [swfshape->movepento](#)
- [swfshape->setleftfill](#)
- [swfshape->setline](#)
- [swfshape->setrightfill](#)
- [swfshape](#)
- [swfsprite->add](#)
- [swfsprite->nextframe](#)
- [swfsprite->remove](#)
- [swfsprite->setframes](#)
- [swfsprite](#)
- [swftext->addstring](#)
- [swftext->getwidth](#)
- [swftext->moveto](#)
- [swftext->setcolor](#)
- [swftext->setfont](#)
- [swftext->setheight](#)
- [swftext->setspacing](#)
- [swftext](#)
- [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](#)

7.59.9 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.

7.59.10 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.

7.59.11 swfaction() : Crée une nouvelle action

```
new 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 il 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 :

<i>time()</i>	Retourne le nombre de milli-secondes depuis le début de l'animation.
<i>random(seed)</i>	Retourne un nombre pseudo-aléatoire, entre 0 et seed.
<i>length(expr)</i>	Retourne la taille de l'expression donnée.
<i>int(number)</i>	Retourne le nombre number, arrondi à l'entier inférieur le plus proche.
<i>concat(expr, expr)</i>	Retourne la concaténation des deux expressions.
<i>ord(expr)</i>	Retourne le code ASCII du caractère expr.
<i>chr(num)</i>	Retourne le caractère pour le code ASCII num.
<i>substr(string, location, length)</i>	Retourne la sous-chaîne, extraite de string, de longueur length et commençant au caractère location.

De plus, les commandes suivantes sont accessibles :

<i>duplicateClip(clip, name, depth)</i>	Duplique le sprite nommé clip. La nouvelle animation a le nouveau nom name et la profondeur depth.
<i>removeClip(expr)</i>	Supprime l'animation nommée expr.
<i>trace(expr)</i>	Ecrit l'expression expr dans le fichier de d'historique. Il est peut probable que le navigateur ou le lecteur ne fasse quoi que ce soit avec.
<i>startDrag(target, lock, [left, top, right, bottom])</i>	Comment 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.
<i>stopDrag()</i>	Cesse le déplacement de l'animation.
<i>callFrame(expr)</i>	Appelle l'image expr comme une fonction.
<i>getURL(url, target, [method])</i>	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). L'argument optionnel method peut être post ou get, si vous voulez envoyer des variables au serveur.
<i>loadMovie(url, target)</i>	La même chose que ci-dessus, plus ou moins. En fait, je ne sais pas trop quelle est la différence.
<i>nextFrame()</i>	Va à l'image suivante.
<i>prevFrame()</i>	Va à l'image précédente.
<i>play()</i>	

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'il est supposé faire, mais je ne l'ai jamais testé, 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);
$sp = new SWFSprite();
$i = $sp->add($s);
$i->setDepth(1);
$sp->nextFrame();
for($n=0; $n<5; ++$n)
{
    $i->rotate(-15);
    $sp->nextFrame();
}
$m = new SWFMovie();
$m->setBackground(0xff, 0xff, 0xff);
$m->setDimension(6000,4000);
$i = $m->add($sp);
$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 */
$b = new SWFSprite();
$b->add(new SWFAction("
    stop();
"));
$i = $b->add($r);
for($n=0; $n<=20; ++$n)
{
    $i->setRatio($n/20);
    $b->nextFrame();
}
/* ce conteneur nous permet d'utiliser la même action plusieurs fois */
$c = new SWFSprite();
$i = $c->add($b);
$i->setName('box');
$c->add(new SWFAction("
    setTarget('box');
    /* ...x signifie abscisse du parent, c'est-à-dire (...)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, SWFILL_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();

```

Cet exemple simple gère le clavier (vous devrez cependant cliquer dans la fenêtre pour lui donner le focus, et vous devrez aussi laisser votre souris dans la fenêtre. Si vous savez comment faire cela automatiquement, dites-le moi!).

Exemple avec `swfaction`

```

<?php
/* Le sprite n'a qu'une lettre par image */
$p = new SWFSprite();
$p->add(new SWFAction("stop();"));
$chars = "abcdefghijklmnopqrstuvwxyz".
        "ABCDEFGHIJKLMNOPQRSTUVWXYZ".
        "1234567890!@#%&*()_+=[\]{}|;:,<.>?`~";
$f = new SWFFont("_sans");
for($n=0; $nremove($i);
    $t = new SWFTextField();
    $t->setFont($f);
    $t->setHeight(240);
    $t->setBounds(600,240);
    $t->align(SWFTEXTFIELD_ALIGN_CENTER);
    $t->addString($c);
    $i = $p->add($t);
    $p->labelFrame($c);
    $p->nextFrame();
}
/* région de clic pour le bouton : toute la fenêtre */
$s = new SWFShape();
$s->setFillStyle0($s->addSolidFill(0, 0, 0, 0));
$s->drawLine(600, 0);
$s->drawLine(0, 400);
$s->drawLine(-600, 0);
$s->drawLine(0, -400);
/* le bouton s'assure des touches pressées, et envoie le bon
   sprite à l'image de droite */
$b = new SWFButton();
$b->addShape($s, SWFBUTTON_HIT);
for($n=0; $naddAction(new SWFAction("
setTarget('/char');
gotoFrame('$c');
    ", SWFBUTTON_KEYPRESS($c));
}
$m = new SWFMovie();
$m->setDimension(600,400);
$i = $m->add($p);
$i->setName('char');
$i->moveTo(0,80);
$m->add($b);
header('Content-type: application/x-shockwave-flash');
$m->output();

```

7.59.12 swfbitmap->getheight() : Retourne la hauteur d'une bitmap

```
int 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` .

7.59.13 swfbitmap->getwidth() : Retourne la largeur d'une bitmap

```
int 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` .

7.59.14 swfbitmap() : Crée un objet bitmap

```
new swfbitmap ( string filename , int alphafilename )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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, nommé `filename` . `alphafilename` indique un fichier de masque à utiliser comme canal alpha sur une image JPEG.

Note

Seule les JPEG baseline (frame 0) sont supportés. Les baseline optimisée 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 librairie 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("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("alphafill.jpg", "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();
?>

```

7.59.15 `swfbutton_keypress()` : Retourne l'option d'actino pour `keyPress (char)`

```
int swfbutton_keypress ( string str )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.59.16 `swfbutton->addaction()` : Ajoute une action au bouton

```
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->setaction` 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`.

7.59.17 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 tous points où le bouton est dessiné est considéré comme accessible.

7.59.18 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 `addAction(shape, SWFBUTTON_MOUSEUP)`. `action` est une `swfaction`.

Voir aussi `swfbutton->addshape` et `swfaction`.

7.59.19 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`.

7.59.20 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](#) .

7.59.21 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 pour `addShape(shape, SWFBUTTON_OVER)` .

Voir aussi [swfbutton->addshape](#) et [swfaction](#) .

7.59.22 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](#) .

7.59.23 swfbutton() : Crée un nouveau bouton

```
new 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 relaché 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);
$ii = $m->add($b);
$ii->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();
?>

```

7.59.24 swfdisplayitem->addcolor() : Ajoute une couleur à une transoformation

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` .

7.59.25 swfdisplayitem->move() : Déplace un objet en coordonnées relatives

void **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` .

7.59.26 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 .

7.59.27 swfdisplayitem->multicolor() : Multiplie la couleur de transformation

```
void swfdisplayitem->multicolor ( 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->multicolor 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->multicolor

```
<?php
$b = new SWFBitmap("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();
?>
```

7.59.28 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 .

7.59.29 swfdisplayitem->rotate() : Fait tourner une forme relativement

void **swfdisplayitem->rotate** (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->rotate fait tourner la forme de ^{degrees} 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 .

7.59.30 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);
$m->setBackground(0xff, 0xff, 0xff);
// Les fonctions avec un nombre d'arguments sont vraiment une bonne idées.
// Sincèrement!
function text($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 modification 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](#) .

7.59.31 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->scaletto](#) .

7.59.32 swfdisplayitem->scalet() : Etire un objet en coordonnées globales

```
void swfdisplayitem->scalet ( 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->scalet étire un objet jusqu'au 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 .

7.59.33 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->rotate 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 .

7.59.34 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 .

7.59.35 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
$g = 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 = $g->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 = $g->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();
?>
```

7.59.36 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 ^{degrees} 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 and swfdisplayitem->skewyto .

7.59.37 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 and swfdisplayitem->skewyto .

7.59.38 swfdisplayitem->skewy() : Incline suivant les Y relativement

void **swfdisplayitem->skewy** (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->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 and swfdisplayitem->skewxto .

7.59.39 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` and `swfdisplayitem->skewxto`.

7.59.40 swfdisplayitem() : Crée un nouvel objet d'affichage displayitem

`new swfdisplayitem ( 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` crée un nouvel objet d'affichage displayitem.

C'est là que toute l'animation prend vie. Une fois que vous avez défini une forme, un texte, un sprite ou un bouton, vous l'ajoutez à une animation, puis vous utilisez la ressource retournée pour déplacer, étirer, contracter, faire tourner ou incliner la forme.

SWFDisplayItem a les méthodes suivantes : `swfdisplayitem->move`, `swfdisplayitem->moveto`, `swfdisplayitem->scaletox`, `swfdisplayitem->scale`, `swfdisplayitem->rotate`, `swfdisplayitem->rotateto`, `swfdisplayitem->skewxto`, `swfdisplayitem->skewx`, `swfdisplayitem->skewyto`, `swfdisplayitem->skewy`, `swfdisplayitem->setdepth`, `swfdisplayitem->remove`, `swfdisplayitem->setname`, `swfdisplayitem->setratio`, `swfdisplayitem->addcolor` et `swfdisplayitem->multicolor`.

7.59.41 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`).

7.59.42 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.

7.59.43 `swffill->scaletto()` : Modifie l'échelle de la forme

```
void swffill->scaletto ( 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->scaletto` modifie l'échelle de la forme de x dans le sens des abscisses et y dans le sens des ordonnées.

7.59.44 `swffill->skewxto()` : Incline (abscisses)

```
void swffill->skewxto ( 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.

7.59.45 `swffill->skewyto()` : Incline (ordonnées)

```
void swffill->skewyto ( 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.

7.59.46 `swffill()` : Crée un objet de remplissage

```
new swffill ( )
```

`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->scaletto` , `swffill->rotateto` , `swffill->skewxto` et `swffill->skewyto` .

7.59.47 swffont->getwidth() : Retourne la taille de la chaîne

```
int 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.

7.59.48 swffont() : Charge une police

```
new 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 `<font name="sans-serif">` .

swffont retourne une ressource de police, à utiliser avec les méthodes swftext->setfont et swftextfield->setfont .

SWFFont a les méthodes suivantes : swffont->getwidth .

7.59.49 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.

7.59.50 swfgradient() : Crée un objet gradient

`new 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();
?>
```

7.59.51 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](#) .

7.59.52 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](#) .

7.59.53 swfmorph() : Crée un morphing

new **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. (Tout cela, parceque Flash commence à dessiner aussitôt qu'il a les commandes de dessins. S'il conservait les descriptions de ses propres formes, et attendait leur totalité avant d'écrire, ceci et bien d'autres choses serait tellement plus simple).

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();
?>

```

7.59.54 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](#) .

7.59.55 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->setframes](#) passe à l'image suivante de l'animation.

7.59.56 swfmovie->output() : Envoie votre animation au navigateur

void **swfmovie->output** (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->output` envoie votre animation au navigateur. En PHP, faite 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.

Voir aussi `swfmovie->save` .

Des exemples d'utilisation dans : `swfmovie->streammp3` , `swfdisplayitem->rotateto` , `swfaction` ... En fait, tous les exemples utilisent cette méthode.

7.59.57 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` .

7.59.58 swfmovie->save() : Sauve dans un fichier

```
void swfmovie->save ( 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.

`swfmovie->save` sauve votre animation dans le fichier `filename` .

Voir aussi `swfmovie->output` .

7.59.59 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.

7.59.60 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.

7.59.61 swfmovie->setframes() : Modifie le nombre total d'images dans l'animation

```
void swfmovie->setframes ( string 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.

`swfmovie->setframes` modifie le nombre total d'images dans l'animation, et le fixe à `numberofframes` .

7.59.62 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 n'y ait du son en stream, auquel cas les images sont sacrifiées pour garder un son fluide.

7.59.63 swfmovie->streammp3() : Envoie un fichier MP3 en streaming

```
void swfmovie->streammp3 ( string mp3filename )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 `mp3filename` 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.

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 animation. Evitez d'en parler à la RIAA ou la SACEM.

Exemple avec `swfmovie->streammp3`

```
<?php
$m = new SWFMovie();
$m->setRate(12.0);
$m->streamMp3("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();
?>
```

7.59.64 swfmovie() : Crée un objet 'animation'

`new 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.

7.59.65 swfshape->addfill() : Ajoute un remplissage plein à la forme Flash

`void swfshape->addfill ( 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.

```
void swfshape->addfill ( SWFBitmap bitmap , 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.

```
void 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. Je crois.

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. Cette fois ci, j'en suis sûr.

[swfshape->addfill](#) retourne un objet [swffill](#) à utiliser avec [swfshape->setleftfill](#) , et [swfshape->setrightfill](#) décrite un peu plus loin.

Voir aussi [swfshape->setleftfill](#) et [swfshape->setrightfill](#) .

Ceci est un exemple simple qui affiche un cadre sur une bitmap. Ah, il y a un petit bug 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
$sp = new SWFMorph();
$bb = new SWFBitmap("alphafill.jpg");
// utilisez vos propres bitmaps!
$width = $bb->getWidth();
$height = $bb->getHeight();
$ss = $sp->getShape1();
$ff = $ss->addFill($bb, SWFFILL_TILED_BITMAP);
$ff->moveTo(-$width/2, -$height/4);
$ff->scaleTo(1.0, 0.5);
$ss->setLeftFill($ff);
$ss->movePenTo(-$width/2, -$height/4);
$ss->drawLine($width, 0);
$ss->drawLine(0, $height/2);
$ss->drawLine(-$width, 0);
$ss->drawLine(0, -$height/2);
$ss = $sp->getShape2();
$ff = $ss->addFill($bb, SWFFILL_TILED_BITMAP);
// ces déplacements n'ont aucun effet
$ff->moveTo(-$width/4, -$height/2);
$ff->scaleTo(0.5, 1.0);
$ss->setLeftFill($ff);
$ss->movePenTo(-$width/4, -$height/2);
$ss->drawLine($width/2, 0);
$ss->drawLine(0, $height);
$ss->drawLine(-$width/2, 0);
$ss->drawLine(0, -$height);
$sm = new SWFMovie();
$sm->setDimension($width, $height);
$si = $sm->add($sp);
$si->moveTo($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();
?>

```

7.59.66 swfshape->drawcurve() : Dessine une courbe relativement

void **swfshape->drawcurve** (int controldx , int controldy , int anchordx , int anchordy)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

Voir aussi [swfshape->drawlineto](#) , [swfshape->drawline](#) , [swfshape->movepento](#) et [swfshape->movepen](#) .

7.59.67 swfshape->drawcurveto() : Dessine une courbe

void **swfshape->drawcurveto** (int controlx , int controly , int anchorx , int anchory)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

Voir aussi [swfshape->drawlineto](#) , [swfshape->drawline](#) , [swfshape->movepento](#) et [swfshape->movepen](#) .

7.59.68 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->movepento](#) , [swfshape->drawcurveto](#) , [swfshape->movepen](#) et [swfshape->drawlineto](#) .

7.59.69 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->movepento](#) , [swfshape->drawcurveto](#) , [swfshape->movepen](#) et [swfshape->drawline](#) .

7.59.70 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->movepento](#) , [swfshape->drawcurveto](#) , [swfshape->drawlineto](#) et [swfshape->drawline](#) .

7.59.71 swfshape->movepento() : Déplace le stylo

```
void swfshape->movepento ( 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](#) .

7.59.72 swfshape->setleftfill() : Modifie la couleur de rastérisation 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)

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->setrightfill .

7.59.73 swfshape->setline() : Modifie le style de ligne de la forme

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.

swfshape->setline accepte 1, 4 ou 5 arguments (mais jamais 3 ou 2).

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 chaine est ici!!!
$i->moveTo(1500-$f->getWidth("!#*%")/2, 1000+$f->getAscent()/2);
header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

7.59.74 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)

Voir aussi [swfshape->setleftfill](#) .

Raccourci pour `swfshape->setrightfill($s->addfill($r, $g, $b [, $a]));` .

7.59.75 swfshape() : Crée une nouvelle forme

new **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->movepento](#) , [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();
?>

```

7.59.76 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.

7.59.77 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->setframes se déplace à la prochaine image du sprite.

7.59.78 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.

7.59.79 swfsprite->setframes() : Fixe le nombre maximum d'image 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 .

7.59.80 swfsprite() : Crée un sprite

```
new 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);
$sp = new SWFSprite();
$i = $sp->add($s);
$sp->nextFrame();
$i->rotate(15);
$sp->nextFrame();
$i->rotate(15);
$sp->nextFrame();
$i->rotate(15);
$sp->nextFrame();
$i->rotate(15);
$sp->nextFrame();
$i->rotate(15);
$sp->nextFrame();
$sm = new SWFMovie();
$i = $sm->add($sp);
$i->moveTo(1500,1000);
$i->setName("blah");
$sm->setBackground(0xff, 0xff, 0xff);
$sm->setDimension(3000,2000);
header('Content-type: application/x-shockwave-flash');
$sm->output();
?>
```

7.59.81 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.

7.59.82 swftext->getwidth() : Calcule la longueur d'une chaîne

```
void 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->addstring retourne la taille de la chaîne `string` , une fois qu'elle est dessinée avec la police et l'espacement courant.

7.59.83 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 style (ou le curseur, si ça a un sens) jusqu'au 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.

7.59.84 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->setspacing change la couleur de la police courante. Par défaut, c'est noir. La couleur est représentée avec la convention RGB.

7.59.85 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` .

7.59.86 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.

7.59.87 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 lettre n'est pas toujours le même. Il faudra que je l'explique plus clairement. Ou bien que je corrige les erreurs.

7.59.88 swftext() : Crée un nouvel objet texte

```
new 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();
?>
```

7.59.89 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.

`swftextfield->setname` concatène la chaîne `string` avec la chaîne courante.

7.59.90 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.

`swftextfield->align` 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`.

7.59.91 swftextfield->setbounds() : Sélectionne la largeur et hauteur du champs

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.

`swftextfield->setbounds` 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 (mais ne le laissez pas faire!!).

7.59.92 swftextfield->setcolor() : Modifie la couleur du champs 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.

swftextfield->setcolor 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.

7.59.93 swftextfield->setfont() : Modifie la police du champs

```
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?).

7.59.94 swftextfield->setheight() : Modifie la hauteur de la police du champs 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.

7.59.95 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} .

7.59.96 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.

7.59.97 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.

7.59.98 swftextfield->setmargins() : Modifie les marges du champs 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.

7.59.99 swftextfield->setname() : Nomme le champs 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.

7.59.100 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.

7.59.101 swftextfield() : Crée un nouveau champs texte

new **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 incliné, 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_NOEDIT : indique que le champ ne doit pas être éditable.
- SWFTEXTFIELD_PASSWORD : indique que c'est un champ mot de passe
- SWFTEXTFIELD_DRAWBOX : dessine le contour du champ
- SWFTEXTFIELD_MULTILINE : autorise les lignes multiples
- SWFTEXTFIELD_WORDWRAP : autorise la mise en forme du texte
- SWFTEXTFIELD_NOSELECT : rend le champ non-sélectionnable

Les flags peuvent être combinés avec l'opérateur OR . Par exemple :

```
<?php
$t = newSWTextField(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 .

7.60 Fonctions diverses

7.60.1 Introduction

Ces fonctions ont été placées là, car elles ne rentraient dans aucune catégorie adéquate.

7.60.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.60.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.60.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>ignore_user_abort</code>	<code>"0"</code>	<code>PHP_INI_ALL</code>
<code>highlight.string</code>	<code>#CC0000</code>	<code>PHP_INI_ALL</code>
<code>highlight.comment</code>	<code>#FF9900</code>	<code>PHP_INI_ALL</code>
<code>highlight.keyword</code>	<code>#006600</code>	<code>PHP_INI_ALL</code>
<code>highlight.bg</code>	<code>#FFFFFF</code>	<code>PHP_INI_ALL</code>
<code>highlight.default</code>	<code>#0000CC</code>	<code>PHP_INI_ALL</code>
<code>highlight.html</code>	<code>#000000</code>	<code>PHP_INI_ALL</code>
<code>browscap</code>	<code>NULL</code>	<code>PHP_INI_SYSTEM</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

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.xxx` string

Couleurs utilisées pour le mode Syntax Highlighting. Ces options peuvent prendre n'importe quelles valeurs valides dans `<font color="??????">`.

`browscap` string

Nom du fichier de descriptif des clients HTML. Voir aussi [get_browser](#).

7.60.5 Types de ressources

Cette extension ne définit aucune ressource.

7.60.6 Constantes prédefinies

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 (*integer*)
 CONNECTION_NORMAL (*integer*)
 CONNECTION_TIMEOUT (*integer*)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [connection_aborted](#)
- [connection_status](#)
- [connection_timeout](#)
- [constant](#)
- [define](#)
- [defined](#)
- [die](#)
- [eval](#)
- [exit](#)
- [get_browser](#)
- [highlight_file](#)
- [highlight_string](#)
- [ignore_user_abort](#)
- [pack](#)
- [show_source](#)
- [sleep](#)
- [uniqid](#)
- [unpack](#)
- [usleep](#)

7.60.8 connection_status() : Retourne les bits de status 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](#) .

7.60.9 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
Deprecated

This function is deprecated, and doesn't even exist anymore as of 4.0.5.

Voyez la section sur la [Gestion de la connexion](#) dans la section [Caractéristiques](#) pour plus de détails.

7.60.10 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 savez pas son nom que durant l'exécution du script. Par exemple, ce nom peut être le résultat d'une fonction.

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](#).

7.60.11 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](#).

7.60.12 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;
}
?>
```

Voir aussi define, constant, get_defined_constants, la section sur les constantes et function_exists.

7.60.13 die() : Alias de la fonction exit

void **die** (string message)

die est un alias de exit.

7.60.14 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 ligne 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. En PHP 4, eval retourne ^{NULL} sauf si ^{return} est appelée dans le code évalué, auquel cas la valeur passée à ^{return} est retournée. En PHP 3, eval ne retourne par 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 devrait afficher :

```
Ceci est une $string avec mon $name dedans.
Ceci est une tasse avec mon café dedans.
```

Array

7.60.15 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.

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.

Note

La fonction die est un alias de la fonction exit.

Exemple avec exit

```
<?php
$filename = '/chemin/jusqua/fichier';
$file = fopen ($filename, 'r')
or exit("Impossible d'ouvrir le fichier $filename");
?>
```

Voir aussi [register_shutdown_function](#).

7.60.16 get_browser() : Indique de quoi est capable le navigateur client.

object **get_browser** (string user_agent)

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.

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.

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

```
<?php
echo $_SERVER['HTTP_USER_AGENT'] . "<hr />\n";

$browser = get_browser();

foreach ($browser as $name => $value) {
    print "<b>$name</b> $value <br />\n";
}

?>
```

L'affichage devrait ressembler à ceci :

```
Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586)<hr />
<b>browser_name_pattern:</b> Mozilla/4\..*<br />
<b>parent:</b> Netscape 4.0<br />
<b>platform:</b> Linux<br />
```

```

<b>majorver:</b> 4<br />
<b>minorver:</b> 5<br />
<b>browser:</b> Netscape<br />
<b>version:</b> 4<br />
<b>frames:</b> 1<br />
<b>tables:</b> 1<br />
<b>cookies:</b> 1<br />
<b>backgroundsounds:</b> <br />
<b>vbscript:</b> <br />
<b>javascript:</b> 1<br />
<b>javaapplets:</b> 1<br />
<b>activexcontrols:</b> <br />
<b>beta:</b> <br />
<b>crawler:</b> <br />
<b>authenticodeupdate:</b> <br />
<b>msn:</b> <br />

```

Pour fonctionner, votre configuration browscap dans le `php.ini` doit pointer sur le fichier `browscap.ini` de votre système. `browscap.ini` n'est pas fourni avec `PHP` mais vous pouvez trouver une version à jour sur ce site . Par défaut, la directive browscap est désactivée.

Note

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

Sur les versions de `PHP` plus ancienne que la 4.0.6, vous devez passer le nom du navigateur comme troisième paramètre optionnel `user_agent` , et si la directive register_globals vaut `off` . Dans ce cas, vous devrez utiliser `$HTTP_SERVER_VARS['HTTP_USER_AGENT']` .

7.60.17 highlight_file() : Colorisation de la syntaxe 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 de `FALSE`.

Note

Beaucoup de soin doit être apporté lors de l'utilisation de show_source et highlight_file pour s'assurer que vous ne révélez pas d'informations critiques telles que des mots de passe ou d'autres informations qui pourrait créer des fuites d'information.

Note

Depuis PHP 4.2.1, cette fonction est aussi affectée par safe_mode et open_basedir .

Colorisation d'URL

Pour configurer une URL qui peut coloriser n'importe quel script que vous lui passez, nous avons besoin d'utiliser la directive Apache "ForceType", pour générer une URL exploitable, puis utiliser la fonction highlight_file pour afficher un code propre.

Dans votre configuration HTTP `httpd.conf`, vous pouvez ajouter le code suivant :

```
<Location /source>
    ForceType application/x-httpd-php
</Location>
```

Puis, faire un fichier appelé `source`, que vous placez dans votre racine de site web.

```
<HTML>
<HEAD>
<TITLE>Source Display</TITLE>
</HEAD>
<BODY BGCOLOR="white">
<?php
    $script = getenv ("PATH_TRANSLATED");
    if(!$script) {
        echo "<BR><B>ERREUR : Le nom du script est nécessaire</B><BR>";
    } else {
        if (ereg("(\\.php|\\.inc)$", $script)) {
            echo "<H1>Source de : $PATH_INFO</H1><HR>";
            highlight_file($script);
        } else {
            echo "<H1>ERREUR: Seuls les noms de fichier PHP ou de fichiers PH inclus sont autorisés</H1>";
        }
    }
    echo "<HR>Traitement : ".date("Y/M/d H:i:s",time());
?>
</BODY>
</HTML>
```

Alors, vous pourrez utiliser une URL telle que celle ci-dessous pour afficher une version colorisée de votre script `/path/to/script.php`.

`http://votre.serveur.com/source/jusqu/au/script.php`

Voir aussi [highlight_string](#) et [show_source](#).

7.60.18 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.

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](#).

7.60.19 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. 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`.

7.60.20 pack() : Compacte des données dans une chaîne binaire

string **pack** (string format , mixed args)

`pack` compacte les arguments 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 :

- Une chaîne complétée avec ^{NULL}
- Une chaîne complétée avec espace (SPACE)
- Chaîne hexadécimale h, bit de poids faible en premier.
- 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 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)
- N entier long non signé (toujours 16 bits, ordre des bits big endian)
- V entier long non signé (toujours 16 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 bit ^{NULL}
- X recule d'un octet
- @ rempli avec ^{NULL}, jusqu'à une position absolue

Compactage d'une chaîne

```
<?php
$binarydata = pack ("nvc*", 0x1234, 0x5678, 65, 66);
?>
```

La chaîne binaire résultante aura 6 octets de long, et contiendra la séquence 0x12, 0x34, 0x78, 0x56, 0x41, 0x42.

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](#) .

7.60.21 `show_source()` : Génère la syntaxe colorisée d'un fichier ^{PHP}

```
bool show_source ( string filename , bool return )
```

`show_source` est un alias de `highlight_file` .

Voir aussi [highlight_string](#) et [highlight_file](#) .

7.60.22 `sleep()` : Retarde l'exécution

```
void sleep ( int seconds )
```

`sleep` retarde l'exécution du programme pendant ^{seconds} secondes.

Voir aussi [usleep](#) .

7.60.23 `uniqid()` : Génère un identifiant unique

```
string uniqid ( string prefix , bool lcg )
```

`uniqid` retourne un identifiant préfixé unique, basé sur l'heure courante, en micro-secondes. Le préfixe peut servir à identifier facilement différents hôtes, si vous générez simultanément des fichiers depuis plusieurs hôtes, à la même micro-seconde. ^{prefix} peut prendre jusqu'à 114 caractères.

Si le paramètre optionnel ^{lcg} 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 ^{lcg} est à ^{TRUE} , elle fera 23 caractères.

Note
Le paramètre ^{lcg} est utilisé à partir de PHP 4 et PHP 3.0.13 et ultérieurs.

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 \$token = md5 (uniqid ("")); // pas de section aleatoire. \$better_token = md5 (uniqid (rand())); // mieux, difficile a deviner ?></pre>

Ceci va créer un identifiant de 32 caractères (un nombre hexadécimal de 128) qui sera très difficile à prédire.

7.60.24 `unpack()` : Déconditionne des données depuis une chaîne binaire.

```
array unpack ( string format , string data )
```

`unpack` déconditionne des données 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](#)

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.

7.60.25 `usleep()` : Retarde l'exécution en micro-secondes

```
void usleep ( int micro_seconds )
```

`usleep` retarde l'exécution du programme pendant `micro_seconds` micro-secondes.

Voir aussi [sleep](#).

Note

`usleep` est inopérante sous Windows

7.61 mnoGoSearch

7.61.1 Introduction

Ces fonctions donnent l'accès à mnoGoSearch (anciennement UdmSearch), moteur de recherche du monde libre. mnoGoSearch est un moteur de recherche complet, destinés aux intranet 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.ru/> .

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.61.2 Pré-requis

Téléchargez mnoGoSearch sur le site <http://www.mnogosearch.ru/> et installez le sur votre système. Vous aurez besoin de la version 3.1.10 ou plus récente de mnoGoSearch, pour pouvoir utiliser ces fonction.

7.61.3 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 librairie interne de PHP, et ne peut fonctionner qu'avec les librairies 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> .

7.61.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.61.5 Types de ressources

7.61.6 Constantes prédefinies

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 prédéfinies](#)

- [udm_add_search_limit](#)
- [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_load_ispell_data](#)
- [udm_open_stored](#)
- [udm_set_agent_param](#)

7.61.8 [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 formaté 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", les mots 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, [udm_alloc_agent](#) n'ouvre pas de connexion, et donc, ne vérifie ni le nom d'utilisateur, ni le mot

de passe.

7.61.9 udm_api_version() : Lit la version des API mnoGoSearch

`int udm_api_version ( void )`

`udm_api_version` retourne le numéro de version des 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 `udm_api_version`

```
<?php
if (udm_api_version() >= 30111) {
    print "Nombre total d'URL dans la base : ".
        udm_get_doc_count($udm)."<br>\n";
}
?>
```

7.61.10 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 courante.

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 contiennent 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.
```

Ce qui peut être affiché comme ceci :

```
Marseille
Lille
Lyon
...
```

Exemple avec `udm_cat_list`

```
<?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];
    $cat_list .= "<a href=\"$_SERVER[PHP_SELF]?cat=$path\">$name</a><br>";
}
?>

```

Voir aussi [udm_cat_path](#) .

7.61.11 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 contiennent 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](#) .

7.61.12 udm_check_charset() : Vérifie si un jeu de caractère 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.

7.61.13 `udm_check_stored()` : Vérifie la connexion MnoGoSearch avec 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.

7.61.14 `udm_clear_search_limits()` : Annule toutes les limites 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`.

7.61.15 `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.

7.61.16 `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.

7.61.17 `udm_errno()` : Numéro d'erreur mnoGoSearch

```
int udm_errno ( resource agent )
```

`udm_errno` retourne le numéro d'erreur mnoGoSearch, ou bien `0` sinon.

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`.

7.61.18 `udm_error()` : Message d'erreur mnoGoSearch

```
string udm_error ( resource agent )
```

`udm_errno` retourne le message d'erreur mnoGoSearch, ou bien une chaîne vide sinon.

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`.

7.61.19 udm_find() : Effectue une recherche

`resource udm_find ( resource agent , string query )`

`udm_add_search_limit` retourne `TRUE` 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 `query`. Pour rechercher, entrez les mots avec lesquels que 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!

() – Groupage de commandes pour les requêtes complexes : par exemple, "(mysql | msq) & ~postgres". Le mode par requête est simple et puissant à la fois. Vous pouvez utiliser les commandes booléennes habituelles avec ce mode.

7.61.20 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 `res` est un identifiant de résultat, obtenu après un appel à `udm_find`.

`udm_free_agent` détruit l'agent de recherche créé par `udm_alloc_agent`.

7.61.21 `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.

7.61.22 `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`.

7.61.23 `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 document dans les bases de données.

`agent` – Agent mnoGoSearch obtenu après un appel à `udm_alloc_agent`.

Note

<code>udm_get_doc_count</code> est supporté à partir de la version mnoGoSearch 3.1.11 ou plus récent.

7.61.24 `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`.

Le paramètre `field` est l'identifiant de champ, et peut prendre l'une des valeurs suivantes :

- `UDM_FIELD_URL` : Champs URL
- `UDM_FIELD_CONTENT` : Champs "Content-type" (par exemple, "text/html").

- `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.

7.61.25 `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 résultat 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.

7.61.26 `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 de 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 chargée 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 <code>udm_load_ispell_data</code>, 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 <code>FALSE</code> et le code d'erreur, avec son message, seront accessibles avec <code>udm_error</code> et <code>udm_errno</code>.
Exemple avec <code>udm_load_ispell_data</code>
<pre>if (! udm_load_ispell_data(\$udm, UDM_ISPELL_TYPE_DB, '', '', 1)) { printf("Erreur #d: '%s'\n", udm_errno(\$udm), udm_error(\$udm)); exit; }</pre>

- **UDM_ISPELL_TYPE_AFFIX** indique que les données ispell doivent être chargée depuis un fichier et initie le chargement. Dans ce cas, ^{val1} définit le code de langue en deux lettre, 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 <code>udm_load_ispell_data</code> et <code>UDM_ISPELL_TYPE_AFFIX</code>
<pre>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; }</pre>
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}.

Exemple avec <code>udm_load_ispell_data</code> et <code>UDM_ISPELL_TYPE_SPELL</code>
<pre><?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; } ?></pre>
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 Speldl lisent les données d'orthographe dans une configuration séparée (par défaut `/usr/local/mnogosearch/etc/speld.conf`), les trie et les stockes 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 `udm_load_ispell_data` et `UDM_ISPELL_TYPE_SERVER`

```
<?php
if (!udm_load_ispell_data($udm,UDM_ISPELL_TYPE_SERVER','',1)) {
    printf("Erreur au chargement de la librairie ispell sur le serveur<BR>\n");
    exit;
}
?>
```

7.61.27 `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.

7.61.28 `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 utilise des phrases(paramètre "phrase" dans `indexer.conf`). Valeurs disponibles : `UDM_PHRASE_ENABLED` and `UDM_PHRASE_DISABLED`. Notez bien que si la recherche par phrase est activé (`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 supporté 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. Pour la situation décrite ci-dessus,

est le facteur de poids 1 (URL)

est le facteur de poids 2 (Corps)

est le facteur de poids 4 (Titre)

est le facteur de poids 8 (Mots clés)

est le facteur de poids 16 (Description)

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)

Keywords avec un poids de 9

Title avec un poids de 4

Body 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 mots.

- UDM_PARAM_MIN_WORD_LEN – définit les tailles extrêmes de mots. Tout mot plus court que la limite inférieur 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 sera ignoré. 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 que "test", "tester", etc.. seront aussi recherchés. Les suffixes sont supportés par défaut. Les préfixes modifie 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 spécifique 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.

7.62 mSQL

7.62.1 Introduction

Ces fonctions vous permettent d'accéder aux bases de données mSQL. Plus d'informations sur mSQL à <http://www.hughes.com.au/>.

7.62.2 Pré-requis

7.62.3 Installation

Pour cela, vous devez compiler PHP avec le support msql, en utilisant l'option de configuration `--with-msql[=dir]`. Par défaut, le chemin est `'/usr/local/Hughes'`.

Note
Note aux utilisateurs Win32
Afin d'activer ce module dans l'environnement Windows, vous devez copier les librairies msql.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).

7.62.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>msql.allow_persistent</code>	"On"	PHP_INI_SYSTEM
<code>msql.max_persistent</code>	"-1"	PHP_INI_SYSTEM
<code>msql.max_links</code>	"-1"	PHP_INI_SYSTEM

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`msql.allow_persistent` bool

Activation ou non des connexions mSQL persistantes.

`msql.max_persistent` integer

Le nombre maximum de connexions mSQL persistantes par processus.

`msql.max_links` integer

Le nombre maximum de connexions mSQL par processus, y compris les connexions persistantes.

7.62.5 Types de ressources

7.62.6 Constantes prédéfinies

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.

MYSQL_ASSOC (*integer*)
MYSQL_NUM (*integer*)
MYSQL_BOTH (*integer*)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [mysql_affected_rows](#)
- [mysql_close](#)
- [mysql_connect](#)
- [mysql_create_db](#)
- [mysql_createdb](#)
- [mysql_data_seek](#)
- [mysql_dbname](#)
- [mysql_drop_db](#)
- [mysql_dropdb](#)
- [mysql_error](#)
- [mysql_fetch_array](#)
- [mysql_fetch_field](#)
- [mysql_fetch_object](#)
- [mysql_fetch_row](#)
- [mysql_field_seek](#)
- [mysql_fieldflags](#)
- [mysql_fieldlen](#)
- [mysql_fieldname](#)
- [mysql_fieldtable](#)
- [mysql_fieldtype](#)
- [mysql_free_result](#)
- [mysql_freeresult](#)
- [mysql_list_dbs](#)
- [mysql_list_fields](#)
- [mysql_list_tables](#)
- [mysql_listdbs](#)
- [mysql_listfields](#)
- [mysql_listtables](#)
- [mysql_num_fields](#)
- [mysql_num_rows](#)
- [mysql_numfields](#)
- [mysql_numrows](#)
- [mysql_pconnect](#)
- [mysql_query](#)
- [mysql_regcase](#)
- [mysql_result](#)
- [mysql_select_db](#)
- [mysql_selectdb](#)
- [mysql_tablename](#)
- [mysql](#)

7.62.8 `mysql_close()` : Ferme une connexion mSQL

```
int mysql_close ( int link_identifiant )
```

`mysql_close` retourne `TRUE` en cas de succès, `FALSE` en cas d'erreur.

`mysql_close` ferme la connexion au serveur de base de données mSQL référencé par l'identifiant fourni. Si aucun identifiant n'est fourni, la dernière connexion sera utilisée.

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.

`mysql_close` ne peut pas fermer les connexions persistantes, générées par `mysql_pconnect`.

Voir aussi `mysql_connect` et `mysql_pconnect`.

7.62.9 `mysql_connect()` : Ouvre une connexion mSQL

```
int mysql_connect ( string hostname , string server , string username , string password )
```

`mysql_connect` établit une connexion à un serveur mSQL. Le nom d'hôte est optionnel, et lorsqu'il manque, localhost est utilisé.

`mysql_connect` retourne un identifiant de connexion positif en cas de succès, et `FALSE` sinon.

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`.

Voir aussi `mysql_pconnect` et `mysql_close`.

7.62.10 `mysql_create_db()` : Crée une base de données mSQL

```
int mysql_create_db ( string database_name , int link_identifiant )
```

`mysql_create_db` essaie de créer une nouvelle base de données nommée `database_name` sur le serveur référencé par l'identifiant `link_identifiant`.

Voir aussi `mysql_drop_db`.

7.62.11 `mysql_createdb()` : Crée une base de données mSQL

```
int mysql_createdb ( string database_name , int link_identifiant )
```

Identique à `mysql_create_db`.

7.62.12 `mysql_data_seek()` : Déplace le pointeur interne mSQL

```
int mysql_data_seek ( int query_identifiant , 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.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi `mysql_fetch_row`.

7.62.13 `mysql_dbname()` : Lit le nom de la base de données courante

```
string mysql_dbname ( int query_identifiant , int i )
```

`mysql_dbname` retourne le nom de la base de données enregistré en position ⁱ du pointeur de résultat retourné par la fonction `mysql_listdbs`. La fonction `mysql_numrows` peut être utilisée pour déterminer le nombre de bases disponibles.

7.62.14 `mysql_drop_db()` : Efface une base de données mSQL

```
int mysql_drop_db ( string database_name , int link_identifiant )
```

`mysql_drop_db` essaie d'effacer une base de données entière sur le serveur référencé par l'identifiant `link_identifiant`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi `mysql_create_db`.

7.62.15 `mysql_dropdb()` : Efface une base de données mSQL

Voir `mysql_drop_db`.

7.62.16 `mysql_error()` : Retourne le message d'erreur mSQL

```
string mysql_error ( int link_identifiant )
```

`mysql_error` retourne les erreurs renvoyées par le serveur. Les erreurs générées par mSQL ne sont plus traitées comme des alertes.

7.62.17 `mysql_fetch_array()` : Lit une ligne sous la forme d'un tableau

```
int mysql_fetch_array ( int query_identifiant , int result_type )
```

`mysql_fetch_array` retourne un tableau qui contient la ligne demandée, ou `FALSE`, si il n'y a pas d'autres lignes.

`mysql_fetch_array` est une version évoluée de `mysql_fetch_row`. En plus d'enregistrer les données dans un tableau à indice numérique, il peut enregistrer les données dans un tableau associatif, en utilisant les noms des champs comme clés.

Le deuxième argument `result_type` de `mysql_fetch_array` est une constante, et peut prendre les valeurs suivantes : `MYSQL_ASSOC`, `MYSQL_NUM`, et `MYSQL_BOTH`.

Méfiez vous des requêtes qui retournent une ligne qui ne contient qu'un champ de valeur 0 (ou `NULL`, ou chaîne vide).

Il est important de noter que `mysql_fetch_array` est marginalement plus lent que `mysql_fetch_row`, alors qu'elle apporte un confort d'utilisation appréciable.

Voir aussi `mysql_fetch_row`.

7.62.18 `mysql_fetch_field()` : Lit la valeur d'un champ mSQL

object `mysql_fetch_field` (int query_identifiant , int field_offset)

`mysql_fetch_field` retourne un objet contenant les informations sur un champ.

`mysql_fetch_field` sert à lire les informations sur les champs, dans certaines requêtes. Si l'indice du champ n'est pas spécifié, le prochain champ sera retourné.

Les propriétés de l'objet sont :

- name – nom de la colonne
- table – nom de la table à qui appartient la colonne.
- not_null – 1 si la colonne ne peut être `NULL`
- primary_key – 1 si la colonne est une clé primaire
- unique – 1 la colonne est une clé unique
- type – le type de la colonne

Voir aussi `mysql_field_seek`.

7.62.19 `mysql_fetch_object()` : Lit une ligne sous la forme d'un objet

int `mysql_fetch_object` (int query_identifiant , int result_type)

`mysql_fetch_object` retourne un objet, dont les propriétés seront affectées suivant les champs de la ligne lue, ou `FALSE` si il ne reste plus de lignes.

`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)

L'argument optionnel `result_type` de `mysql_fetch_array` est une constante qui peut prendre les valeurs suivantes : `MYSQL_ASSOC`, `MYSQL_NUM`, et `MYSQL_BOTH`.

`mysql_fetch_object` est aussi rapide que `mysql_fetch_array`, et marginalement plus lente que `mysql_fetch_row` (la différence est non significative).

Voir aussi [mysql_fetch_array](#) et [mysql_fetch_row](#) .

7.62.20 `mysql_fetch_row()` : Retourne une ligne sous la forme d'un objet

```
array mysql_fetch_row ( int query_identifiant )
```

[mysql_fetch_row](#) retourne un tableau qui contient la ligne demandée, ou `FALSE` , si il n'y a plus de lignes à lire.

[mysql_fetch_row](#) retourne une ligne, extraite du résultat associé à l'identifiant de résultat `query_identifiant` . La ligne est retournée sous la forme d'un tableau. Chaque résultat est enregistré dans un champ, indexé numériquement, à partir de 0.

Les appels ultérieurs à [mysql_fetch_row](#) retourneront les lignes suivantes, ou `FALSE` , lorsqu'il n'y aura plus de ligne.

Voir aussi [mysql_fetch_array](#) , [mysql_fetch_object](#) , [mysql_data_seek](#) et [mysql_result](#) .

7.62.21 `mysql_field_seek()` : Fixe d'offset d'un champ mSQL

```
int mysql_field_seek ( int query_identifiant , 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.

Voir aussi [mysql_fetch_field](#) .

7.62.22 `mysql_fieldflags()` : Retourne le flag d'un champ mSQL

```
string mysql_fieldflags ( int query_identifiant , int i )
```

[mysql_fieldflags](#) retourne le flag du champ spécifié. Actuellement, il peut valoir soit "not `NULL` ", "primary key", ou une combinaison des deux ou "" (chaîne vide).

7.62.23 `mysql_fieldlen()` : Retourne la longueur d'un champ mSQL

```
int mysql_fieldlen ( int query_identifiant , int i )
```

[mysql_fieldlen](#) retourne la longueur du champ `i` .

7.62.24 `mysql_fieldname()` : Lit le nom d'un champ mSQL

```
string mysql_fieldname ( int query_identifiant , int field )
```

[mysql_fieldname](#) retourne le nom du champ à l'index `field` . `query_identifiant` est un identifiant de résultat, et `field` est un index de champ. `mysql_fieldname($result, 2);` retournera le nom du deuxième champ, dans le résultat associé à `query_identifiant` .

7.62.25 `mysql_fieldtable()` : Retourne le nom d'une table à partir d'un nom de champ mSQL

```
int mysql_fieldtable ( int query_identifiant , int field )
```

`mysql_fieldtable` retourne le nom de la table d'où est le champ `field` a été extrait.

7.62.26 `mysql_fieldtype()` : Retourne le type de champs mSQL

```
string mysql_fieldtype ( int query_identifiant , int i )
```

`mysql_fieldtype` est similaire à `mysql_fieldname`. Les arguments sont identiques, mais c'est le type du champ qui est retourné. Cela produira un résultat tel que "int", "string" ou "real".

7.62.27 `mysql_free_result()` : Libère le résultat de la mémoire

```
int mysql_free_result ( int query_identifiant )
```

`mysql_free_result` libère de la mémoire le résultat associé à l'identifiant de résultat `query_identifiant`. 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.

7.62.28 `mysql_freeresult()` : Libère le résultat de la mémoire

Voir `mysql_free_result`

7.62.29 `mysql_list_dbs()` : Liste les bases de données mSQL sur un serveur

```
int mysql_list_dbs ( void )
```

`mysql_list_dbs` retourne un pointeur de résultat, qui contiendra les noms des bases de données disponibles sur la connexion mSQL courante. Utilisez `mysql_dbname` pour passer en revue toutes les lignes.

7.62.30 `mysql_list_fields()` : Liste les champs d'une table

```
int mysql_list_fields ( string database , string tablename )
```

`mysql_list_fields` lit les informations de la table `tablename`. Les arguments sont le nom de la base de données, `database` et le nom de la table `tablename`. Cette fonction retourne un identifiant de résultat qui sera utilisé avec `mysql_fieldflags`, `mysql_fieldlen`, `mysql_fieldname` et `mysql_fieldtype`. Un identifiant de résultat est un entier positif. La fonction retourne -1 si une erreur survient. Une chaîne décrivant l'erreur sera placée dans la variable `$phperrormsg`, et à moins que cette fonction n'ai été appelée avec @ (`@mysql_list_fields()`), alors cette erreur sera affichée.

Voir aussi [mysql_error](#) .

7.62.31 `mysql_list_tables()` : Liste les tables mSQL sur une base de données

```
int mysql_list_tables ( string database )
```

[mysql_list_tables](#) prend un nom de base de données, et fourni un résultat, un peu comme la fonction [mysql](#) . La fonction [mysql_tablename](#) devrait être utilisée de préférence pour extraire les nom de table d'un pointeur de résultat.

7.62.32 `mysql_listdbs()` : Liste les bases de données mSQL sur un serveur

Voir [mysql_list_dbs](#) .

7.62.33 `mysql_listfields()` : Liste les champs d'une table

Voir [mysql_list_fields](#) .

7.62.34 `mysql_listtables()` : Liste les tables mSQL sur une base de données

Voir [mysql_list_tables](#) .

7.62.35 `mysql_num_fields()` : Retourne le nombre de champs dans un résultat

```
int mysql_num_fields ( int query_identifiant )
```

[mysql_num_fields](#) retourne le nombre de champs du résultat associé à l'identifiant `query_identifiant` .

Voir aussi [mysql](#) , [mysql_query](#) , [mysql_fetch_field](#) et [mysql_num_rows](#) .

7.62.36 `mysql_num_rows()` : Retourne le nombre de lignes dans un résultat

```
int mysql_num_rows ( int query_identifiant )
```

[mysql_num_rows](#) retourne le nombre de lignes du résultat associé à l'identifiant `query_identifiant` .

Voir aussi [mysql](#) , [mysql_query](#) et [mysql_fetch_row](#) .

7.62.37 `mysql_numfields()` : Retourne le nombre de champs dans un résultat

```
int mysql_numfields ( int query_identifieur )
```

Identique à `mysql_num_fields`.

7.62.38 `mysql_numrows()` : Retourne le nombre de lignes dans un résultat

```
int mysql_numrows ( void )
```

Identique à `mysql_num_rows`.

7.62.39 `mysql_pconnect()` : Ouvre une connexion persistante à un serveur mSQL

```
int mysql_pconnect ( string server , string username , string password )
```

`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`).

`mysql_pconnect` retourne un identifiant de connexion persistante à un serveur mSQL en cas de succès, et `FALSE` sinon.

C'est pourquoi une telle connexion est considérée comme 'persistante'.

7.62.40 `mysql_query()` : Envoie une requête mSQL

```
int mysql_query ( string query , int link_identifieur )
```

`mysql_query` envoie une requête à la base de données active, sur le serveur associé à l'identifiant de connexion `link_identifieur`. Si `link_identifieur` n'est pas fourni, PHP tentera d'utiliser la dernière connexion ouverte. Si aucune connexion n'a été ouverte, la fonction tentera de se connecter par elle-même, avec `mysql_connect` appelé sans argument.

`mysql_query` retourne un identifiant positif mSQL en cas de succès, et `FALSE` sinon.

Exemple avec `mysql_query`

```
<?php
$link = mysql_connect("dbserver")
    or die("unable to connect to mysql server: ".mysql_error());
mysql_select_db("db", $link)
    or die("unable to select database 'db': ".mysql_error());
$result = mysql_query("SELECT * FROM table WHERE id=1", $link);
```

```

if (!$result) {
    die("query failed: ".mysql_error());
}

while ($row = mysql_fetch_array($result)) {
    echo $row["id"];
}
?>

```

Voir aussi [mysql](#) , [mysql_select_db](#) , et [mysql_connect](#) .

7.62.41 [mysql_regcase\(\)](#) : Prépare une chaîne pour une recherche par expression régulière insensible à la casse.

Voir [sql_regcase](#) .

7.62.42 [mysql_result\(\)](#) : Retourne les données de résultat

```

int mysql_result ( int query_identifiant , int i , mixed field )

```

[mysql_result](#) retourne la valeur de la cellule, à la ligne ⁱ et l'offset spécifié, ^{field} dans le résultat mSQL ^{query_identifiant} .

[mysql_result](#) retourne le contenu d'une cellule depuis un résultat mSQL ^{query_identifiant} . L'argument de champ ^{field} peut être aussi bien un offset, qu'un nom de champ, ou encore le nom de la table point le nom du fichier (nom_table.nom_champ). Si la colonne est un alias, (par exemple 'select foo as bar from...'), utilisez de préférence l'alias au nom de colonne.

Lorsque vous travailler 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](#) . De plus, sachez qu'accéder à un champ avec son indice numérique est beaucoup plus rapide qu'en utilisant les autres méthodes.

Alternatives recommandées : [mysql_fetch_row](#) , [mysql_fetch_array](#) et [mysql_fetch_object](#) .

7.62.43 [mysql_select_db\(\)](#) : Sélectionne une base de données mSQL

```

int mysql_select_db ( string database_name , int link_identifiant )

```

[mysql_select_db](#) retourne ^{TRUE} en cas de succès, et ^{FALSE} en cas d'erreur.

[mysql_select_db](#) choisi la base de données courante sur le serveur associé à l'identifiant de connexion ^{link_identifiant} . Si ^{link_identifiant} n'est pas fourni, PHP tentera d'utiliser la dernière connexion ouverte. Si aucune connexion n'a été ouverte, la fonction tentera de se connecter par elle-même, avec [mysql_connect](#) appelée sans argument.

Les prochains appels à [mysql_query](#) seront fait dans la base de données active.

Voir aussi [mysql_connect](#) , [mysql_pconnect](#) et [mysql_query](#) .

7.62.44 `mysql_selectdb()` : Sélectionne une base de données mSQL

Reportez vous à [mysql_select_db](#) .

7.62.45 `mysql_tablename()` : Retourne le nom d'une table à partir d'un nom de champs

```
string mysql_tablename ( int query_identifieur , int field )
```

`mysql_tablename` prend un pointeur de résultat (retourné par la fonction [mysql_list_tables](#)), ainsi qu'un index, et retourne le nom d'une table. La fonction [mysql_numrows](#) peut servir à déterminer le nombre de table dans le pointeur de résultat.

Exemple `mysql_tablename`

```
<?php
mysql_connect ("localhost");
$result = mysql_list_tables("limousin");
$i = 0;
while ($i < mysql_numrows ($result)) {
    $tb_names[$i] = mysql_tablename ($result, $i);
    echo $tb_names[$i] . "<BR>";
    $i++;
}
?>
```

7.62.46 `mysql()` : Exécute une requête mSQL

```
int mysql ( string database , string query , int link_identifieur )
```

`mysql` retourne un identifiant positif de résultat de requête, ou `FALSE` en cas d'erreur.

`mysql` sélectionne la base de données `database` , et y exécute la requête `query` . Si l'identifiant de connexion `link_identifieur` n'est pas fourni, la fonction va rechercher un lien ouvert à un serveur mSQL, et sinon, il va tenter d'en créer une, avec [mysql_connect](#) , sans argument.

7.63 MySQL

7.63.1 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://www.mysql.com/documentation/> .

7.63.2 Pré-requis

Afin de pouvoir les utiliser, vous devez compiler PHP avec le support MySQL.

7.63.3 Installation

En utilisant l'option `--with-mysql` . Si vous utilisez cette fonction sans préciser le chemin d'accès à la base MySQL, PHP utilisera les bibliothèques clientes MySQL fournies en standard. Les utilisateurs qui font tourner d'autres applications qui utilisent elles-mêmes MySQL (par exemple, PHP 3 et PHP 4 utilisés comme des modules concurrents apache, ou encore auth-mysql), devraient toujours spécifier le chemin jusqu'à MySQL : `--with-mysql=/path/to/mysql` . Cela va forcer PHP à utiliser les bibliothèques clientes installées par MySQL et évitera les conflits.

En ^{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.

Cette extension MySQL extension ne fonctionnera pas avec les versions de MySQL supérieure à la version 4.1.0. Pour cela, il faut utiliser l'extension [MySQLi](#) .

Attention

Des crashes et des problèmes de démarrage de ^{PHP} peuvent être rencontré lorsque vous chargez cette fonction en même temps que l'extension [recode](#). Voyez l'extension [recode](#) 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 libmysql (non fournie), compilée avec ce jeu de caractères.

7.63.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<code>mysql.allow_persistent</code>	"On"	PHP_INI_SYSTEM
<code>mysql.max_persistent</code>	"-1"	PHP_INI_SYSTEM
<code>mysql.max_links</code>	"-1"	PHP_INI_SYSTEM
<code>mysql.default_port</code>	NULL	PHP_INI_ALL
<code>mysql.default_socket</code>	NULL	PHP_INI_ALL
<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>	"0"	PHP_INI_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [ini_set](#).

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` integer

Le nombre maximum de connexions persistantes aux bases de données MySQL, par processus.

`mysql.max_links` integer

Le nombre maximum de connexions aux bases de données MySQL, y compris les connexions persistantes, par processus.

`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 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 order.

Win32 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 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 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 est activé.

`mysql.connect_timeout` integer

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.

7.63.5 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`.

7.63.6 Constantes prédefinies

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
<code>MYSQL_CLIENT_COMPRESS</code>	Utilise le protocole avec compression
<code>MYSQL_CLIENT_IGNORE_SPACE</code>	Autorise les espaces après les noms de fonction
<code>MYSQL_CLIENT_INTERACTIVE</code>	Autorise <code>interactive_timeout</code> secondes d'inactivité sur la connexion (au lieu de <code>wait_timeout</code>).

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.

7.63.7 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_host", "mysql_user", "mysql_password") or die("Could not connect"); print "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 */ print "<table>\n"; while (\$line = mysql_fetch_array(\$result, MYSQL_ASSOC)) { print "\t<tr>\n"; foreach (\$line as \$col_value) { print "\t\t<td>\$col_value</td>\n"; } print "\t</tr>\n"; } print "</table>\n"; /* Libération des résultats */ mysql_free_result(\$result); /* Fermeture de la connexion */ mysql_close(\$link); ?> </pre>

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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](#)

7.63.9 [mysql_change_user\(\)](#) : Change le nom de session de l'utilisateur actif

```
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`. Si une base est spécifiée, elle deviendra la base par défaut de l'utilisateur. Si une erreur de connexion survient, la connexion en cours restera active.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note
<u>mysql_change_user</u> a été introduite en PHP 3.0.13 et requiert MySQL 3.23.3 ou plus récent.

7.63.10 `mysql_client_encoding()` : Retourne le nom du jeu de caractères utilisé par le client MySQL

`int mysql_client_encoding ( resource link_identifieur )`

`mysql_client_encoding` retourne le nom du jeu de caractères utilisé par le client mysql, pour la connexion courante (si omise), ou pour la connexion `link_identifieur`.

Exemple avec `mysql_client_encoding`

```
<?php
$link = mysql_connect('localhost', 'utilisateur_mysql', 'mot_de_passe');
$charset = mysql_client_encoding($link);
printf ("Le jeu de caractères actuel est %s.\n", $charset);
?>
```

L'exemple ci-dessus va afficher ceci :

Le jeu de caractères actuel est latin1.

Voir aussi `mysql_real_escape_string`.

7.63.11 `mysql_close()` : Ferme la connexion MySQL

`bool mysql_close ( resource link_identifieur )`

`mysql_close` ferme la connexion au serveur MySQL associée à l'identifiant `link_identifieur`. Si cet identifiant n'est pas spécifié, cette commande s'applique à la dernière connexion ouverte.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note

Notez que cette commande n'est pas nécessaire, car toutes les connexions non persistantes seront automatiquement fermées à la fin du script.

`mysql_close` ne ferme pas les connexions persistantes générées par `mysql_pconnect`.

Exemple `MySQL_close`

```
<?php
$link = mysql_connect("serveur.mysql.com", "utilisateur", "secret")
or die("Impossible de se connecter");

print("Connexion réussie");
mysql_close($link);
?>
```

Voir aussi `mysql_connect` et `mysql_pconnect`.

7.63.12 `mysql_connect()` : Ouvre une connexion à un serveur MySQL

`resource mysql_connect ( string server , string username , string password , bool new_link , int client_flags )`

`mysql_connect` retourne une ressource de connexion au serveur MySQL en cas de succès, et sinon `FALSE`.

`mysql_connect` établit une connexion à un serveur MySQL. Tous les arguments sont optionnels, et s'ils manquent, les valeurs par défaut sont utilisées ('localhost', nom du propriétaire du processus, mot

de passe vide).

Le nom d'hôte peut aussi inclure un numéro de port, sous la forme : "hostname:port" ou un chemin jusqu'à une socket sous la forme ":/path/to/socket" pour l'hôte localhost.

Note

Le support des "port" a été ajouté à partir de la version 3.0B4.

Le support de ":/path/to/socket" a été ajouté à partir de la version 3.0.10.

Vous pouvez supprimer le message d'erreur de connexion en ajoutant une arobase '@' au nom de la fonction.

Si un second appel à `mysql_connect` est fait avec les mêmes arguments, PHP ne va pas ouvrir une nouvelle connexion, mais va retourner l'identifiant de la connexion déjà ouverte. Le paramètre `new_link` modifie ce comportement et permet que `mysql_connect` ouvre une nouvelle connexion même si `mysql_connect` a été appelée auparavant avec les mêmes paramètres. Le paramètre `client_flags` peut être une combinaison des constantes suivantes : `MYSQL_CLIENT_SSL`, `MYSQL_CLIENT_COMPRESS`, `MYSQL_CLIENT_IGNORE_SPACE` et `MYSQL_CLIENT_INTERACTIVE`.

Note

La paramètre `new_link` a été ajouté en PHP 4.2.0.

La paramètre `client_flags` a été ajouté en PHP 4.3.0

Le lien sera fermé automatiquement dès que l'exécution du script sera terminée, à moins d'être fermé explicitement avec `mysql_close`.

Exemple avec `mysql_connect`

```
<?php
$link = mysql_connect("localhost", "mysql_user", "mysql_password")
    or die("Impossible de se connecter : " . mysql_error());
print ("Connexion réussie.");
mysql_close($link);
?>
```

Voir aussi `mysql_pconnect` et `mysql_close`.

7.63.13 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 nommée `database_name` sur le serveur associé à l'identifiant `link_identifier`, ou sur la dernière connexion ouverte.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple de création de base MySQL avec `mysql_create_db`

```
<?php
$link = mysql_pconnect ("kron", "jutta", "geheim") {
    or die ("Connexion impossible");
}
if (mysql_create_db ("my_db")) {
    print ("Base de données créée\n");
} else {
    echo "Erreur lors de la création de la base: ".mysql_error();
}
?>
```

Pour des raisons de compatibilité ascendante, `mysql_createdb()` est toujours utilisable.

Note

La fonction `mysql_create_db` est obsolète. Il est préférable d'utiliser la fonction `mysql_query`, avec une commande SQL "CREATE DATABASE".

Voir aussi `mysql_drop_db` et `mysql_query`

7.63.14 `mysql_data_seek()` : Déplace le pointeur interne de résultat

`bool mysql_data_seek ( resource result_identifiant , int row_number )`

`mysql_data_seek` déplace le pointeur interne de résultat, dans le résultat associé à l'identifiant de résultat `result_identifiant`. Il le fait pointer à la ligne `row_number`. Le prochain appel à `mysql_fetch_row` retournera cette ligne.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

`row_number` commence à 0. `row_number` doit être une valeur qui va de 0 à `mysql_num_rows` - 1.

Note

La fonction `mysql_data_seek` peut être utilisée en conjonction avec `mysql_query`, mais pas avec la fonction `mysql_unbuffered_query`.

Exemple `mysql_data_seek`

```
<?php
$link = mysql_pconnect ("kron", "jutta", "geheim") {
    or die ("Connexion impossible");
}
mysql_select_db ("samp_db") {
    or die ("Sélection de base impossible");
}
$query = "SELECT last_name, first_name FROM friends";
$result = mysql_query ($query) {
    or die ("Requête impossible");
}
// récupère les lignes dans l'ordre inverse
for ($i = mysql_num_rows ($result) - 1; $i >= 0; $i--) {
    if (!mysql_data_seek ($result, $i)) {
        echo "Impossible d'atteindre la ligne $i\n";
        continue;
    }
    if (!($row = mysql_fetch_object ($result)))
        continue;
    echo "{$row->last_name} {$row->first_name}<BR>\n";
}
mysql_free_result ($result);
?>
```

Voir aussi `mysql_query` et `mysql_num_rows`.

7.63.15 `mysql_db_name()` : Lit les noms des bases de données

`string mysql_db_name ( resource result , int row , mixed field )`

`mysql_db_name` prend comme premier argument le pointeur de résultat `result_identifiant`, issu de `mysql_list_dbs`. `row` est l'index dans le résultat.

Si une erreur survient, `FALSE` est retourné. Utilisez `mysql_errno` et `mysql_error` pour connaître la nature de l'erreur.

Exemple `mysql_db_name`

```
<?php
error_reporting(E_ALL);
mysql_connect('dbhost', 'username', 'password');
$db_list = mysql_list_dbs();
$i = 0;
$cnt = mysql_num_rows($db_list);
while ($i < $cnt) {
    echo mysql_db_name($db_list, $i) . "\n";
    $i++;
}
?>
```

Pour des raisons de compatibilité ascendante, `mysql_dbname()` est aussi accepté, mais obsolète.

7.63.16 `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` retourne un identifiant de résultat si la requête réussit et `FALSE` sinon.

`mysql_db_query` sélectionne une base de données et exécute une requête. Si l'identifiant de lien `link_identifier` n'est pas précisé, `mysql_db_query` prendra par défaut la dernière connexion ouverte sur le serveur et si elle n'en trouve pas, elle tentera de se connecter, en utilisant la fonction `mysql_connect`, sans arguments.

Voir aussi `mysql_connect` et `mysql_query`.

Note

Cette fonction est obsolète, et abandonnée depuis PHP 4.0.6. Ne l'utilisez plus (ou pas!). Alternative recommandée : `mysql_select_db` et `mysql_query`.

7.63.17 `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 une base de données entière sur le serveur associé à la ressource de connexion `link_identifier`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Pour des raisons de compatibilité ascendante, `mysql_dropdb()` peut encore être utilisée. Son usage est déconseillé.

Note

La fonction `mysql_drop_db` est obsolète. Il est préférable d'utiliser `mysql_query` avec la requête SQL DROP DATABASE à la place.

Voir aussi `mysql_create_db` et `mysql_query`.

7.63.18 mysql_errno() : Retourne le numéro d'erreur de la dernière commande MySQL

int **mysql_errno** (resource link_identifiant)

mysql_errno retourne le numéro de message d'erreur de la dernière opération MySQL sur la connexion courante, ou sur la connexion spécifiée avec l'option `link_identifiant`. 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.

Exemple avec mysql_errno

```
<?php
mysql_connect("marliesle");
echo mysql_errno()." : ".mysql_error()."<BR>";
mysql_select_db("nonexistentdb");
echo mysql_errno()." : ".mysql_error()."<BR>";
$conn = mysql_query("SELECT * FROM nonexistenttable");
echo mysql_errno()." : ".mysql_error()."<BR>";
?>
```

The above example would produce the following output:

```
1049: Unknown database 'nonexistentdb'
1146: Table 'kossu.nonexistenttable' doesn't exist
```

Voir aussi mysql_error.

7.63.19 mysql_error() : Retourne le texte associé avec l'erreur générée lors de la dernière requête.

string **mysql_error** (resource link_identifiant)

mysql_error retourne le message d'erreur généré par la dernière commande MySQL, ou bien '' (la chaîne vide) si aucune erreur n'est survenue.

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, ce qui fait que si vous souhaitez l'utiliser, vous devez vous assurer de sa valeur avant de lancer une autre requête.

Exemple avec mysql_error

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password");

mysql_select_db("nonexistentdb");
echo mysql_errno() . " : " . mysql_error() . "\n";

mysql_select_db("kossu");
mysql_query("SELECT * FROM nonexistenttable");
echo mysql_errno() . " : " . mysql_error() . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
1049: Unknown database 'nonexistentdb'
1146: Table 'kossu.nonexistenttable' doesn't exist
```

Note

Si le paramètre optionnel `link_identifiant` est utilisé, c'est le dernier message d'erreur sur cette connexion qui sera lue. Sinon, c'est la dernière connexion MySQL ouverte qui est utilisée.

Voir aussi [mysql_errno](#) .

7.63.20 `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` . Elle retourne la chaîne modifiée.

Note

`mysql_escape_string` n'échappe pas les caractères pourcentage % ou - .

Note

`mysql_escape_string` est identique à la fonction `mysql_real_escape_string` hormi 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 respecte pas le jeu de caractères courant.

Exemple avec `mysql_escape_string`

```
<?php
$item = "C'est un exemple de Zak";
$escaped_item = mysql_escape_string($item);
printf ("Chaîne protégée : %s\n", $escaped_item);
?>
```

L'exemple ci-dessus affichera :

Chaîne protégée : C'est un exemple de Zak

Voir aussi [mysql_real_escape_string](#) , [addslashes](#) et les [magic_quotes_gpc](#) directive.

7.63.21 `mysql_fetch_array()` : Retourne une ligne de résultat 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, ou `FALSE` si il ne reste plus de ligne.

`mysql_fetch_array` est une version étendue de `mysql_fetch_row` . En plus d'enregistrer les données sous forme d'un tableau à indice numérique, elle peut aussi les enregistrer dans un tableau associatif, en utilisant les noms des champs comme indices.

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.

Requête avec duplication de nom de colonnes

```
select table1.field as foo, table2.field as bar from table1, table2
```

Il est important de souligner que `mysql_fetch_array` N'est PAS plus lente que `mysql_fetch_row`, tandis qu'elle ajoute un confort d'utilisation notable.

L'option `result_type` de `mysql_fetch_array` est une constante qui peut prendre les valeurs suivantes : `MYSQL_ASSOC`, `MYSQL_NUM` et `MYSQL_BOTH`. Cette fonctionnalité a été ajoutée en PHP 3.0.7. `MYSQL_BOTH` est la valeur par défaut de cet argument.

En utilisant `MYSQL_BOTH`, vous obtiendrez un tableau qui sera à la fois associatif et indexé numériquement. En utilisant `MYSQL_ASSOC`, vous obtiendrez un tableau associatif (tout comme le fait `mysql_fetch_assoc`), et en utilisant `MYSQL_NUM`, vous obtiendrez un tableau indexé numériquement (comme le fait `mysql_fetch_row`).

Note

Les noms des champs retournés par cette fonction sont **sensibles à la casse**.

Exemple de `mysql_fetch_array` avec `MYSQL_NUM`

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or
    die("Impossible de se connecter : " . mysql_error());
mysql_select_db("mydb");

$result = mysql_query("SELECT id, name FROM mytable");

while ($row = mysql_fetch_array($result, MYSQL_NUM)) {
    printf ("ID: %s Name: %s", $row[0], $row[1]);
}

mysql_free_result($result);
?>
```

Exemple de `mysql_fetch_array` avec `MYSQL_ASSOC`

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or
    die("Impossible de se connecter : " . mysql_error());
mysql_select_db("mydb");

$result = mysql_query("SELECT id, name FROM mytable");

while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
    printf ("ID: %s Name: %s", $row["id"], $row["name"]);
}

mysql_free_result($result);
?>
```

Exemple de `mysql_fetch_array` avec `MYSQL_BOTH`

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or
    die("Impossible de se connecter : " . mysql_error());
mysql_select_db("mydb");

$result = mysql_query("SELECT id, name FROM mytable");

while ($row = mysql_fetch_array($result, MYSQL_BOTH)) {
    printf ("ID: %s Name: %s", $row[0], $row["name"]);
}

mysql_free_result($result);
?>
```

Voir aussi `mysql_fetch_row` et `mysql_fetch_assoc`.

7.63.22 mysql_fetch_assoc() : Lit une ligne de résultats 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` , ou bien `FALSE` , s'il ne reste plus de lignes à lire.

[mysql_fetch_assoc](#) est équivalente à [mysql_fetch_array](#) utilisée avec l'option `MYSQL_ASSOC`. Elle ne retourne qu'un tableau associatif. C'est le fonctionnement original de [mysql_fetch_array](#) . Si vous avez besoin d'indices numériques, utilisez [mysql_fetch_array](#) .

Si plusieurs colonnes portent le même nom, la dernière aura la priorité. Pour accéder aux autres colonnes du même nom, vous devez utiliser [mysql_fetch_array](#) et les indices numériques.

Une chose importante à noter est que [mysql_fetch_assoc](#) n'est PAS significativement plus lente que [mysql_fetch_row](#) , alors qu'elle apporte un confort d'utilisation important.

Pour plus de détails, reportez-vous à [mysql_fetch_row](#) et [mysql_fetch_array](#) .

Note

Les noms des champs retournés par cette fonction sont **sensibles à la casse** .

Exemple avec [mysql_fetch_assoc](#)

```
<?php
$conn = mysql_connect("localhost", "mysql_user", "mysql_password");

if (!$conn) {
    echo "Impossible de se connecter à la base de données : " . mysql_error();
    exit;
}

if (!mysql_select_db("mydbname")) {
    echo "Impossible d'accéder à la base de données : " . mysql_error();
    exit;
}

$sql = "SELECT id as userid, fullname, userstatus
      FROM   sometable
      WHERE  userstatus = 1";

$result = mysql_query($sql);

if (!$result) {
    echo "Impossible d'exécuter la requête ($sql) dans la base : " . mysql_error();
    exit;
}

if (mysql_num_rows($result) == 0) {
    echo "Aucune ligne trouvées, rien à afficher.";
    exit;
}

// Tant qu'une ligne existe, place cette ligne dans la variable $row
// sous la forme d'un tableau associatif.
// Note : Si vous n'attendez qu'une seule ligne, oubliez la boucle
// Note : Si vous utilisez extract($row); dans la boucle suivante
// vous créez then create $userid, $fullname, and $userstatus
while ($row = mysql_fetch_assoc($result)) {
    echo $row["userid"];
    echo $row["fullname"];
    echo $row["userstatus"];
}

mysql_free_result($result);

?>
```

Voir aussi [mysql_fetch_row](#) , [mysql_fetch_array](#) , [mysql_query](#) et [mysql_error](#) .

7.63.23 mysql_fetch_field() : Retourne les données enregistrées dans une colonne sous forme d'objet.

object **mysql_fetch_field** (resource result , int field_offset)

mysql_fetch_field retourne un objet contenant les données, lu dans le résultat `result` .

mysql_fetch_field sert à obtenir des informations à propos des champs, dans certaines requêtes. Si l'indice du champ n'est pas spécifié, le champ suivant le dernier champ retourné, est retourné.

Les propriétés de l'objet sont :

- name – nom de la colonne
- table – nom de la table de la colonne
- max_length – taille maximale de la colonne
- not_null – 1 si la colonne ne peut pas être ^{NULL} (attribut NOT NULL)
- primary_key – 1 si la colonne est une clé primaire (attribut PRIMARY KEY)
- unique_key – 1 si la colonne est une clé unique (attribut UNIQUE)
- multiple_key – 1 si la colonne est une clé non-unique
- numeric – 1 si la colonne est numérique
- blob – 1 si la colonne est BLOB
- type – le type de la colonne
- unsigned – 1 si la colonne est non signée
- zerofill – 1 si la colonne est complétée par des zéros.

Note

Les noms des champs retournés par cette fonction sont **sensibles à la casse** .

Exemple avec mysql_fetch_field

```
<?php
mysql_connect('localhost:3306', $user, $password)
    or die("Impossible de se connecter : " . mysql_error());
mysql_select_db("database");
$result = mysql_query("select * from table")
    or die("Query failed: " . mysql_error());
/* Lecture des méta données de la colonne */
$i = 0;
while ($i < mysql_num_fields($result)) {
    echo "Détails sur la colonne $i :<br />\n";
    $meta = mysql_fetch_field($result);
    if (!$meta) {
        echo "Aucun détail disponible<br />\n";
    }
    echo "<pre>
blob:          $meta->blob
max_length:    $meta->max_length
multiple_key:  $meta->multiple_key
name:          $meta->name
not_null:      $meta->not_null
numeric:       $meta->numeric
primary_key:   $meta->primary_key
table:         $meta->table
type:          $meta->type
unique_key:    $meta->unique_key
unsigned:      $meta->unsigned
zerofill:      $meta->zerofill
</pre>";
    $i++;
}
mysql_free_result($result);
?>
```

Voir aussi mysql_field_seek .

7.63.24 `mysql_fetch_lengths()` : Retourne la taille de chaque colonne d'une ligne de résultat

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`, et lue dans le résultat `result`, sinon `FALSE`.

`mysql_fetch_lengths` stocke les tailles de chaque colonne de la dernière ligne retournée par `mysql_fetch_row`, `mysql_fetch_array` et `mysql_fetch_object` dans un tableau, en commençant à la position 0.

Voir aussi `mysql_fetch_row`.

7.63.25 `mysql_fetch_object()` : Retourne une ligne de résultat MySQL sous la forme d'un objet

object `mysql_fetch_object ( resource result )`

`mysql_fetch_object` retourne un objet dont les propriétés correspondent à une ligne du résultat `result`, ou `FALSE` si il n'y a plus d'autres lignes.

`mysql_fetch_object` est identique à `mysql_fetch_array`, à la différence qu'elle retourne un objet à la place d'un tableau. Vous pourrez ainsi accéder aux valeurs des champs par leur nom, mais plus par leur offset (les nombres ne sont pas des noms MySQL).

```
<?php
/* Ceci est valide */
echo $row->field;
/* Ceci est invalide */
echo $row->0;

?>
```

Note
Les noms des champs retournés par cette fonction sont sensibles à la casse .

L'argument optionnel `result_type` est une constante qui peut prendre les valeurs suivantes : `MYSQL_ASSOC`, `MYSQL_NUM` et `MYSQL_BOTH`.

Concernant la vitesse, `mysql_fetch_object` est aussi rapide que `mysql_fetch_array` et presque aussi rapide que `mysql_fetch_row` (la différence est insignifiante)

Exemple avec <code>mysql_fetch_object</code>
<pre><?php mysql_connect("hostname", "user", "password"); mysql_select_db(\$db); \$result = mysql_query("select * from table"); while (\$row = mysql_fetch_object(\$result)) { echo \$row->user_id; echo \$row->fullname; } mysql_free_result(\$result); ?></pre>

Voir aussi `mysql_fetch_assoc`, `mysql_fetch_array` et `mysql_fetch_row`.

7.63.26 `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`, ou bien `FALSE` s'il ne reste plus de ligne.

Note

Les noms des champs retournés par cette fonction sont **sensibles à la casse**.

`mysql_fetch_row` va rechercher une ligne dans le résultat associé à l'identifiant de résultat spécifié. La ligne est retournée sous la forme d'un tableau. Chaque colonne est enregistrée sous la forme d'un tableau commençant à la position ⁰.

Les appels suivants à `mysql_fetch_row` retourneront la ligne suivante dans le résultat, ou `FALSE` si il n'y a plus de ligne disponible.

Voir aussi `mysql_fetch_assoc`, `mysql_fetch_array`, `mysql_fetch_object`, `mysql_data_seek`, `mysql_fetch_lengths` et `mysql_result`.

7.63.27 `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`. Les sémaphores sont retournés comme des mots, séparés par des espaces, ce qui les rend faciles à séparer, avec la commande `explode`.

Les valeurs suivantes (pour une version suffisamment récente de MySQL) sont disponibles : "not_null", "primary_key", "unique_key", "multiple_key", "blob", "unsigned", "zerofill", "binary", "enum", "auto_increment", "timestamp".

Pour des raisons de compatibilité ascendante, `mysql_fieldflags()` peut encore être utilisé.

7.63.28 `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 `field_offset`-ième champ dans le résultat `result`. Les champs sont indexés à partir de 0.

Pour des raisons de compatibilité ascendante, `mysql_fieldlen()` peut encore être utilisé.

7.63.29 `mysql_field_name()` : Retourne le nom d'une colonne

string `mysql_field_name` (resource result , int field_index)

`mysql_field_name` retourne le nom de la `field_index` –ième colonne du résultat MySQL `result`. `result` doit être une ressource de résultat MySQL et `field_index` est un offset numérique du champ.

Note

`field_index` commence à 0.

e.g. L'index du troisième champ sera en fait l'offset 2, et l'index du quatrième champ sera en fait l'offset 3, etc....

Note

Les noms des champs retournés par cette fonction sont **sensibles à la casse** .

Exemple avec `mysql_field_name`

```
/* Supposons que la table utilisée contienne trois champs :
 *   user_id
 *   username
 *   password.
 */
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
mysql_select_db($dbname, $link)
    or die("Impossible de se connecter à la base $dbname : " . mysql_error());
$res = mysql_query("select * from users", $link);

echo mysql_field_name($res, 0) . "\n";
echo mysql_field_name($res, 2);
```

L'exemple ci-dessus va afficher :

```
user_id
password
```

Pour des raisons de compatibilité ascendante, `mysql_fieldname()` peut encore être utilisé.

7.63.30 `mysql_field_seek()` : Déplace le pointeur de résultat

```
int mysql_field_seek ( resource result , int field_offset )
```

Place le pointeur de résultat sur le champ spécifié. Lors du prochain appel à `mysql_fetch_field` qui n'aura pas d'argument d'index de champ, le champ désormais pointé sera retourné.

Voir aussi `mysql_fetch_field` .

7.63.31 `mysql_field_table()` : Retourne le nom de la table 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` .

Pour des raisons de compatibilité ascendante, `mysql_fieldtable()` peut encore être utilisé. Toutefois, elle est obsolète.

7.63.32 mysql_field_type() : Retourne le type de la colonne courante

string **mysql_field_type** (resource result , int field_offset)

mysql field type est similaire à la fonction mysql field name . Les arguments sont identiques : c'est le type du champ qui est retourné. Il vaudra "int", "real", "string", "blob" ou d'autres, comme détaillé dans la documentation MySQL .

Types MySQL

```
<?php
mysql_connect("localhost", "mysql_utilisateur", "mysql_mot_de_passe");
mysql_select_db("mysql");
$result = mysql_query("SELECT * FROM func");
$fields = mysql_num_fields($result);
$rows = mysql_num_rows($result);
$table = mysql_field_table($result, 0);
echo "Votre table '". $table. "' dispose de ".$fields." colonnes et ".$rows." ligne(s)\n";
echo "Les colonnes sont les suivantes : \n";
for ($i=0; $i < $fields; $i++) {
    $type = mysql_field_type($result, $i);
    $name = mysql_field_name($result, $i);
    $len = mysql_field_len($result, $i);
    $flags = mysql_field_flags($result, $i);
    echo $type." ". $name." ". $len." ". $flags." \n";
}
mysql_free_result($result);
mysql_close();
?>
```

L'exemple ci-dessus produit l'affichage suivant :

```
Votre table 'func' dispose de 4 colonnes et 1 ligne(s)
Les colonnes sont les suivantes :
string name 64 not_null primary_key binary
int ret 1 not_null
string dl 128 not_null
string type 9 not_null enum
```

Pour des raisons de compatibilité ascendante, mysql_fieldtype() peut encore être utilisé. Toutefois, elle est obsolète.

7.63.33 mysql_free_result() : Efface 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.

Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

Pour des raisons de compatibilité ascendante, mysql_freeresult() peut encore être utilisé.

7.63.34 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.

Exemple avec mysql get client info

```
<?php
```

```
printf ("Détails sur le client MySQL : %s\n", mysql_get_client_info());
?>
```

L'exemple ci-dessus pourrait afficher par exemple :

```
Détails sur le client MySQL : 3.23.39
```

Voir aussi [mysql_get_host_info](#) , [mysql_get_proto_info](#) et [mysql_get_server_info](#) .

7.63.35 mysql_get_host_info() : Lit les informations sur l'hôte MySQL

string **mysql_get_host_info** (resource link_identifiant)

[mysql_get_host_info](#) retourne une chaîne qui représente le type de connexion utilisé avec la connexion `link_identifiant` , y compris le nom du serveur hôte. Si `link_identifiant` est omis, la dernière connexion ouverte est utilisée.

Exemple avec [mysql_get_host_info](#)

```
<?php
mysql_connect("hote.mysql", "utilisateur", "mot de passe") or
die("Impossible de se connecter : " . mysql_error());
printf ("Informations sur l'hôte MySQL : %s\n", mysql_get_host_info());
?>
```

L'exemple ci-dessus va afficher (par exemple) :

```
Informations sur l'hôte MySQL : Localhost via UNIX socket
```

Voir aussi [mysql_get_client_info](#) , [mysql_get_proto_info](#) et [mysql_get_server_info](#) .

7.63.36 mysql_get_proto_info() : Lit les informations sur le protocole MySQL

int **mysql_get_proto_info** (resource link_identifiant)

[mysql_get_proto_info](#) retourne une chaîne qui représente la version du protocole utilisé par la connexion `link_identifiant` . Si `link_identifiant` est omis, la dernière connexion ouverte est utilisée.

Exemple avec [mysql_get_proto_info](#)

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or
die("Impossible de se connecter : " . mysql_error());
printf ("Version du protocole MySQL : %s\n", mysql_get_proto_info());
?>
```

L'exemple ci-dessus pourrait afficher :

```
Version du protocole MySQL : 10
```

Voir aussi [mysql_get_client_info](#) , [mysql_get_host_info](#) et [mysql_get_server_info](#) .

7.63.37 `mysql_get_server_info()` : Lit les informations sur le serveur MySQL

string `mysql_get_server_info` (resource link_identifiant)

`mysql_get_server_info` retourne une chaîne qui représente la version du serveur, dont la connexion est Si `link_identifiant` est omis, la dernière connexion ouverte est utilisée.

Exemple avec `mysql_get_server_info`

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or
    die("Could not connect: " . mysql_error());
printf ("MySQL server version: %s\n", mysql_get_server_info());
?>
```

L'exemple ci-dessus va afficher (par exemple) :

MySQL server version: 4.0.1-alpha

Voir aussi `mysql_get_client_info` , `mysql_get_host_info` et `mysql_get_proto_info` .

7.63.38 `mysql_info()` : Lit des informations à propos de la dernière requête

string `mysql_info` (resource link_identifiant)

`mysql_info` retourne des informations détaillées sur la dernière requête exécutée sur la dernière connexion ouverte (si omise), ou sur la connexion `link_identifiant` .

`mysql_info` retourne une chaîne avec tous les détails listés ci-dessous. Pour toutes les autres commandes, elle retourne `FALSE` . Le format de la chaîne dépend de la commande.

Commande utilisables avec `mysql_info`

```
INSERT INTO ... SELECT ...
String format: Records: 23 Duplicates: 0 Warnings: 0
INSERT INTO ... VALUES (...),(...),(...)...
String format: Records: 37 Duplicates: 0 Warnings: 0
LOAD DATA INFILE ...
String format: Records: 42 Deleted: 0 Skipped: 0 Warnings: 0
ALTER TABLE
String format: Records: 60 Duplicates: 0 Warnings: 0
UPDATE
String format: Rows matched: 65 Changed: 65 Warnings: 0
```

Les nombres ci-dessus ne sont là que pour illustrer. Leur valeur s'adapte à la situation de la requête.

Note

`mysql_info` retourne une valeur différente de `FALSE` pour les commandes `INSERT ... VALUES` uniquement si plusieurs jeux de valeurs à insérer ont été spécifiés dans la commande.

Voir aussi `mysql_affected_rows` .

7.63.39 `mysql_insert_id()` : Retourne l'identifiant généré par la dernière requête INSERT.

`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 bien sûr la connexion spécifiée par `link_identifieur`. Si `link_identifieur` est omis, la dernière connexion ouverte est utilisée.

`mysql_insert_id` retourne 0 si la dernière requête SQL n'a pas généré de valeur `AUTO_INCREMENT`. Si vous devez lire cette valeur pour la conserver et l'utiliser ultérieurement, appelez la fonction `mysql_insert_id` juste après l'exécution de la requête, pour obtenir cette valeur non parasitée.

Note

La fonction MySQL `LAST_INSERT_ID()` contient toujours la valeur `AUTO_INCREMENT` la plus récente, et n'est pas remis à zéro entre deux requêtes.

Attention

`mysql_insert_id` convertit le type de valeur retourné par la fonction C de MySQL `C mysql_insert_id()` en type PHP `long` (appelé `int` en PHP). Si votre colonne de type `AUTO_INCREMENT` est une colonne de type `BIGINT`, la valeur retournée par `mysql_insert_id` sera incorrecte. A la place, utilisez la fonction interne MySQL `LAST_INSERT_ID()` dans une requête SQL.

Exemple avec `mysql_insert_id`

```
<?php
mysql_connect("localhost", "mysql_user", "mysql_password") or
    die("Impossible de se connecter : " . mysql_error());
mysql_select_db("mydb");

mysql_query("INSERT INTO mytable (product) values ('kossu')");
printf ("Le dernier ID inséré dans est le %d\n", mysql_insert_id());
?>
```

Voir aussi [mysql_query](#).

7.63.40 `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 bien sur la connexion spécifiée par `link_identifieur`. La fonction `mysql_tablename`, ainsi que toutes les fonctions d'accès aux résultats comme `mysql_fetch_array` peuvent être utilisées pour déterminer le nom des bases dans le pointeur de résultat.

Exemple avec `mysql_list_dbs`

```
<?php
$link = mysql_connect('localhost', 'utilisateur_mysql', 'mot_de_passe_mysql');
$db_list = mysql_list_dbs($link);

while ($row = mysql_fetch_object($db_list)) {
    echo $row->Database . "\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
database1
database2
```

```
database3
...
```

Note

The above code would just as easily work with [mysql_fetch_row](#) or other similar functions.

Pour des raisons de compatibilité ascendante, `mysql_listdbs()` est encore disponible.

Voir aussi [mysql_db_name](#).

7.63.41 mysql_list_fields() : Liste les champs du résultat MySQL

resource **mysql_list_fields** (string database_name , string table_name , resource link_identifier)

[mysql_list_fields](#) recherche les informations relatives à la table `table_name` sur la connexion MySQL courante, ou bien sûr la connexion spécifiée par `link_identifier`. Les arguments sont la base de données et le nom de la table. Un pointeur de résultat est retourné et pourra être passé à [mysql_field_flags](#), [mysql_field_len](#), [mysql_field_name](#) et [mysql_field_type](#).

Un identifiant de résultat est un entier positif. La fonction retourne -1 si une erreur survient. Une chaîne décrivant le problème rencontré sera placée dans la variable `$phperrormsg` et, à moins que la fonction n'ait été appelée sous la forme `@mysql()`, cette erreur sera aussi affichée.

Exemple avec [mysql_list_fields](#)

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');

$fields = mysql_list_fields("database1", "table1", $link);
$num_columns = mysql_num_fields($fields);

for ($i = 0; $i < $num_columns; $i++) {
    echo mysql_field_name($fields, $i) . "\n";
}
```

Le script ci-dessus va afficher :

```
field1
field2
field3
...
```

Pour des raisons de compatibilité ascendante, `mysql_listfields()` est encore disponible.

7.63.42 mysql_list_processes() : Liste les processus MySQL

resource **mysql_list_processes** (resource link_identifier)

[mysql_list_processes](#) retourne une ressource de résultat qui liste les processus courants sur le serveur.

Exemple avec [mysql_list_processes](#)

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');

$result = mysql_list_processes($link);
while ($row = mysql_fetch_row($result)){
    printf("%s %s %s %s %s\n", $row["Id"], $row["Host"], $row["db"],
        $row["Command"], $row["Time"]);
}
mysql_free_result ($result);
?>
```

L'exemple ci-dessus va afficher :

```
1 localhost test Processlist 0
4 localhost mysql sleep 5
```

Voir aussi [mysql_thread_id](#) .

7.63.43 mysql_list_tables() : Liste les tables d'une base de données

resource **mysql_list_tables** (string database , resource link_identifiant)

[mysql_list_tables](#) prend le nom d'une base de données `database` et retourne un identifiant de résultat, qui contiendra la liste des tables sur la connexion MySQL courante, ou bien sûr la connexion spécifiée par `link_identifiant` . La fonction [mysql_tablename](#) , ainsi que toutes les fonctions d'accès aux résultats comme [mysql_fetch_array](#) peuvent être utilisées pour déterminer le nom des tables dans le pointeur de résultat.

Le paramètre `database` est le nom de la base de données à utiliser pour lister les tables. En cas d'échec, [mysql_list_tables](#) retourne `FALSE` .

Pour des raisons de compatibilité ascendante, la fonction `mysql_listtables()` est toujours disponible. Elle est désormais obsolète.

Exemple avec [mysql_list_tables](#)

```
<?php
$dbname = 'test';

if (!mysql_connect('serveur_mysql', 'utilisateur_mysql', 'mot_de_passe_mysql')) {
    print 'Impossible de se connecter à MySQL';
    exit;
}

$result = mysql_list_tables($dbname);

if (!$result) {
    print "Erreur : impossible de lister les bases de données\n";
    print 'Erreur MySQL : ' . mysql_error();
    exit;
}

while ($row = mysql_fetch_row($result)) {
    print "Table : $row[0]\n";
}

mysql_free_result($result);
?>
```

Voir aussi [mysql_list_dbs](#) et [mysql_tablename](#) .

7.63.44 mysql_num_fields() : Retourne le nombre de champs d'un résultat MySQL

int **mysql_num_fields** (resource result)

[mysql_num_fields](#) retourne le nombre de champs du résultat MySQL `result` .

Voir aussi [mysql_select_db](#) , [mysql_query](#) , [mysql_fetch_field](#) et [mysql_num_rows](#) .

Pour des raisons de compatibilité ascendante `mysql_numfields()` est encore disponible.

7.63.45 `mysql_num_rows()` : Retourne le nombre de lignes d'un résultat

`int mysql_num_rows ( resource result )`

`mysql_num_rows` retourne le nombre de lignes d'un résultat. Cette commande n'est valide que pour les commandes `SELECT`. Pour connaître le nombre de lignes retournées par `INSERT`, `UPDATE` ou `DELETE`, utilisez `mysql_affected_rows`.

Exemple `mysql_num_rows`

```
<?php
$link = mysql_connect("localhost", "mysql_user", "mysql_password");
mysql_select_db("database", $link);

$result = mysql_query("SELECT * FROM table1", $link);
$num_rows = mysql_num_rows($result);

echo "$num_rows Rows\n";

?>
```

Note

Si vous utilisez `mysql_unbuffered_query`, `mysql_num_rows` ne retournera pas le nombre correct de lignes lues tant que vous n'aurez pas lu toutes les lignes de la base.

Voir aussi `mysql_affected_rows`, `mysql_connect`, `mysql_data_seek`, `mysql_select_db` et `mysql_query`.

Pour des raisons de compatibilité ascendante `mysql_numrows()` est encore disponible.

7.63.46 `mysql_pconnect()` : Ouvre une connexion persistante à un serveur MySQL.

`resource mysql_pconnect ( string server , string username , string password , int client_flags )`

`mysql_pconnect` retourne un lien persistant positif en cas de succès et sinon `FALSE` en cas d'erreur.

`mysql_pconnect` établit une connexion persistante à un serveur MySQL. Tous les arguments sont optionnels et des valeurs par défaut seront utilisés en cas d'omission ('localhost', nom d'utilisateur propriétaire du processus, mot de passe vide).

Le nom de l'hôte peut aussi inclure le numéro de port, c'est-à-dire "hostname:port" ou un chemin jusqu'à la socket `:/path/to/socket` pour l'hôte local.

Note

Le support de "":port" a été ajouté à partir de la version 3.0B4.

Le support de "":/path/to/socket" a été ajouté à partir de la version 3.0.10.

`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'.

Note

Notez que les connexions persistantes ne fonctionnent que si vous utilisez ^{PHP} en version module. Voyez la section sur les connexions persistantes aux bases de données pour plus d'informations.

Attention

L'utilisation des connexions persistantes requiert des paramétrages d'Apache et de MySQL pour vous assurer que vous n'atteindrez pas la limite maximale de nombre de connexions simultanées.

7.63.47 mysql_ping() : Pingue le serveur, et se reconnecte au besoin

`bool mysql_ping ( resource link_identifieur )`

`mysql_ping` vérifie si la connexion au serveur 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. `mysql_ping` retourne ^{TRUE} si la connexion au serveur fonctionne, et ^{FALSE} sinon.

Voir aussi `mysql_thread_id` et `mysql_list_processes` .

7.63.48 mysql_query() : Envoie une requête SQL à un serveur MySQL

`resource mysql_query ( string query , resource link_identifieur )`

`mysql_query` envoie une requête SQL à la base de données actuellement active sur le serveur MySQL. Si `link_identifieur` n'est pas précisé, la dernière connexion est utilisée. Si aucune connexion n'a été ouverte, la fonction tentera d'en ouvrir une, avec la fonction `mysql_connect` mais sans aucun paramètre (c'est-à-dire avec les valeurs par défaut).

Le paramètre optionnel `result_mode` peut prendre les valeurs de `MYSQL_USE_RESULT` ou `MYSQL_STORE_RESULT` . Par défaut, il vaut `MYSQL_STORE_RESULT` , ce qui fait que le résultat est mis dans un buffer. Voyez aussi la fonction `mysql_unbuffered_query` pour comprendre cas où le résultat n'est pas mis en buffer.

Note

La requête ne doit pas être terminée par un point-virgule.

Seules les requêtes SQL `SELECT` , `SHOW` , `EXPLAIN` et `DESCRIBE` , utilisées avec `mysql_query` retourne une ressource de résultat MySQL, ou bien ^{FALSE} si la requête n'a pu être exécuté correctement. Pour les autres types de requêtes SQL, `mysql_query` retourne ^{TRUE} en cas de succès et ^{FALSE} en cas d'erreur. Un résultat différent de ^{FALSE} , la valeur signifie que la requête soumise est valide et qu'elle a pu être exécutée correctement. Cela n'apporte aucune indication sur le nombre de lignes affectées. Il est parfaitement possible qu'une requête soit réussie, mais qu'elle ne modifie aucune ligne, ou qu'elle n'en retourne aucune.

L'exemple suivant est syntaxiquement invalide, ce qui conduit `mysql_query` à l'échec et retourne ^{FALSE} :

Exemple avec `mysql_query`

```
<php
$result = mysql_query("SELECT * WHERE 1=1")
    or die ("Requête invalide");
```



```
?>
```

L'exemple suivant est sémantiquement invalide si `ma_colonne` n'est pas une colonne de la table `my_table`, ce qui conduit `mysql_query` à l'échec et retourne `FALSE` :

Exemple avec `mysql_query` (2)

```
<?php
$result = mysql_query("SELECT ma_colonne FROM ma_table")
    or die ("Requête invalide");
?>
```

`mysql_query` échouera aussi et retournera aussi `FALSE` si les droits d'accès ne sont pas suffisants.

En supposant que la requête réussisse, vous pouvez appeler `mysql_affected_rows` pour connaître le nombre de lignes affectées (pour les commandes DELETE, INSERT, REPLACE, ou UPDATE).

Pour les commandes SELECT , `mysql_query` retourne un identifiant de résultat que vous pouvez passer à `mysql_fetch_array` , ou n'importe quelle autre fonction de lecture de résultat. Lorsque vous avez terminé avec le résultat, libérez la mémoire avec `mysql_free_result` .

Voir aussi `mysql_num_rows` , `mysql_affected_rows` , `mysql_unbuffered_query` , `mysql_free_result` , `mysql_fetch_array` , `mysql_fetch_row` , `mysql_fetch_assoc` , `mysql_result` , `mysql_select_db` et `mysql_connect` .

7.63.49 mysql_real_escape_string() : Protège les caractères spéciaux d'une commande SQL.

`string mysql_real_escape_string ( string unescaped_string , resource link_identifieur )`

`mysql_real_escape_string` protège les caractères spéciaux de la chaîne `unescaped_string` , en prenant en compte le jeu de caractères courant. Le résultat peut être utilisé sans problème avec la fonction `mysql_query` .

Note

`mysql_real_escape_string` ne remplace ni `%` , ni `-` .

Exemple avec `mysql_real_escape_string`

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
$item = "C'est la belle vie, n'est ce pas?";
$escaped_item = mysql_real_escape_string($item);
printf ("Chaîne résultat : : %s\n", $escaped_item);
?>
```

L'exemple ci-dessus va afficher ceci :

Chaîne résultat : C'est la belle vie, n'est ce pas?

Voir aussi `mysql_escape_string` et `mysql_character_set_name` .

7.63.50 mysql_result() : Retourne un champ d'un résultat

`mixed mysql_result ( resource result , int row , mixed field )`

`mysql_result` retourne le contenu d'un champ dans le résultat MySQL `result_identifieur` . L'argument `field` peut être un indice de champ, ou le nom d'un champ, ou le nom de la table + point + le nom du champ ("table.champ"). Si la colonne a été aliassée, utilisez de préférence l'alias.

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 numériques est aussi beaucoup plus rapide que spécifier un nom littéral.

Les appels [mysql_result](#) ne devraient pas être mélangés avec d'autres fonctions qui travaillent aussi sur le résultat.

Exemple avec [mysql_result](#)

```
<?php
$link = mysql_connect("hote", "utilisateur", "mot_de_passe")
    or die("Impossible de se connecter : " . mysql_error());

$result = mysql_query("SELECT name FROM work.employee")
    or die("Impossible d'exécuter la requête : " . mysql_error());

echo mysql_result($result,2); // Affiche le nom du troisième employé

mysql_close($link);
?>
```

Alternatives à haut rendement, RECOMMANDÉES : [mysql_fetch_row](#), [mysql_fetch_array](#) et [mysql_fetch_object](#).

7.63.51 mysql_select_db() : Sélectionne une base de données MySQL

bool **mysql_select_db** (string database_name , resource link_identifier)

[mysql_select_db](#) change la base de données active sur la connexion représentée par [link_identifier](#). Si aucun identifiant n'est spécifié, la dernière connexion est utilisée. S'il n'y a pas de dernière connexion, la fonction tentera de se connecter seule, avec [mysql_connect](#) et les paramètres par défaut.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Toutes les requêtes suivantes avec [mysql_query](#) seront faites avec la base de données active.

Voir aussi [mysql_connect](#), [mysql_pconnect](#) et [mysql_query](#).

Pour des raisons de compatibilité ascendante [mysql_selectdb\(\)](#) est encore disponible. Elle est toutefois obsolète.

7.63.52 mysql_stat() : Retourne le statut courant du serveur

string **mysql_stat** (resource link_identifier)

[mysql_stat](#) retourne le statut courant du serveur.

Note

[mysql_stat](#) actuellement ne retourne que le temps de fonctionnement, les threads, les requêtes, les tables ouvertes, les tables flushées, et le nombre de requêtes par secondes. Pour une liste complète des statuts du serveur, utilisez la requête SHOW STATUS.

Exemple [mysql_stat](#)

```
<?php
$link = mysql_connect('localhost', "mysql_user", "mysql_password");
$status = explode(' ',mysql_stat($link));
print_r($status);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => Uptime: 5380
    [1] => Threads: 2
    [2] => Questions: 1321299
    [3] => Slow queries: 0
    [4] => Opens: 26
    [5] => Flush tables: 1
    [6] => Open tables: 17
    [7] => Queries per second avg: 245.595
)
```

7.63.53 `mysql_tablename()` : Lit le nom de la table qui contient un champ

string **mysql_tablename** (resource result , int i)

`mysql_tablename` prend le pointeur de résultat obtenu avec `mysql_list_tables` ou bien un index entier et retourne le nom de la table. La fonction `mysql_tablename`, ainsi que toutes les fonctions d'accès aux résultats comme `mysql_fetch_array` peuvent être utilisées pour déterminer le nom des tables dans le pointeur de résultat.

Exemple avec `mysql_tablename`

```
<?php
mysql_connect("localhost", "utilisateur_mysql", "mot_de_passe_mysql");
$result = mysql_list_tables("ma_bdd");

for ($i = 0; $i < mysql_num_rows($result); $i++)
    printf ("Table : %s\n", mysql_tablename($result, $i));

mysql_free_result($result);
?>
```

Voir aussi `mysql_list_tables`.

7.63.54 `mysql_thread_id()` : Retourne l'identifiant du thread courant

int **mysql_thread_id** (resource link_identifiant)

`mysql_thread_id` retourne l'identifiant du thread courant. Si la connexion est perdue et que vous vous reconnectez (avec `mysql_ping` par exemple), alors l'identifiant de thread va changer. Cela signifie que vous devez lire l'identifiant de thread, et le garder pour plus tard. Lisez le lorsque vous en avez besoin.

Exemple avec `mysql_thread_id`

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
$thread_id = mysql_thread_id($link);
if ($thread_id){
    printf ("L'identifiant du thread courant est %d\n", $thread_id);
}
?>
```

L'exemple ci-dessus va afficher :

```
L'identifiant du thread courant est 73
```

Voir aussi `mysql_ping` et `mysql_list_processes`.

7.63.55 `mysql_unbuffered_query()` : Exécute une requête SQL sans mobiliser les résultats

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 grandes tailles. 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.

Note

L'intérêt de <code>mysql_unbuffered_query</code> est tempéré par une limitation : <code>mysql_num_rows</code> ne fonctionne pas sur une ressource retournée par <code>mysql_unbuffered_query</code> . Vous devez aussi lire tous les résultats d'une première requête exécutée avec <code>mysql_unbuffered_query</code> , avant de pouvoir en exécuter une autre.

Voir aussi `mysql_query` .

7.64 Extension MySQL améliorée

7.64.1 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://www.mysql.com/documentation/> .

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.

7.64.2 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 ou plus de MySQL. Pour les versions plus anciennes, voyez la documentation de l'extension mysql.

7.64.3 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 librairie embarquée libmysql distribuée avec ^{PHP} ne peut être utilisée. Utilisez à la place les librairies clientes installées par MySQL avec des versions inférieures à la 4.1. Cela forcera ^{PHP} à utiliser les librairies du client installées par MySQL pour éviter les conflits.

7.64.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` ^{php.ini} .

Nom	Par défaut	Modifiable
<code>mysqli.max_links</code>	<code>"-1"</code>	<code>PHP_INI_SYSTEM</code>
<code>mysqli.default_port</code>	<code>NULL</code>	<code>PHP_INI_ALL</code>
<code>mysqli.default_socket</code>	<code>NULL</code>	<code>PHP_INI_ALL</code>
<code>mysqli.default_host</code>	<code>NULL</code>	<code>PHP_INI_ALL</code>
<code>mysqli.default_user</code>	<code>NULL</code>	<code>PHP_INI_ALL</code>
<code>mysqli.default_pw</code>	<code>NULL</code>	<code>PHP_INI_ALL</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

```
mysqli.max_links integer
```

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. Si 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_password` 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.

7.64.5 Types de ressources

7.64.5.1 `mysqli_link`

Représente une connexion entre ^{PHP} et une base de données MySQL.

7.64.5.2 `mysqli_stmt`

Représente une commande préparée.

7.64.5.3 `mysqli_result`

Représente le jeu de résultats obtenu d'une requête sur la base de données.

7.64.6 Constantes prédefinies

Nom	Description
<code>MYSQLI_READ_DEFAULT_GROUP</code> (<i>integer</i>)	
<code>MYSQLI_READ_DEFAULT_FILE</code> (<i>integer</i>)	
<code>MYSQLI_OPT_CONNECT_TIMEOUT</code> (<i>integer</i>)	
<code>MYSQLI_OPT_LOCAL_INFILE</code> (<i>integer</i>)	
<code>MYSQLI_INIT_COMMAND</code> (<i>integer</i>)	
<code>MYSQLI_CLIENT_SSL</code> (<i>integer</i>)	
<code>MYSQLI_CLIENT_COMPRESS</code> (<i>integer</i>)	
<code>MYSQLI_CLIENT_INTERACTIVE</code> (<i>integer</i>)	
<code>MYSQLI_CLIENT_IGNORE_SPACE</code> (<i>integer</i>)	
<code>MYSQLI_CLIENT_NO_SCHEMA</code> (<i>integer</i>)	
<code>MYSQLI_CLIENT_MULTI_QUERIES</code> (<i>integer</i>)	
<code>MYSQLI_STORE_RESULT</code> (<i>integer</i>)	
<code>MYSQLI_USE_RESULT</code> (<i>integer</i>)	
<code>MYSQLI_ASSOC</code> (<i>integer</i>)	
<code>MYSQLI_NUM</code> (<i>integer</i>)	
<code>MYSQLI_BOTH</code> (<i>integer</i>)	

<u>MYSQLI_NOT_NULL_FLAG (integer)</u>
<u>MYSQLI_PRI_KEY_FLAG (integer)</u>
<u>MYSQLI_UNIQUE_KEY_FLAG (integer)</u>
<u>MYSQLI_MULTIPLE_KEY_FLAG (integer)</u>
<u>MYSQLI_BLOB_FLAG (integer)</u>
<u>MYSQLI_UNSIGNED_FLAG (integer)</u>
<u>MYSQLI_ZEROFILL_FLAG (integer)</u>
<u>MYSQLI_AUTO_INCREMENT_FLAG (integer)</u>
<u>MYSQLI_TIMESTAMP_FLAG (integer)</u>
<u>MYSQLI_SET_FLAG (integer)</u>
<u>MYSQLI_NUM_FLAG (integer)</u>
<u>MYSQLI_PART_KEY_FLAG (integer)</u>
<u>MYSQLI_GROUP_FLAG (integer)</u>
<u>MYSQLI_TYPE_DECIMAL (integer)</u>
<u>MYSQLI_TYPE_TINY (integer)</u>
<u>MYSQLI_TYPE_SHORT (integer)</u>
<u>MYSQLI_TYPE_LONG (integer)</u>
<u>MYSQLI_TYPE_FLOAT (integer)</u>
<u>MYSQLI_TYPE_DOUBLE (integer)</u>
<u>MYSQLI_TYPE_NULL (integer)</u>
<u>MYSQLI_TYPE_TIMESTAMP (integer)</u>
<u>MYSQLI_TYPE_LONGLONG (integer)</u>
<u>MYSQLI_TYPE_INT24 (integer)</u>
<u>MYSQLI_TYPE_DATE (integer)</u>
<u>MYSQLI_TYPE_TIME (integer)</u>
<u>MYSQLI_TYPE_DATETIME (integer)</u>
<u>MYSQLI_TYPE_YEAR (integer)</u>
<u>MYSQLI_TYPE_NEWDATE (integer)</u>
<u>MYSQLI_TYPE_ENUM (integer)</u>
<u>MYSQLI_TYPE_SET (integer)</u>
<u>MYSQLI_TYPE_TINY_BLOB (integer)</u>
<u>MYSQLI_TYPE_MEDIUM_BLOB (integer)</u>
<u>MYSQLI_TYPE_LONG_BLOB (integer)</u>
<u>MYSQLI_TYPE_BLOB (integer)</u>
<u>MYSQLI_TYPE_VAR_STRING (integer)</u>
<u>MYSQLI_TYPE_STRING (integer)</u>
<u>MYSQLI_TYPE_CHAR (integer)</u>
<u>MYSQLI_TYPE_INTERVAL (integer)</u>
<u>MYSQLI_TYPE_GEOMETRY (integer)</u>
<u>MYSQLI_BIND_STRING (integer)</u>
<u>MYSQLI_BIND_INT (integer)</u>
<u>MYSQLI_BIND_DOUBLE (integer)</u>
<u>MYSQLI_BIND_SEND_DATA (integer)</u>
<u>MYSQLI_RPL_MASTER (integer)</u>
<u>MYSQLI_RPL_SLAVE (integer)</u>
<u>MYSQLI_RPL_ADMIN (integer)</u>
<u>MYSQLI_NEED_DATA (integer)</u>
<u>MYSQLI_NO_DATA (integer)</u>
<u>MYSQLI_PR_REPORT_STDERR (integer)</u>
<u>MYSQLI_PR_REPORT_PORT (integer)</u>
<u>MYSQLI_PR_REPORT_FILE (integer)</u>

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [mysqli_affected_rows](#)
- [mysqli_autocommit](#)
- [mysqli_bind_param](#)
- [mysqli_bind_result](#)
- [mysqli_change_user](#)
- [mysqli_character_set_name](#)
- [mysqli_close](#)
- [mysqli_commit](#)
- [mysqli_connect](#)

- [mysqli_data_seek](#)
- [mysqli_debug](#)
- [mysqli_disable_reads_from_master](#)
- [mysqli_disable_rpl_parse](#)
- [mysqli_dump_debug_info](#)
- [mysqli_enable_reads_from_master](#)
- [mysqli_enable_rpl_parse](#)
- [mysqli_errno](#)
- [mysqli_error](#)
- [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_host_info](#)
- [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_num_fields](#)
- [mysqli_num_rows](#)
- [mysqli_options](#)
- [mysqli_param_count](#)
- [mysqli_ping](#)
- [mysqli_prepare_result](#)
- [mysqli_prepare](#)
- [mysqli_profiler](#)
- [mysqli_query](#)
- [mysqli_read_query_result](#)
- [mysqli_real_connect](#)
- [mysqli_real_escape_string](#)
- [mysqli_real_query](#)
- [mysqli_reload](#)
- [mysqli_rollback](#)
- [mysqli_rpl_parse_enabled](#)
- [mysqli_rpl_probe](#)
- [mysqli_rpl_query_type](#)
- [mysqli_select_db](#)
- [mysqli_send_long_data](#)
- [mysqli_send_query](#)

- [mysqli_slave_query](#)
- [mysqli_ssl_set](#)
- [mysqli_stat](#)
- [mysqli_stmt_affected_rows](#)
- [mysqli_stmt_close](#)
- [mysqli_stmt_errno](#)
- [mysqli_stmt_error](#)
- [mysqli_stmt_store_result](#)
- [mysqli_store_result](#)
- [mysqli_thread_id](#)
- [mysqli_thread_safe](#)
- [mysqli_use_result](#)
- [mysqli_warning_count](#)

7.64.8 [mysqli_autocommit\(\)](#) : Active ou désactive le mode auto-commit

`bool mysqli\_autocommit ( resource link , 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`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Utilisation de la fonction [mysqli_autocommit](#)

```
<?php

/* Ouvre une connexion */
$link = mysqli_connect("localhost", "user", "pass");
mysqli_select_db("mydb");

/* active le mode auto-commit */
mysqli_autocommit($link, true);

?>
```

7.64.9 [mysqli_bind_param\(\)](#) : Lie des variables à une requête préparée

`bool mysqli\_bind\_param ( resource stmt , mixed variable , int type )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Note

Vous pouvez passer des paires additionnelles de variables/types à cette fonction.

7.64.10 [mysqli_bind_result\(\)](#) : Lie des variables à une requête préparée pour le stockage de résultat

`bool mysqli\_bind\_result ( resource stmt , mixed var , int len )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.11 `mysqli_change_user()` : Change l'utilisateur de la connexion spécifiée

`bool mysqli_change_user ( resource link , 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`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

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 l'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.

Utilisation de `mysqli_change_user`

```
<?php

/* Ouvre une connexion en tant que foo@localhost et sélectionne foo_db */
$link = mysqli_connect("localhost", "foo", "pass");
mysqli_select_db($link, "foo_db");

/* Change l'utilisateur en bar@localhost et la base de données en bar_db */
mysqli_change_user($link, "bar", "otherpass", "bar_db");

?>
```

7.64.12 `mysqli_character_set_name()` : Retourne le jeu de caractères courant pour la connexion

`string mysqli_character_set_name ( resource link )`

Retourne le jeu de caractères courant pour la connexion spécifiée par le paramètre `link`.

Utilisation de `mysqli_character_set_name`

```
<?php

/* Ouvre une connexion */
$link = mysqli_connect("localhost", "username", "password");

/* Affiche le jeu de caractères courant */
$charset = mysqli_character_set_name($link);
printf("Le jeu de caractères courant est %s\n", $charset);

?>
```

Voir aussi `mysqli_real_escape_string`.

7.64.13 `mysqli_close()` : Ferme une connexion

`bool mysqli_close ( resource link )`

La fonction `mysqli_close` ferme la connexion spécifiée par le paramètre `link`.

Voir aussi `mysqli_connect` et `mysqli_real_connect`.

7.64.14 `mysqli_commit()` : Valide la transaction courante

`bool mysqli_commit ( resource link )`

Valide la transaction courante pour la base de données spécifiée par le paramètre `link`.

Voir aussi `mysqli_autocommit` et `mysqli_rollback`.

7.64.15 `mysqli_connect()` : Ouvre une connexion à un serveur MySQL

`resource mysqli_connect ( string hostname , 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 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 n'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 qu'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 `socket` ne déterminera pas explicitement le type de connexion qui sera utilisé lors de la connexion au serveur MySQL. Cela est déterminé par le paramètre `host`.

Utilisation de la fonction `mysqli_connect`

```
<?php
/* Ouvre une connexion en tant que foo@localhost et sélectionne
   bar comme base de données par défaut */
$link = mysqli_connect("localhost", "foo", "password", "bar");
?>
```

Voir aussi [mysqli_close](#) et [mysqli_real_connect](#) .

7.64.16 mysqli_data_seek() : Déplace le pointeur interne de résultat

void **mysqli_data_seek** (resource result , 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.

Note

Cette fonction ne peut être utilisée qu'avec des résultats obtenus avec la fonction [mysqli_store_result](#) .

Utilisation de [mysqli_data_seek](#)

```
<?php

/* Connexion à la base de données */
$link = mysqli_connect("localhost", "username", "password");
mysqli_select_db($link, "mydb");

/* Récupération de quelques lignes et stockage */
$query = "SELECT DINCTINCT name FROM employee ORDER BY name";
$result = mysqli_query($link, $query) or die(mysqli_error($link));

$rows = mysqli_store_result($result);

$total = mysqli_num_fields($rows);

if ($total > 0) { // il y'a au moins une ligne
    /* Obtenir le dernier employé */
    mysqli_data_seek($rows, mysqli_num_rows($result) -1);
    $employee = mysqli_fetch_row($rows);
    printf ("Nom de l'employé : %s\n", $employee[0]);
}

mysqli_free_result($rows);

?>
```

Voir aussi [mysqli_store_result](#) , [mysqli_fetch_row](#) et [mysqli_num_rows](#) .

7.64.17 mysqli_debug() : Effectue des actions de débogage

void **mysqli_debug** (string debug)

La fonction [mysqli_debug](#) est utilisée pour effectuer des actions de débogage en utilisant la librairie de débogage Fred Fish. Le paramètre `debug` est une chaîne de caractères représentant l'opération de débogage à effectuer.

Générer un journal de suivi

```
<?php

/* Crée un journal de suivi dans '/tmp/client.trace' sur la machine locale (client) */
mysqli_debug("d:t:0,/tmp/client.trace");

?>
```

7.64.18 mysqli_disable_reads_from_master() :

void **mysqli_disable_reads_from_master** (resource link)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.19 mysqli_disable_rpl_parse() :

```
void mysqli_disable_rpl_parse ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.20 mysqli_dump_debug_info() : Ecrit les informations de débogage dans le log

```
bool mysqli_dump_debug_info ( resource 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`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.64.21 mysqli_enable_reads_from_master() :

```
void mysqli_enable_reads_from_master ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.22 mysqli_enable_rpl_parse() :

```
void mysqli_enable_rpl_parse ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.23 mysqli_errno() : Retourne un code d'erreur pour la dernière commande MySQL

```
int mysqli_errno ( resource link )
```

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

Une liste complète des codes erreurs et leur signification peut être trouvée dans la section des constantes de la documentation MySQLi

Voir aussi [mysqli_error](#) .

7.64.24 mysqli_error() : Retourne une chaîne décrivant la dernière erreur

string **mysqli_error** (resource link)

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.

Utilisation de la fonction [mysqli_error](#)

```
<?php
/* Tentative de connexion échouée */
$host = "hote_inconnu";

$link = mysqli_connect($host, "username", "password")
    or die("Connexion impossible : " . mysqli_error());

?>
```

Voir aussi [mysqli_errno](#) .

7.64.25 mysqli_execute() : Exécute une requête préparée

int **mysqli_execute** (resource stmt)

La fonction [mysqli_execute](#) exécute une requête préparée en utilisant la fonction [mysqli_prepare](#) représentée par la ressource ^{stmt} . Lors de son exécution, tous les marqueurs de paramètres qui existent seront remplacés par les données appropriées.

Si la requête est un UPDATE, DELETE, ou INSERT, le nombre total de lignes affectées peut être déterminé en utilisant la fonction [mysqli_stmt_affected_rows](#) . De même, si la requête retourne un jeu de résultats, la fonction [mysqli_fetch](#) est utilisée.

Note

Lors de l'utilisation de la fonction [mysqli_execute](#) , la fonction [mysqli_fetch](#) doit être appelée pour récupérer les données avant toute autre requête.

Utilisation de [mysqli_execute](#)

```
<?php

/* Connexion au serveur */
$link = mysqli_connect("localhost", "user", "pass");
/* Activation du mode autocommit */
mysqli_autocommit($link, true);
/* Sélection de la base */
mysqli_select_db($link, "mabdd")

/* Prépare une requête d'insertion */
$query = "INSERT INTO matable VALUES(?, ?)";
$stmt = mysqli_prepare($link, $query);

$value_one = "hello";
$value_two = "world";

mysqli_bind_param($link, $value_one, $value_two);

/* Exécute la requête */
mysqli_execute($stmt);

/* Retourne le nombre de lignes affectées */
```

```

    $affected_rows = mysqli_stmt_affected_rows($stmt);

    /* Ferme la requête */
    mysqli_stmt_close($stmt);

    echo "Le nombre total de lignes affectées est $affected_rows";
?>

```

Voir aussi [mysqli_prepare](#) et [mysqli_bind_param](#) .

7.64.26 mysqli_fetch_array() : Retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux.

array **mysqli_fetch_array** (resource result , int resulttype)

Retourne un tableau qui correspond à la ligne lue ou `FALSE` s'il n'y a plus de lignes pour la connexion spécifiée par le paramètre `link` .

[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 clefs.

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 `result_type` 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.

Note

Les noms des champs retournés par cette fonction sont **sensibles à la casse** .

[mysqli_fetch_array](#) avec `MYSQLI_NUM`

```

<?php
    mysqli_connect("localhost", "mysql_user", "mysql_password") or
        die("Impossible de se connecter : " . mysqli_error());
    mysqli_select_db("mydb");

    $result = mysqli_query("SELECT id, name FROM mytable");

    while ($row = mysqli_fetch_array($result, MYSQLI_NUM)) {
        printf ("ID: %s  Name: %s", $row[0], $row[1]);
    }

    mysqli_free_result($result);
?>

```

[mysqli_fetch_array](#) avec `MYSQLI_ASSOC`

```

<?php
    mysqli_connect("localhost", "mysql_user", "mysql_password") or
        die("Impossible de se connecter : " . mysqli_error());
    mysqli_select_db("mydb");

    $result = mysqli_query("SELECT id, name FROM mytable");

```

```

while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
    printf ("ID: %s  Name: %s", $row["id"], $row["name"]);
}

mysqli_free_result($result);
?>

```

mysqli_fetch_array avec `MYSQLI_BOTH`

```

<?php
mysqli_connect("localhost", "mysql_user", "mysql_password") or
    die("Impossible de se connecter : " . mysqli_error());
mysqli_select_db("mydb");

$result = mysqli_query("SELECT id, name FROM mytable");

while ($row = mysqli_fetch_array($result, MYSQLI_BOTH)) {
    printf ("ID: %s  Name: %s", $row[0], $row["name"]);
}

mysqli_free_result($result);
?>

```

Voir aussi [mysqli_fetch_assoc](#) , [mysqli_fetch_row](#) et [mysqli_fetch_object](#) .

7.64.27 `mysqli_fetch_assoc()` : Récupère une ligne de résultat sous forme de tableau associatif

array `mysqli_fetch_assoc` (resource result)

Retourne un tableau associatif qui correspond à la ligne récupérée ou `FALSE` s'il n'y a plus de lignes.

La fonction [mysqli_fetch_assoc](#) est utilisée pour retourner un tableau associatif représentant la prochaine ligne dans le jeu de résultat représenté par le paramètre `result` , où chaque clef 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** .

Exemple étendu avec [mysqli_fetch_assoc](#)

```

<?php

$conn = mysqli_connect("localhost", "mysql_user", "mysql_password");

if (!$conn) {
    echo "Impossible de se connecter : " . mysqli_error();
    exit;
}

if (!mysqli_select_db("mydbname")) {
    echo "Impossible de sélectionner mydbname: " . mysqli_error();
    exit;
}

$sql = "SELECT id as userid, fullname, userstatus
      FROM   sometable
      WHERE  userstatus = 1";

$result = mysqli_query($sql);

if (!$result) {
    echo "Impossible d'exécuter la requête ($sql) : " . mysqli_error();
    exit;
}

```



```

if (mysqli_num_rows($result) == 0) {
    echo "Pas de lignes trouvées, je stoppe";
    exit;
}

// Tant qu'il y'a des lignes, les placer dans $row en tableau associatif
// Note : Si vous ne récupérez qu'une ligne, pas besoin de boucle
// Note : Si vous placez extract($row); dans la boucle, vous créez
//        alors $userid, $fullname, et $userstatus
while ($row = mysqli_fetch_assoc($result)) {
    echo $row["userid"];
    echo $row["fullname"];
    echo $row["userstatus"];
}

mysqli_free_result($result);

?>

```

Voir aussi [mysqli_fetch_array](#) , [mysqli_fetch_row](#) et [mysqli_fetch_object](#) .

7.64.28 mysqli_fetch_field_direct() : Récupère les données métas d'un champ unique

```
int mysqli_fetch_field_direct ( resource result , int offset )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.29 mysqli_fetch_field() : Retourne le prochain champ dans le jeu de résultats

```
object mysqli_fetch_field ( resource result )
```

[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. 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.

Attributs	Description
name	Le nom de la colonne
orgname	?
table	Le nom de la table d'où provient le champ (si non calculé)
orgtable	?
def	La valeur par défaut pour ce champ, en tant que chaîne de caractères
max_length	La taille maximale du champ dans le jeu de résultat
flags	Un entier représentant les bits pour ce champ
type	Le type de données utilisés pour ce champ
decimals	Le nombre de décimales utilisées (pour les champs numériques)

7.64.30 mysqli_fetch_fields() : Retourne un tableau d'objets représentant les champs dans le résultat

```
array mysqli_fetch_fields ( resource result )
```

Cette fonction fonctionne comme la fonction [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. Pour une description des attributs de chaque objet et leur signification, voyez la fonction [mysqli_fetch_field](#) .

7.64.31 `mysqli_fetch_lengths()` : Retourne les longueurs des colonnes de la ligne courante du jeu de résultats

```
array mysqli_fetch_lengths ( resource result )
```

La fonction `mysqli_fetch_lengths` retourne un tableau contenant les longueurs 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.

7.64.32 `mysqli_fetch_object()` : Retourne la ligne courante d'un jeu de résultat sous forme d'objet

```
object mysqli_fetch_object ( resource result )
```

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 résultat. Si il n'y a plus de lignes à lire, la fonction retourne `FALSE`.

Note

Les noms des champs retournés par cette fonction sont sensibles à la casse .

Voir aussi `mysqli_fetch_array`, `mysqli_fetch_assoc` et `mysqli_fetch_row`.

7.64.33 `mysqli_fetch_row()` : Récupère une ligne de résultat sous forme de tableau indexé

```
array mysqli_fetch_row ( resource result )
```

Retourne un tableau correspondant à la ligne récupérée ou `FALSE` s'il n'y a plus de lignes.

`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.

Voir aussi `mysqli_fetch_array`, `mysqli_fetch_assoc` et `mysqli_fetch_object`.

7.64.34 `mysqli_fetch()` : Récupère un résultat à partir d'une requête préparée dans les variables

```
int mysqli_fetch ( resource stmt )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.35 `mysqli_field_count()` : Retourne le nombre de colonnes pour la dernière requête

```
int mysqli_field_count ( resource link )
```

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 non-vide ou non, sans en connaître la nature.

7.64.36 `mysqli_field_seek()` : Déplace le pointeur de résultat sur le champ spécifié

```
int mysqli_field_seek ( resource link , int fieldnr )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.37 `mysqli_field_tell()` : Retourne la position du champ courant dans un pointeur de résultat

```
int mysqli_field_tell ( resource result )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.38 `mysqli_free_result()` : Libère la mémoire associée à un résultat

```
int mysqli_free_result ( resource result )
```

La fonction `mysqli_free_result` libère la mémoire associée au résultat représenté par le paramètre `result`.

7.64.39 `mysqli_get_client_info()` : Retourne une chaîne contenant la version du client MySQL

```
string mysqli_get_client_info ( void )
```

La fonction `mysqli_get_client_info` est utilisée pour retourner une chaîne de caractères représentant la version du client utilisé par l'extension MySQLi.

7.64.40 `mysqli_get_host_info()` : Retourne une chaîne contenant le type de connexion utilisé

```
string mysqli_get_host_info ( resource link )
```

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).

7.64.41 `mysqli_get_proto_info()` : Retourne la version du protocole MySQL utilisé

```
int mysqli_get_proto_info ( resource link )
```

Retourne un entier représentant la version du protocole MySQL utilisé par la connexion représentée par le paramètre `link`.

7.64.42 `mysqli_get_server_info()` : Retourne la version du serveur MySQL

```
string mysqli_get_server_info ( resource link )
```

Retourne une chaîne de caractères représentant la version du serveur MySQL auquel l'extension MySQLi est connectée (représenté par le paramètre `link`).

7.64.43 `mysqli_get_server_version()` : Retourne un entier représentant la version du serveur MySQL

```
int mysqli_get_server_version ( resource link )
```

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).

7.64.44 `mysqli_info()` : Retourne des informations à propos de la dernière requête exécutée

```
string mysqli_info ( resource link )
```

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 <code>FALSE</code> .

7.64.45 `mysqli_init()` : Initialise MySQLi et retourne une ressource à utiliser avec `mysqli_real_connect`

```
resource mysqli_init ( void )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.46 `mysqli_insert_id()` : Retourne l'identifiant automatiquement généré par la dernière requête

```
mixed mysqli_insert_id ( resource link )
```

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 INSERT ou UPDATE utilisant la fonction LAST_INSERT_ID() modifiera aussi la valeur retournée par la fonction <code>mysqli_insert_id</code> .

7.64.47 `mysqli_kill()` : Demande au serveur de terminer un thread MySQL

```
bool mysqli_kill ( resource link , 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`.

7.64.48 `mysqli_master_query()` : Force l'exécution d'une requête sur le maître dans une configuration maître/esclave (master/slave)

```
bool mysqli_master_query ( resource link , string query )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.49 `mysqli_num_fields()` : Retourne le nombre de champs dans un résultat

```
int mysqli_num_fields ( resource result )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.50 `mysqli_num_rows()` : Retourne le nombre de lignes dans un résultat

```
int mysqli_num_rows ( resource result )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.51 mysqli_options() : Modifie les options

```
bool mysqli_options ( resource link , int flags , mixed values )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.52 mysqli_param_count() : Retourne le nombre de paramètres pour la requête donnée

```
int mysqli_param_count ( resource stmt )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.53 mysqli_ping() : Ping la connexion au serveur et reconnecte si elle n'existe plus

```
int mysqli_ping ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.54 mysqli_prepare_result() :

```
resource mysqli_prepare_result ( resource stmt )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.55 mysqli_prepare() : Prépare une requête SQL pour l'exécution

```
resource mysqli_prepare ( resource link , string query )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.56 mysqli_profiler() :

```
bool mysqli_profiler ( int flags , string info , int port )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.57 mysqli_query() : Performe une requête sur la base de données

```
resource mysqli_query ( resource link , string query , int resultmode )
```

La fonction [mysqli_query](#) est utilisée pour simplifier l'acte de performer 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.

7.64.58 mysqli_read_query_result() :

```
bool mysqli_read_query_result ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.59 mysqli_real_connect() : Ouvre une connexion à un serveur MySQL

```
bool mysqli_real_connect ( resource link , string hostname , string username , string passwd , string dbname , int port , string socket )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Voir aussi [mysqli_connect](#) et [mysqli_close](#).

7.64.60 mysqli_real_escape_string() : 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

```
string mysqli_real_escape_string ( resource link , string escapestr )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Voir aussi [mysqli_character_set_name](#).

7.64.61 mysqli_real_query() : Exécute une requête SQL

```
bool mysqli_real_query ( resource link , 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 [mysqli_field_count](#) .

7.64.62 mysqli_reload() :

```
bool mysqli_reload ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.63 mysqli_rollback() :

```
bool mysqli_rollback ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.64 mysqli_rpl_parse_enabled() :

```
int mysqli_rpl_parse_enabled ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.65 mysqli_rpl_probe() :

```
bool mysqli_rpl_probe ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.66 mysqli_rpl_query_type() :

```
int mysqli_rpl_query_type ( string query )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.67 mysqli_select_db() : Sélectionne une base de données par défaut pour les requêtes

```
bool mysqli_select_db ( resource 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` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.64.68 `mysqli_send_long_data()` :

```
bool mysqli_send_long_data ( resource stmt , int param_nr , string data )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.69 `mysqli_send_query()` :

```
bool mysqli_send_query ( resource link , string query )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.70 `mysqli_slave_query()` : Force l'exécution d'une requête sur un esclave dans une configuration maître/esclave (master/slave)

```
bool mysqli_slave_query ( resource link , string query )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.71 `mysqli_ssl_set()` :

```
string mysqli_ssl_set ( resource link , 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.

7.64.72 `mysqli_stat()` : Obtient le statut courant du système

```
string mysqli_stat ( resource link )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.73 `mysqli_stmt_affected_rows()` :

```
mixed mysqli_stmt_affected_rows ( object stmt )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.74 mysqli_stmt_close() : Ferme une requête

bool **mysqli_stmt_close** (resource stmt)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.75 mysqli_stmt_errno() :

int **mysqli_stmt_errno** (resource stmt)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.76 mysqli_stmt_error() :

string **mysqli_stmt_error** (resource stmt)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.77 mysqli_stmt_store_result() :

resource **mysqli_stmt_store_result** (resource stmt)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.78 mysqli_store_result() : Transfère un jeu de résultats à partir de la dernière requête

resource **mysqli_store_result** (resource link)

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.

7.64.79 mysqli_thread_id() : Retourne l'identifiant du thread pour la connexion courante

int **mysqli_thread_id** (resource link)

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.

7.64.80 [mysqli_thread_safe\(\)](#) : Indique si le thread safety est donné ou non

bool [mysqli_thread_safe](#) (void)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.64.81 [mysqli_use_result\(\)](#) : Initialise la récupération d'un jeu de résultats

resource [mysqli_use_result](#) (resource link)

[mysqli_use_result](#) est utilisée pour initialiser la récupération d'un jeu de résultats à partir de la dernière requête exécuté 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 de la prochaine requête sur la connexion à la base de données.

Note

La fonction [mysqli_use_result](#) ne transfère pas le jeu de résultat en entier à partir de la base de données et on ne peut donc pas utiliser des fonctions telles que [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](#) .

Voir aussi [mysqli_real_query](#) et [mysqli_store_result](#) .

7.64.82 [mysqli_warning_count\(\)](#) : Retourne le nombre d'avertissements généré par la dernière requête

resource [mysqli_warning_count](#) (resource link)

[mysqli_warning_count](#) retourne le nombre d'avertissements généré par la dernière requête de la connexion représentée par le paramètre ^{link} .

7.65 Sessions Mohawk

7.65.1 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 cohérentes pour une ferme de serveurs PHP. Plus d'informations sur les msession et le démon sont disponible sur le site <http://devel.mohawksoft.com/msession.html> .

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.65.2 Pré-requis

7.65.3 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.

7.65.4 Configuration à l'exécution

7.65.5 Types de ressources

7.65.6 Constantes prédéfinies

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [msession_connect](#)
- [msession_count](#)
- [msession_create](#)
- [msession_destroy](#)
- [msession_disconnect](#)
- [msession_find](#)
- [msession_get_array](#)
- [msession_get](#)
- [msession_getdata](#)
- [msession_inc](#)
- [msession_list](#)
- [msession_listvar](#)
- [msession_lock](#)
- [msession_plugin](#)
- [msession_randstr](#)

- [msession_set_array](#)
- [msession_set](#)
- [msession_setdata](#)
- [msession_timeout](#)
- [msession_uniq](#)
- [msession_unlock](#)

7.65.8 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.

7.65.9 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.

7.65.10 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.

7.65.11 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.

7.65.12 msession_find() : Trouve une valeur

```
array msession_find ( string name , string value )
```

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.65.13 msession_get_array() : Lit un tableau de ... (??)

```
array msession_get_array ( string session )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.65.14 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.

7.65.15 msession_getdata() : Lit les données ... (??)

```
string msession_getdata ( string session )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.65.16 msession_inc() : Incrémente 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.

7.65.17 msession_list() : Liste ... (??)

```
array msession_list ( void )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.65.18 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.

7.65.19 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.

7.65.20 msession_plugin() : Appele 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.

7.65.21 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.

7.65.22 msession_set_array() : Modifie un tableau de ... (??)

```
bool msession_set_array ( string session , array tuples )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.65.23 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.

7.65.24 msession_setdata() : Modifie les données (??)

```
bool msession_setdata ( string session , string value )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.65.25 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.

7.65.26 msession_uniq() : Lit un identifiant unique

`string msession_uniq ( int param )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.65.27 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.

7.66 Muscat

7.66.1 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.

7.66.2 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](#)

7.66.4 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.

`muscat_get` lit une ligne de réponse de l'API muscat. Elle retourne `FALSE` lorsqu'il n'y a plus rien à lire. Il ne faut pas confondre ce résultat avec une chaîne vide. Utilisez `=== FALSE` et `!== FALSE` pour faire la différence entre une réponse vide et la fin de la réponse.

7.66.5 muscat_give() : Envoie une chaîne à l'API de base muscat

```
int 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.

7.66.6 muscat_setup_net() : Crée une nouvelle session muscat

resource **muscat_setup_net** (string muscat_host , int port)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

`muscat_setup_net` crée une nouvelle session muscat, et retourne une ressource de connexion. `muscat_host` est le nom de l'hôte serveur et `port` est le numéro de port ou se connecter : en fait, cette fonction prend les mêmes arguments que `fsockopen` .

7.66.7 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"` , par défaut.

7.67 Réseau

7.67.1 Introduction

7.67.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.67.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.67.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>define_syslog_variables</code>	"0"	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, voyez [ini_set](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`define_syslog_variables` booléen

Définit si 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`.

7.67.5 Types de ressources

Cette extension ne définit aucune ressource.

7.67.6 Constantes prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

Constante	Description
<code>LOG_CONS</code>	Si 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_PERROR</code>	Affiche aussi le message sur la sortie standard
<code>LOG_PID</code>	Ajoute le PID dans chaque message

Constante	Description
<code>LOG_EMERG</code>	Le système est inutilisable
<code>LOG_ALERT</code>	Une mesure doit être prise immédiatement
<code>LOG_CRIT</code>	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	Tout 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 champs.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [checkdnsrr](#)
- [closelog](#)
- [debugger_off](#)
- [debugger_on](#)
- [define_syslog_variables](#)
- [dns_check_record](#)
- [dns_get_mx](#)
- [dns_get_record](#)
- [fsockopen](#)
- [gethostbyaddr](#)
- [gethostbyname](#)
- [gethostbyname_l](#)
- [getmxrr](#)
- [getprotobyname](#)
- [getprotobynumber](#)
- [getservbyname](#)
- [getservbyport](#)
- [ip2long](#)
- [long2ip](#)
- [openlog](#)
- [pfsockopen](#)
- [socket_get_status](#)
- [socket_set_blocking](#)
- [socket_set_timeout](#)
- [syslog](#)

7.67.8 closelog() : Ferme la connexion à l'historique système

```
int 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](#) .

7.67.9 `debugger_off()` : Désactive le debugger interne de PHP.

```
int debugger_off ( void )
```

[debugger_off](#) désactive le debugger interne de PHP. Le debugger est toujours en cours de développement.

7.67.10 `debugger_on()` : Active le debugger interne de PHP

```
int debugger_on ( string address )
```

[debugger_on](#) active le debugger interne de PHP, et le connecte à l'adresse `address` . Le debugger est toujours en cours de développement.

7.67.11 `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](#) .

7.67.12 `dns_check_record()` : Synonyme de [checkdnsrr](#)

```
int dns_check_record ( string host , string type )
```

[dns_check_record](#) vérifie les enregistrements DNS correspondant à l'hôte internet `host` , donnés sous forme d'adresse IP ou de nom Internet.

7.67.13 `dns_get_mx()` : Synonyme de [getmxrr](#)

```
int dns_get_mx ( string hostname , array mxhosts , array &weight )
```

[dns_get_mx](#) lit le champ MX de l'hôte internet `hostname` .

7.67.14 `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 plate-formes Windows. Utilisez alors la classe [PEAR Net_DNS](#) .

dns_get_record retourne un tableau associatif, contenant **au minimum** les index suivants : `host`, `type`, `class`, `ttl`.

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 par exemple `www.exemple.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_ALL` et `DNS_ANY`. Par défaut, il vaut **DNS_ANY**.

Note
A cause des performances excentriques de la librairie libresolv, suivant les plate-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 `addtl` sont passé 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.

Suivant le type de valeur de l'index `type`, le tableau associatif peut aussi contenir les index suivants : `ip`, `pri`, `target`, `cpu`, `os`, `mname`, `rname`, `serial`, `refresh`, `retry`, `expire`, `minimum-ttl`, `txt`.

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 sur 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 seconde) d'attente après un rafraîchissement échoué, avant de faire une seconde tentative. <code>expire</code> : Durée maximale (en seconde) de conservation d'une copie des données de zone sans pouvoir faire de rafraîchissement. <code>minimum-ttl</code> : Durée minimale (en seconde) pour 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
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 targets. <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 utilisateur.hote (par exemple : <code>webmestre.exemple.com</code> au contraire du format <code>webmestre@exemple.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>
<pre><?php \$result = dns_get_record("php.net"); print_r(\$result); ?></pre>
/*

Produit un affichage similaire à ceci :

```
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,
   pu 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,$addtl);
print "Résultat = ";
print_r($result);
print "Autorité NS = ";
print_r($authns);
print "Additionnel = ";
print_r($addtl);
?>
```

/*
Produit un affichage similaire à ceci :

```
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](#)

7.67.15 fsockopen() : Ouvre une socket de connexion Internet ou Unix

```
int fsockopen ( string target , int port , int errno , string errstr , float timeout )
```


`fsocketopen` crée un flût de connexion à l'Internet (AF_INET) ou à un domaine Unix (AF_UNIX). Via Internet, cette fonction va ouvrir une socket de connexion TCP avec l'hôte `target` sur le port `port`. Pour les connexions UDP, vous devez explicitement spécifier le protocole : `udp://target`. Via un domaine Unix, `target` représente le chemin jusqu'à la socket, et `port` doit être mis à 0. L'option `timeout` sert à donner une durée maximale à cet appel.

`fsocketopen` retourne un pointeur de fichier qui peut être utilisé avec d'autres fonctions fichiers, telles que `fgets`, `fgetss`, `fputs`, `fclose` et `feof`.

Si l'appel échoue, `fsocketopen` 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 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 : `socket_set_blocking`.

Exemple avec `fsocketopen`

```
<?php
$fp = fsocketopen("www.php.net", 80, &$errno, &$errstr, 30);
if (!$fp) {
    echo "$errstr ($errno)<br>\n";
} else {
    fputs($fp, "GET / HTTP/1.0\n\n");
    while(!feof($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
$fp = fsocketopen("udp://127.0.0.1", 13, &$errno, &$errstr);
if (!$fp) {
    echo "ERREUR: $errno - $errstr<br>\n";
} else {
    fwrite($fp, "\n");
    echo fread($fp, 26);
    fclose($fp);
}
?>
```

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 `fsocketopen`, `socket_set_blocking`, `socket_set_timeout`, `fgets`, `fgetss`, `fputs`, `fclose` et `feof`.

7.67.16 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']);

print $hostname;
?>
```

Voir aussi [gethostbyname](#) .

7.67.17 gethostbyname() : Retourne l'adresse IP correspondant à un hôte

string **gethostbyname** (string hostname)

[gethostbyname](#) retourne l'adresse IP correspondant à l'hôte `hostname` .

Exemple avec [gethostbyaddr](#)

```
<?php
$ip = gethostbyname('www.exemple.com');

print $ip;
?>
```

Voir aussi [gethostbyaddr](#) .

7.67.18 gethostbyname_l() : Retourne la liste d'IP correspondante à un hôte.

array **gethostbyname_l** (string hostname)

[gethostbyname_l](#) retourne la liste d'IP correspondant à l'hôte `hostname` .

Exemple avec [gethostbyname_l](#)

```
<?php
$notes = gethostbyname_l('www.yahoo.com');
print_r($notes);
?>
```

Le résultat du script ci-dessus sera :

```
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.

7.67.19 getmxrr() : Retourne les enregistrements MX d'un hôte

```
int 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'adresse. Seuls le serveur de mail trouvé dans les enregistrements DNS sont retournés. Suivant la [RFC 974](#) 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 [checkdnsrr](#), [gethostbyname](#), [gethostbyname_l](#), [gethostbyaddr](#), et la page 8 du manuel.

7.67.20 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`.

Voir aussi [getprotobynumber](#).

7.67.21 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](#).

7.67.22 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`.

Voir aussi [getservbyport](#) .

7.67.23 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` .

Voir aussi [getservbyname](#) .

7.67.24 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.

Exemple ip2long
<pre><?php \$ip = gethostbyname("www.php.net"); \$out = "Les URLS suivantes sont équivalentes :
\n"; \$out .= "http://www.php.net/, http://". \$ip ."/, et http://". ip2long(\$ip) ."/
\n"; echo \$out; ?></pre>

Ce second exemple montre comment afficher une adresse convertie à l'aide de la fonction [printf](#) :

Affichage d'adresse IP
<pre><?php \$ip = gethostbyname("www.php.net"); printf ("%u\n", ip2long (\$ip)); echo \$out; ?></pre>

Voir aussi [long2ip](#) et [sprintf](#) .

7.67.25 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](#)

7.67.26 openlog() : Ouvre la connexion à l'historique système

```
int 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. L'utilisation de [openlog](#) est optionnelle; cette fonction sera automatiquement appelée par [syslog](#) si nécessaire, et dans ce cas, l'identification sera mise par défaut à `FALSE`. `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 optionnel. 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 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	réserve pour utilisation local. 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 générique
LOG_UUCP	UUCP subsystem

Voir aussi [define syslog variables](#), [syslog](#) et [closelog](#).

7.67.27 pfsockopen() : Ouvre une socket de connexion Internet ou Unix persistante.

```
int pfsockopen ( string hostname , int port , int errno , string errstr , int 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](#).

7.67.28 socket_get_status() : Retourne les informations sur une socket

```
array socket_get_status ( resource socket_get_status )
```

`socket_get_status` est un [alias](#) de [stream_get_meta_data](#).

7.67.29 socket_set_blocking() : Active/désactive le mode bloquant d'une socket

```
int socket_set_blocking ( int socket_descriptor , int mode )
```

Si `mode` est `FALSE`, la socket est mise en mode non bloquant, et si il est `TRUE`, la socket est mise en mode bloquant. Cela affecte des appels tels que `fgets` qui lisent depuis une socket. En mode non bloquant, un appel `fgets` retournera immédiatement toujours `TRUE` tandis qu'en mode bloquant, elle va attendre que des données arrivent pour répondre `TRUE`.

7.67.30 `socket_set_timeout()` : Alias de `stream_set_timeout`

```
bool socket_set_timeout ( int socket_descriptor , int seconds , int microseconds )
```

`socket_set_timeout` est un alias de `stream_set_timeout`.

7.67.31 `syslog()` : Génère un message dans l'historique système

```
int 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
LOG_EMERG	système inutilisable
LOG_ALERT	une décision doit être prise immédiatement
LOG_CRIT	conditions critiques
LOG_ERR	conditions d'erreur
LOG_WARNING	conditions d'alerte
LOG_NOTICE	condition normale, mais significative
LOG_INFO	message d'information
LOG_DEBUG	message de débogage

Utilisation de `syslog`

```
<?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("myScripLog", 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 $REMOTE_ADDR ($HTTP_USER_AGENT)");
}

closelog();
?>
```

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 `LOG_LOCAL0` à `LOG_LOCAL7` pour le paramètre `facility` de la fonction [openlog](#) n'est pas disponible sous Windows.

Voir aussi [define_syslog_variables](#), [openlog](#) et [closelog](#).

7.68 Ncurses terminal screen control functions

7.68.1 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, supporte les pavé numériques, les couleurs, les colorations multiples les caractères de formulaire, et les touches 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.

Ncurses est disponible sur les plate-formes suivantes :

- AIX
- BeOS
- Cygwin
- Digital Unix (aka OSF1)
- FreeBSD
- GNU/Linux
- HPUX
- IRIX
- OS/2
- SCO OpenServer
- Solaris
- SunOS

7.68.2 Pré-requis

Vous devez disposer des librairies 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.

7.68.3 Installation

Pour installer cette extension, compilez ^{PHP} en mode CGI ou CLI, avec l'option `--with-ncurses[=DIR]` .

7.68.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<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 [ini_set](#) .

7.68.5 Types de ressources

Cette extension ne définit aucune ressource.

7.68.6 Constantes prédefinies

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.

7.68.6.1 Error codes

On error ncurses functions return NCURSES_ERR.

7.68.6.2 Colors

Constante	Signification
NCURSES_COLOR_BLACK	Pas de couleur
NCURSES_COLOR_WHITE	blanc
NCURSES_COLOR_RED	rouge - supporté lorsque le terminal est en mode couleurs
NCURSES_COLOR_GREEN	vert - supporté lorsque le terminal est en mode couleurs
NCURSES_COLOR_YELLOW	jaune - supporté lorsque le terminal est en mode couleurs
NCURSES_COLOR_BLUE	bleu - supporté lorsque le terminal est en mode couleurs
NCURSES_COLOR_CYAN	cyan - supporté lorsque le terminal est en mode couleurs
NCURSES_COLOR_MAGENTA	magenta - supporté lorsque le terminal est en mode couleurs

7.68.6.3 Keys

constant	meaning
NCURSES_KEY_F0 - NCURSES_KEY_F64	Touches de fonctions F1 - F64
NCURSES_KEY_DOWN	flèche vers le bas
NCURSES_KEY_UP	flèche vers le haut
NCURSES_KEY_LEFT	flèche vers la gauche
NCURSES_KEY_RIGHT	flèche vers la droite
NCURSES_KEY_HOME	touche home (vers le haut + flèche vers la gauche)
NCURSES_KEY_BACKSPACE	retour en arrière
NCURSES_KEY_DL	efface une ligne
NCURSES_KEY_IL	insère une ligne
NCURSES_KEY_DC	efface un caractère
NCURSES_KEY_IC	insère un caractère, ou bien passe en mode insertion
NCURSES_KEY_EIC	termine le mode insertion
NCURSES_KEY_CLEAR	efface l'écran
NCURSES_KEY_EOS	efface l'écran jusqu'en bas
NCURSES_KEY_EOL	efface l'écran jusqu'à la fin de la ligne
NCURSES_KEY_SF	scrolle une ligne vers le bas
NCURSES_KEY_SR	scroll 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)

Manuel PHP 4.3.0 – Version Française

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	rafraichis
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_SMMESSAGE	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) relâché
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 prédefinies](#)
- [ncurses_addch](#)
- [ncurses_addchnstr](#)
- [ncurses_addchstr](#)
- [ncurses_addnstr](#)
- [ncurses_addstr](#)
- [ncurses_assume_default_colors](#)
- [ncurses_attroff](#)
- [ncurses_attron](#)
- [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_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_phoutrefresh](#)
- [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](#)

7.68.8 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.

7.68.9 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.

7.68.10 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.

7.68.11 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.

7.68.12 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.

7.68.13 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.

7.68.14 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.

7.68.15 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.

7.68.16 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.

7.68.17 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](#) .

7.68.18 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.

7.68.19 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.

7.68.20 ncurses_border() : Dessine un bord autour de l'écran avec les caractères données

```
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.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.21 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.

7.68.22 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.

7.68.23 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 flût de caractères de contrôles sont intouchées), rendant les caractères tapés par l'utilisateur directement accessible au programme.

[`ncurses\_cbreak`](#) retourne `TRUE` en cas de succès, et `NCURSES_ERR` en cas d'erreur.

Voir aussi [`ncurses\_nocbreak`](#).

7.68.24 `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 représente la couleur de fond d'écran. Pour effacer le fond d'écran avec les blancs, utilisez plutôt [`ncurses\_erase`](#).

Voir aussi [`ncurses\_erase`](#).

7.68.25 `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 sont créés avec la fonction [`ncurses\_clrtoebot`](#) pour avoir le bon rendu de fond d'écran. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [`ncurses\_clear`](#) et [`ncurses\_clrtoeol`](#).

7.68.26 `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 sont créés par la fonction `ncurses_clrtoeol` pour avoir le bon rendu de fond d'écran. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi `ncurses_clear` et `ncurses_clrtoeol`

7.68.27 `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.

7.68.28 `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.

7.68.29 `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.

7.68.30 `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` .

7.68.31 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` .

7.68.32 ncurses_define_key() : Définit 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.

7.68.33 ncurses_del_panel() : Retire un conteneur de la pile et l'efface (mais pas la fenêtre associée)

```
int ncurses_del_panel ( resource panel )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.34 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.

7.68.35 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 .

7.68.36 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 sousjacentes sont alors remontée 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 .

7.68.37 ncurses_delwin() : Efface une fenêtre ncurses

int 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.

7.68.38 ncurses_doupdate() : Affiche tout ce qui est prêt, et rafraichit 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 rafraichissements multiples. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.68.39 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` .

7.68.40 ncurses_echochar() : Affiche un caractère et rafraichit 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.

7.68.41 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.

7.68.42 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, configuré 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

7.68.43 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 .

7.68.44 ncurses_filter() : Filtre pour ncurses

int **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.

7.68.45 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](#) .

7.68.46 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.

7.68.47 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.

7.68.48 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.

7.68.49 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'événement 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`

```
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` .

7.68.50 `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.

7.68.51 `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.

7.68.52 `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` .

7.68.53 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` .

7.68.54 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 terinal. `ncurses_has_il` retourne ^{TRUE} si le terminal a ces capacités, et sinon ^{FALSE} .

Voir aussi `ncurses_has_ic` .

7.68.55 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.

7.68.56 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.

7.68.57 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.

7.68.58 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.

ncurses_inch retourne le caractère de la position courante.

7.68.59 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.

7.68.60 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.

7.68.61 ncurses_init() : Initialise ncurses

```
int ncurses_init ( 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.

7.68.62 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 )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.68.63 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 )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.68.64 ncurses_insertln() : Insère une ligne et décale le reste de l'écran vers le bas

```
bool ncurses_insertln ( 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_insertln](#) insère une nouvelle ligne au dessus de la ligne courante. La ligne du bas de l'écran sera perdue.

7.68.65 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.

7.68.66 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.

7.68.67 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](#) .

7.68.68 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.

7.68.69 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.

7.68.70 ncurses_killchar() : Retourne la ligne du caractère actuellement supprimé

```
bool 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 .

7.68.71 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 .

7.68.72 ncurses_meta() : Active/désactive les informations de méta clé 8-bits

```
long ncurses_meta ( resource window , bool 8bit )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.73 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.

7.68.74 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.

7.68.75 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

Voir aussi [ncurses_getmouse](#) , [ncurses_ungetmouse](#) et [ncurses_getch](#) .

Exemple avec [ncurses_mousemask](#)

```
$newmask = NCURSES_BUTTON1_CLICKED + NCURSES_BUTTON1_RELEASED;
$mask = ncurses_mousemask($newmask, &$oldmask);
if ($mask & $newmask){
    printf ("Toutes les options de souris spécifiées sont supportées\n");
}
```

7.68.76 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.

7.68.77 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.

7.68.78 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.

7.68.79 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.

7.68.80 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.

7.68.81 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.

7.68.82 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.

7.68.83 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.

7.68.84 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.

7.68.85 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.

7.68.86 ncurses_mvline() : Choisi une nouvelle position et dessine une ligne horizontale avec le caractère donné, et 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.

7.68.87 ncurses_mvinch() : Déplace la position et lit le caractère 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.

7.68.88 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.

7.68.89 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.

7.68.90 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.

7.68.91 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.

7.68.92 ncurses_newpad() : Crée une 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.

7.68.93 ncurses_newwin() : Crée une nouvelle fenêtre

```
int 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.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.94 ncurses_nl() : Convertit les nouvelles lignes et retour chariot en saut 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.

7.68.95 ncurses_nocbreak() : Passe le terminal en mode cooked

`bool 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](#) .

7.68.96 ncurses_noecho() : Désactive l'écho clavier

`bool 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](#)

7.68.97 ncurses_nonl() : Ne convertit pas les nouvelles lignes et retour chariot en saut de ligne

`bool 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.

7.68.98 ncurses_noqiflush() : Ne vide pas les buffers sur reception des caractères de signaux

```
int 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.

7.68.99 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 passé au programme. La différence est qu'en mode brut, les caractères d'interruption, de fin de programme, de suspension et de suivi du flût de caractères sont passés sans être interprété. ncurses_noraw retourne ^{FALSE} en cas de succès, et ^{TRUE} sinon.

Voir aussi ncurses_raw , ncurses_cbreak et ncurses_nocbreak

7.68.100 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.

7.68.101 ncurses_panel_above() : Retourne le conteneur au dessus du conteneur. Si ce dernier est null, retourne le dernier conteneur de la pile

```
int ncurses_panel_above ( resource panel )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.102 ncurses_panel_below() : Retour le conteneur sous le conteneur. Si ce dernier est null, le premier conteneur de la pile est retourné

```
int ncurses_panel_below ( resource panel )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.103 ncurses_panel_window() : Retourne la fenêtre associée avec un conteneur

```
int ncurses_panel_window ( resource panel )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.104 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.

7.68.105 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.

7.68.106 ncurses_putp() : ncurses_putp (???)

```
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.

7.68.107 ncurses_qiflush() : Vide les buffers après détection d'un caractère de signal

```
int 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.

7.68.108 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 passé au programme. La différence est qu'en mode brut, les caractères d'interruption, de fin de programme, de suspension et de suivi du flût de caractères sont passés sans être interprété. `ncurses_raw` retourne `FALSE` en cas de succès, et `TRUE` sinon.

Voir aussi `ncurses_noraw` , `ncurses_cbreak` et `ncurses_nocbreak`

7.68.109 ncurses_refresh() : Rafraichit 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.

7.68.110 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.

7.68.111 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.

7.68.112 ncurses_reset_shell_mode() : Restaure le mode shell mode sauvé ar def_shell_mode

```
int ncurses_reset_shell_mode ( void )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.113 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](#) .

7.68.114 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 rétablit avec la fonction [ncurses_resetty](#) . [ncurses_savetty](#) retourne toujours `FALSE` .

Voir aussi [ncurses_resetty](#) .

7.68.115 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.

7.68.116 ncurses_scr_init() : Initilise 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.

7.68.117 ncurses_scr_restore() : Rétablit un écran depuis une 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.

7.68.118 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.

7.68.119 ncurses_scrll() : Scrolle le contenu de la fenêtre vers le haut ou le bas, dans changer la position courante

```
int ncurses_scrll ( 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.

7.68.120 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.

7.68.121 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.

7.68.122 ncurses_slk_attroff() : ncurses_slk_attroff

```
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.

7.68.123 ncurses_slk_attron() : ncurses_slk_attron

```
int ncurses_slk_attron ( 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.

7.68.124 ncurses_slk_attrset() : ncurses_slk_attrset

```
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.

7.68.125 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.

7.68.126 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.

7.68.127 ncurses_slk_init() : Initialises 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 soit utilisées. Si ncurses_initscr utilisent 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 de 0, vous utiliserez un arrangement de type 3-2-3; la valeur de 1 indiquera un arrangement 4-4; la valeur de 2 indiquera un mode 4-4-4, mais une ligne d'index sera créée en plus.

7.68.128 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.

7.68.129 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.

7.68.130 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 .

7.68.131 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.

7.68.132 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`.

7.68.133 `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.

7.68.134 `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.

7.68.135 `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.

7.68.136 `ncurses_termattrs()` : Retourne toutes les options supportées par le terminal, combinées ensemble par l'opérateur OR

```
bool ncurses_termattrs ( 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.

7.68.137 ncurses_termname() : Retourne le nom court du terminal

```
string ncurses_termname ( 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_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 .

7.68.138 ncurses_timeout() : Configure le temps d'expiration des des séquences clavier spéciales

```
void ncurses_timeout ( int millisec )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.68.139 ncurses_top_panel() : Place un conteneur visible sur le haut de la pile

```
int ncurses_top_panel ( resource panel )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.140 ncurses_typeahead() : Spécifie un autre descripteur de fichier pour la vérification à la volée des données saisies

```
int ncurses_typeahead ( int fd )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.68.141 ncurses_ungetch() : Replace 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.

7.68.142 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, spécifié dans ^{mevent}. Les options d'événement 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 .

7.68.143 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.

7.68.144 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.

7.68.145 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.

7.68.146 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.

7.68.147 ncurses_vidattr() : ncurses_vidattr

```
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.

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.

7.68.148 ncurses_vline() : Dessine une ligne verticale à la position courante en utilisant un caractère donnée, 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.

7.68.149 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.

7.68.150 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.

7.68.151 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.

7.68.152 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.

7.68.153 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.

7.68.154 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.

7.68.155 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.

7.68.156 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.

7.68.157 ncurses_werase() : Efface le contenu de la fenêtre

```
long ncurses_werase ( resource window )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.68.158 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.

7.68.159 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.

7.68.160 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.

7.68.161 ncurses_wmove() : Déplace le 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.

7.68.162 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.

7.68.163 ncurses_wrefresh() : Rafraichit 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.

7.68.164 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.

7.68.165 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.

7.68.166 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.

7.69 Lotus Notes

7.69.1 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](#)
- [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](#)

7.69.3 notes_copy_db() : Crée une note à partir du formulaire form_name

```
string 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.

7.69.4 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.

7.69.5 notes_create_note() : Crée une note à partir du formulaire form_name

```
string 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.

7.69.6 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.

7.69.7 notes_find_note() : Retourne un identifiant de note, dans le serveur database_name. Spécifie le nom de la note.

```
bool 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.

7.69.8 notes_header_info() : Ouvre le message msg_number dans la boîte aux lettres et le serveur spécifié

```
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.

7.69.9 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.

7.69.10 notes_mark_read() : Marque la note note_id comme lue par l'utilisateur user_name

```
string 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.

7.69.11 notes_mark_unread() : Marque la note note_id comme non lue pour l'utilisateur user_name

```
string 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.

7.69.12 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.

7.69.13 notes_search() : Cherche les notes par mots clés dans database_name

```
string 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.

7.69.14 notes_unread() : Retourne l'identifiant de la note non lue pour l'utilisateur user_name

```
string 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.

7.69.15 notes_version() : Lit la version de Lotus Notes

```
string 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.

7.70 Netscape Server API

7.70.1 Introduction

Ces fonctions ne sont disponibles que lorsque vous utilisez ^{PHP} comme module NSAPI avec les serveurs Netscape/iPlanet/SunONE.

7.70.2 Installation

Pour l'installation de ^{PHP} sur les serveurs Netscape/iPlanet/SunONE voyez la section NSAPI dans le chapitre d'installation.

7.70.3 Configuration à l'exécution

Le comportement du module PHP NSAPI PHP 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	Fonction
<code>nsapi.read_timeout</code>	60	PHP_INI_ALL	Configure la durée d'attente des données de POST envoyées par le client.

7.70.4 Types de ressources

Cette extension ne définit aucune ressource.

7.70.5 Constantes prédefinies

Cette extension ne définit aucune constante.

7.70.6 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
<code>apache_request_headers</code>	<code>nsapi_request_headers</code>	Lit tous les entêtes de requête HTTP
<code>apache_response_headers</code>	<code>nsapi_response_headers</code>	Lit tous les entêtes de réponse HTTP
<code>getallheaders</code>	<code>nsapi_request_headers</code>	Lit tous les entêtes de requête HTTP
<code>virtual</code>	<code>nsapi_virtual</code>	Fait une sous requête NSAPI

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Voir aussi](#)
- [nsapi_request_headers](#)
- [nsapi_response_headers](#)

- [nsapi_virtual](#)

7.70.8 nsapi_response_headers() : Lit tous les entêtes de réponse HTTP sur Netscape serveur

array **nsapi_response_headers** (void)

[nsapi_response_headers](#) retourne un tableau contenant tous les entêtes de réponse HTTP sur un serveur Netscape. Cette fonctionnalité n'est disponible que depuis ^{PHP} 4.3.3 et plus récent.

Voir aussi [nsapi_request_headers](#) et [headers_sent](#) .

7.70.9 nsapi_virtual() : Effectue une sous-requête NSAPI

int **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 entêtes.

Vous ne pouvez pas faire de requêtes récursives avec cette fonction, vers un autre scripts ^{PHP} . Si vous voulez inclure des scripts ^{PHP} utilisez la fonction [include](#) ou [require](#) .

Note

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` . Il se peut que les nouveaux serveurs disposent d'une nouvelle version de cette DLL, qui n'est pas détectée. Contactez nous si vous avez des problèmes avec.

7.71 ODBC unifié

7.71.1 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 .

7.71.2 Pré-requis

Pour accéder à l'une des bases de données supportées, il faut que les librairies équivalentes soient installées.

7.71.3 Installation

```
--with-adas=[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-empress=[dir]
    Inclut le support Empress. DIR est le dossier d'installation de Empress. Par défaut,
    c'est $EMPRESSPATH . Depuis PHP4, cette option ne supporte que Empress Version 8.60 et
    plus récent.
--with-empress-bcs=[dir]
    Inclut le support Empress Local Access. DIR est le dossier d'installation de Empress.
    Par défaut, c'est $EMPRESSPATH . Depuis PHP4, 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 a user defined ODBC. 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 ).
```

To disable unified ODBC support in **PHP 3** add `--disable-unified-odbc` to your configure line. Only applicable if iODBC, Adabas, Solid, Velocis or a custom ODBC interface is enabled.

La version Windows de ^{PHP} dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

7.71.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<code>odbc.default_db *</code>	NULL	PHP_INI_ALL
<code>odbc.default_user *</code>	NULL	PHP_INI_ALL
<code>odbc.default_pw *</code>	NULL	PHP_INI_ALL
<code>odbc.allow_persistent</code>	"1"	PHP_INI_SYSTEM
<code>odbc.check_persistent</code>	"1"	PHP_INI_SYSTEM
<code>odbc.max_persistent</code>	"-1"	PHP_INI_SYSTEM
<code>odbc.max_links</code>	"-1"	PHP_INI_SYSTEM
<code>odbc.defaultlrl</code>	"4096"	PHP_INI_ALL
<code>odbc.defaultbinmode</code>	"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 [ini_set](#) .

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.

`odbc.defaultbinmode` entier

Gestion des données binaires.

7.71.5 Types de ressources

Cette extension ne définit aucune ressource.

7.71.6 Constantes prédefinies

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 prédéfinies](#)
- [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](#)

7.71.8 odbc_binmode() : Modifie la gestion des colonnes de données binaires

`int 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 la valeur par défaut de <code>odbc_binmode</code> est <code>ODBC_BINMODE_RETURN</code> . La gestion des colonnes binaires est aussi modifiée par odbc_longreadlen .

7.71.9 `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

`odbc_close_all` échouera s'il y a des transactions en cours sur cette connexion. Dans ce cas, la connexion restera ouverte.

7.71.10 `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

`odbc_close` échouera s'il y a des transactions en cours sur cette connexion. Dans ce cas, la connexion restera ouverte.

7.71.11 `odbc_columnprivileges()` : Liste les colonnes et leurs droits associés

```
int 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).

7.71.12 `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, ou bien `FALSE` en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- COLUMN_NAME
- DATA_TYPE
- TYPE_NAME
- PRECISION
- LENGTH
- SCALE
- RADIX
- NULLABLE
- REMARKS

Le résultat est ordonné par TABLE_QUALIFIER, TABLE_OWNER 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.

7.71.13 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.

7.71.14 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](#).

7.71.15 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`.

7.71.16 odbc_data_source() : Retourne des informations sur la connexion courante

```
resource odbc_data_source ( resource connection_id , constant fetch_type )
```

[odbc_data_source](#) retourne `FALSE` en cas d'erreur, et un tableau en cas de réussite.

[odbc_data_source](#) retourne des informations sur la connexion identifiée par la ressource `connection_id`, suivies d'informations sur le DSN. 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, et puis `SQL_FETCH_NEXT`.

7.71.17 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`.

7.71.18 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.

Voir aussi [odbc_errormsg](#) et [odbc_exec](#).

7.71.19 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.

Voir aussi [odbc_error](#) et [odbc_exec](#).

7.71.20 `odbc_exec()` : Prépare et exécute une requête SQL

```
resource odbc_exec ( resource connection_id , string query_string )
```

[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.

7.71.21 `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 envoyer 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 pas** 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 anti slash, 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](#).

7.71.22 `odbc_fetch_array()` : Lit une ligne de résultat dans un tableau associatif

```
array odbc_fetch_array ( resource result , int rownumber )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.71.23 `odbc_fetch_into()` : Lit une ligne de résultat, et la place dans un tableau.

`bool odbc_fetch_into ( resource result_id , int rownumber , array result_array ) resource odbc_fetch_into ( resource result_id , array result_array , int rownumber )`

`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
$src = odbc_fetch_into($res_id, $my_array);
?>
```

OU

```
<?php
$src = odbc_fetch_into($res_id, $row, $my_array);
$src = odbc_fetch_into($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
$src = odbc_fetch_into($res_id, $my_array);
?>
```

OU

```
<?php
$row = 1;
$src = odbc_fetch_into($res_id, $row, $my_array);
?>
```

Exemple avec `odbc_fetch_into` en PHP 4.2.0

```
<?php
$src = odbc_fetch_into($res_id, $my_array);
?>
```

OU

```
<?php
$src = odbc_fetch_into($res_id, $my_array, 2);
?>
```

7.71.24 `odbc_fetch_object()` : Lit une ligne de résultat dans un objet

object **odbc_fetch_object** (resource result , int rownumber)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.71.25 `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 continue 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.

7.71.26 `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 référence par le nombre `field_number`, dans la connexion ODBC `result_id`. Les numéros de champs commencent à 1.

7.71.27 `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.

7.71.28 `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.

7.71.29 `odbc_field_precision()` : Alias de [odbc_field_len](#)

```
string 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.

7.71.30 `odbc_field_scale()` : Lit l'échelle d'un champ

```
string odbc_field_scale ( resource result_id , int field_number )
```

[odbc_field_precision](#) retourne l'échelle du champ référencé par son numéro de champ `field_number` dans le résultat ODBC `result_id` .

7.71.31 `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.

7.71.32 `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.

7.71.33 `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.

7.71.34 `odbc_gettypeinfo()` : Liste les types de données supportés par une source

```
int 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`.

7.71.35 `odbc_longreadlen()` : Gestion des colonnes de type LONG

```
int 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 odbc_binmode .

7.71.36 `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.

7.71.37 `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](#).

7.71.38 `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.

7.71.39 `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 terminé. 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}.

7.71.40 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.

[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](#).

7.71.41 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

7.71.42 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).

7.71.43 `odbc_procedures()` : Liste les procédure 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, 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).

7.71.44 `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.

7.71.45 `odbc_result()` : Lit un champ de résultat UODBC

```
string odbc_result ( resource result_id , mixed field )
```

`odbc_result` retourne le contenu d'un champ.

`field` peut être aussi bien un entier, contenant le numéro de colonne du champ, dans le résultat, ou bien une chaîne de caractère, 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`.

7.71.46 odbc_rollback() : Annule une transaction

```
int 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.

7.71.47 odbc_setoption() : Modifie les paramètres ODBC.

```
int odbc_setoption ( resource id , int function , 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.

`function` 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 meme 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);
?>
```

7.71.48 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` est `SQL_ROWVER`, [odbc_specialcolumns](#) retourne l'ensemble optimal de colonnes tel qu'en lisant les valeurs de ces colonnes, on puisse spécifier n'importe quelle ligne de manière unique.

[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.

7.71.49 odbc_statistics() : Calcule 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.

7.71.50 `odbc_tableprivileges()` : Liste les tables et leurs privilèges

```
int 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).

7.71.51 `odbc_tables()` : Liste les tables d'une source.

```
int 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, 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` sont disponibles :

- Si `qualifier` est un signe de pourcentage (%), et `owner` et `name` sont des chaînes vides, alors le résultat contient la liste des qualifieurs 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.

7.72 Aggregation 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.

7.72.1 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' **Aggregation** .

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 = $_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 propriétés

    function Report() {
        // initialisation
    }

    function setDateTime(&$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
$rep->setDateTime(&$dt);
echo $rep->generateReport();

// plus loin dans le code

// generation du rapport avec une date désignée
$rep->setDateTime(&$dtp);
$output = $rep->generateReport();
// sauvegarde pour affichage dans la base
// ... etc ...
?>

```

L' **Aggregation** , d'un autre coté, 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'aggregation d'instances (aggrégation 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'aggrégation d'object est une méthode naturelle pour représenter des relation de type tout-partie (par exemple, une molécule est une aggrégation 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 aggrégations d'objets sont plus souples, car nous pouvons alors sélectionner les méthodes et les propriétés qui sont transmises à l'objet aggrégé.

7.72.2 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 issues 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, désaggrégeons WDDXStorage\n";
deaggregate($fs, "WDDXStorage");
echo "L'objet \$fs\n";
echo object_info($fs)."\n";

?>
```

Etudions 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 si 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 `$version`, et la méthode `store`, qui sont définies dans `WDDXStorage`. Une chose importante à noter est que les éléments privés n'ont pas été agrégés, même si ils sont présents dans l'objet `$ws`. Un autre absent est le constructeur de `WDDXStorage`, qu'il n'est pas logique d'aggréger.

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 agrégés les méthodes et propriétés dynamiquement, nous pouvons aussi les désaggréguer. Si vous désaggrégeons la classe `WDDXStorage` de l'objet `$fs`, nous allons obtenir :

```

Désagrégation d'objets

Et finalement, désaggrégeons 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'aggrégation 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'aggrégation.

- [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](#)

7.72.4 aggregate_methods_by_list() : Aggrège 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` aggrège 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_regex](#), [aggregate_properties](#), [aggregate_properties_by_list](#), [aggregate_properties_by_regex](#) et [deaggregate](#)

7.72.5 `aggregate_methods_by_regex()` : Aggrège sélectivement les méthodes d'une classe grâce à une expression régulière

```
void aggregate_methods_by_regex ( object object , string class_name , string regex , bool exclude )
```

[aggregate_methods_by_regex](#) aggrège les méthodes de la classe `class_name` avec l'objet `object`, en filtrant les méthodes agrégées grâce à l'expression régulière `regex`, 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_list](#), [aggregate_properties](#), [aggregate_properties_by_list](#), [aggregate_properties_by_regex](#) et [deaggregate](#)

7.72.6 `aggregate_methods()` : Aggrège dynamiquement les méthodes d'une classe à un objet

```
void aggregate_methods ( object object , string class_name )
```

[aggregate_methods](#) aggrège 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_regex](#), [aggregate_properties](#), [aggregate_properties_by_list](#), [aggregate_properties_by_regex](#) et [deaggregate](#)

7.72.7 `aggregate_properties_by_list()` : Aggrège 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](#) aggrège 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_regex](#), [aggregate_properties](#), [aggregate_properties_by_regex](#), [aggregate_info](#), [deaggregate](#)

7.72.8 `aggregate_properties_by_regexp()` : Aggrège sélectivement les propriétés d'une classe grâce à une expression régulière

```
void aggregate_properties_by_regexp ( object object , string class_name , string regexp , bool exclude )
```

[aggregate_properties_by_regexp](#) aggrège les propriétés de la classe `class_name` avec l'objet `object`, en filtrant les propriétés agrégées grâce à l'expression régulière `regexp`, appliquée au nom de la propriétés. 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 list](#), [aggregate info](#) et [deaggregate](#)

7.72.9 `aggregate_properties()` : Aggrège dynamiquement les propriétés d'une classe à un objet

```
void aggregate_properties ( object object , string class_name )
```

[aggregate_properties](#) aggrège 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](#)

7.72.10 `aggregate()` : Aggrège dynamiquement des classes et objets

```
void aggregate ( object object , string class_name )
```

[aggregate](#) aggrège 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'aggrégation. Les constructeurs sont aussi exclus de l'aggrégation.

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](#)

7.72.11 `aggregation_info()` : Retourne un tableau associatif décrivant les classes agrégées

```
array aggregation_info ( object object )
```

[aggregation_info](#) retourne les informations d'aggrégation pour l'objet `object`, sous la forme d'un tableau associatif.

7.72.12 deaggregate() : Desaggrège 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](#)

7.73 Oracle 8

7.73.1 Introduction

Ces fonctions vous permettront d'accéder aux serveurs Oracle8 et Oracle7. Elles utilisent l'interface Oracle8 Call-Interface (oci8). Vous aurez donc besoin des bibliothèques clientes Oracle8 pour pouvoir les utiliser.

Il faut noter que cette extension est plus souple que l'extension Oracle officielle. Elle supporte notamment les liaisons entre les variables globales et locales de PHP avec des emplacements Oracle; elle supporte complètement les types LOB, FILE et ROWID et vous permet d'utiliser des variables de définitions personnalisables.

7.73.2 Pré-requis

Avant d'utiliser cette extension, assurez-vous que vous avez bien paramétré vos variables d'environnement Oracle, ainsi que votre démon utilisateur. Les variables dont vous pouvez avoir besoin sont :

- ORACLE_HOME
- ORACLE_SID
- LD_PRELOAD
- LD_LIBRARY_PATH
- NLS_LANG
- ORA_NLS33

Après avoir configuré ces variables pour votre utilisateur "serveur web", assurez-vous aussi d'ajouter cet utilisateur (nobody, www) au group Oracle.

Note
Si votre serveur web ne démarre pas, ou crashe au démarrage
Vérifiez que Apache a bien été compilé avec la librairie pthread :
Configuration Apache avec pthread
<pre># 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)</pre>
Si la libpthread n'est pas listée, vous devez réinstaller Apache :
Reinstaller Apache avec pthread
<pre># cd /usr/src/apache_1.3.xx # make clean # LIBS=-lpthread ./config.status # make # make install</pre>

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.

7.73.3 Installation

Vous devez compiler PHP avec l'option `--with-oci8[=DIR]` , où DIR vaut par défaut la variable d'environnement ORACLE_HOME.

7.73.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.73.5 Types de ressources

7.73.6 Constantes prédefinies

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)
OCI_DESCRIBE_ONLY (entier)
OCI_COMMIT_ON_SUCCESS (entier)
OCI_EXACT_FETCH (entier)
SQLT_BFILEE (entier)
SQLT_CFILEE (entier)
SQLT_CLOB (entier)
SQLT_BLOB (entier)
SQLT_RDD (entier)
OCI_B_SQLT_NTY (entier)
OCI_SYSDATE (entier)
OCI_B_BFILE (entier)
OCI_B_CFILEE (entier)
OCI_B_CLOB (entier)
OCI_B_BLOB (entier)
OCI_B_ROWID (entier)
OCI_B_CURSOR (entier)
OCI_B_BIN (entier)
OCI_FETCHSTATEMENT_BY_COLUMN (entier)
OCI_FETCHSTATEMENT_BY_ROW (entier)
OCI_ASSOC (entier)
OCI_NUM (entier)
OCI_BOTH (entier)
OCI_RETURN_NULLS (entier)
OCI_RETURN_LOBS (entier)
OCI_DTYPE_FILE (entier)
OCI_DTYPE_LOB (entier)
OCI_DTYPE_ROWID (entier)
OCI_D_FILE (entier)
OCI_D_LOB (entier)
OCI_D_ROWID (entier)
```

7.73.7 Exemples

Exemple avec OCI8

```
<?php
// par sergo@bacup.ru

// Utilisez l'option: OCI_DEFAULT pour exécuter la commande avec exécution
// retardée.
OCIExecute($stmt, OCI_DEFAULT);

// pour lire les données, après un fetch :

$result = OCIResult($stmt, $n);
if (is_object ($result)) $result = $result->load();

// Pour les commandes INSERT et UPDATE utilisez :

$sql = "insert into table (field1, field2) values (field1 = 'value',
field2 = empty_clob()) returning field2 into :field2";
OCIParse($conn, $sql);
$clob = OCINewDescriptor($conn, OCI_D_LOB);
OCIBindByName ($stmt, ":field2", &$clob, -1, OCI_B_CLOB);
OCIExecute($stmt, OCI_DEFAULT);
$clob->save ("some text");
OCICommit($conn);

?>
```

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
$stmt = OCIParse ( $dbh, "begin sp_newaddress( :address_id, '$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 :

OCIBindByName ( $sth, ":address_id", $addr_id, 10 );
OCIBindByName ( $sth, ":error_code", $errorcode, 10 );
OCIExecute ( $sth );

?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [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](#)
- [ocinewdescriptor](#)
- [ocinlogon](#)
- [ocinumcols](#)
- [ociparse](#)
- [ociplogon](#)
- [ocireult](#)
- [ocirollback](#)
- [ocirowcount](#)
- [ocisavelob](#)
- [ocisavelobfile](#)
- [ociserverversion](#)
- [ocisetprefetch](#)
- [ocistatementtype](#)
- [ociwritelobtofile](#)
- [ociwritetemporarylob](#)

7.73.9 **ocicancel()** : Termine la lecture de ressources Oracle

`bool ocicancel ( resource stmt )`

[ocicancel](#) détruit les ressources liées au dernier résultat `stmt` . Si vous ne souhaitez plus lire d'informations dans ce résultat, utilisez cette fonction.

7.73.10 **ocicloselob()** : Ferme un lob Oracle

`bool ocicloselob ( )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.11 ocicollappend() : Ajoute un objet à une collection Oracle

```
bool ocicollappend ( string value )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.12 ocicollassign() : Assigne une collection depuis une autre collection Oracle

```
bool ocicollassign ( object from )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.13 ocicollassignelem() : Assigne l'élément val à une collection Oracle à l'index ndx

```
bool ocicollassignelem ( int ndx , string val )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.14 ocicollgetelem() : Lit la valeur d'un élément d'une collection Oracle à l'index ndx

```
string ocicollgetelem ( int ndx )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.15 ocicollmax() : Retourne la plus grande valeur d'une collection Oracle

```
int ocicollmax ( )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Pour un varray, c'est la taille maximale du tableau.

7.73.16 ocicollsize() : Retourne la taille d'une collection Oracle

```
int ocicollsize ( )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.17 ocicolltrim() : Supprime num éléments à la fin de la collection Oracle

bool **ocicolltrim** (int num)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.18 ocicolumnisnull() : Teste si la valeur d'une colonne Oracle est NULL

bool **ocicolumnisnull** (resource stmt , mixed col)

ocicolumnisnull retourne `TRUE` si la colonne `column` 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 `column`.

7.73.19 ocicolumnname() : Retourne le nom d'une colonne Oracle

string **ocicolumnname** (resource stmt , int col)

ocicolumnname retourne le nom de la colonne numéro `col` (l'indexation des colonnes commence à 1).

Exemple avec ocicolumnname

```
<?php
print "<HTML><PRE>\n";
$conn = OCILogon("scott", "tiger");
$stmt = OCIParse($conn,"select * from emp");
OCIExecute($stmt);
print "<TABLE BORDER=\\"1\">";
print "<TR>";
print "<TH>Nom</TH>";
print "<TH>Type</TH>";
print "<TH>Taille</TH>";
print "</TR>";
$ncols = OCINumCols($stmt);
for ( $i = 1; $i <= $ncols; $i++ ) {
    $column_name = OCIColumnName($stmt,$i);
    $column_type = OCIColumnType($stmt,$i);
    $column_size = OCIColumnSize($stmt,$i);
    print "<TR>";
    print "<TD>$column_name</TD>";
    print "<TD>$column_type</TD>";
    print "<TD>$column_size</TD>";
    print "</TR>";
}
print "</TABLE>\n";
OCIFreeStatement($stmt);
OCILogoff($conn);
print "</PRE>";
print "</HTML>\n";
?>
```

Voir aussi ocinumcols, ocicolumntype et ocicolumnsize.

7.73.20 ocicolumnprecision() : Lit la précision d'une colonne Oracle

int **ocicolumnprecision** (resource stmt , int col)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.21 ocicolumnscale() : Lit la taille d'une colonne Oracle

```
int ocicolumnscale ( resource stmt , int col )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.22 ocicolumnsize() : Retourne la taille d'une colonne Oracle

```
int ocicolumnsize ( resource stmt , mixed column )
```

ocicolumnsize retourne la taille de la colonne. Vous pouvez utiliser l'index de colonne (l'indexation commence à 1) ou le nom de la colonne dans le paramètre `column`.

Exemple avec <u>ocicolumnsize</u>

```
<?php
print "<HTML><PRE>\n";
$conn = OCILogon("scott", "tiger");
$stmt = OCIParse($conn,"select * from emp");
OCIDeclare($stmt);
print "<TABLE BORDER=\\"1\">";
print "<TR>";
print "<TH>Name</TH>";
print "<TH>Type</TH>";
print "<TH>Length</TH>";
print "</TR>";
$numcols = OCINumCols($stmt);
for ( $i = 1; $i <= $numcols; $i++ ) {
    $column_name = OCIColumnName($stmt,$i);
    $column_type = OCIColumnType($stmt,$i);
    $column_size = OCIColumnSize($stmt,$i);
    print "<TR>";
    print "<TD>$column_name</TD>";
    print "<TD>$column_type</TD>";
    print "<TD>$column_size</TD>";
    print "</TR>";
}
print "</TABLE>";
OCIFreeStatement($stmt);
OCILogoff($conn);
print "</PRE>";
print "</HTML>\n";
?>
```

Voir aussi ocinumcols , ocicolumnname et ocicolumnsize .

7.73.23 ocicolumntype() : Retourne le type de données d'une colonne Oracle

```
mixed ocicolumntype ( resource stmt , int col )
```

ocicolumntype retourne le type de données de la colonne correspondant au numéro de colonne `col` dans le résultat `stmt` (les colonnes sont indexées à partir de 1).

Exemple avec <u>ocicolumntype</u>

```
<?php
print "<HTML><PRE>\n";
$conn = OCILogon("scott", "tiger");
$stmt = OCIParse($conn,"select * from emp");
OCIDeclare($stmt);
print "<TABLE BORDER=\\"1\">";
```

```

print "<TR>";
print "<TH>Nom</TH>";
print "<TH>Type</TH>";
print "<TH>Longueur</TH>";
print "</TR>";
$numcols = OCINumCols($stmt);
for ( $i = 1; $i <= $numcols; $i++ ) {
    $column_name = OCIColumnName($stmt,$i);
    $column_type = OCIColumnType($stmt,$i);
    $column_size = OCIColumnSize($stmt,$i);
    print "<TR>";
    print "<TD>$column_name</TD>";
    print "<TD>$column_type</TD>";
    print "<TD>$column_size</TD>";
    print "</TR>";
}
print "</TABLE>\n";
OCIFreeStatement($stmt);
OCILogoff($conn);
print "</PRE>";
print "</HTML>\n";
?>

```

Voir aussi [ocinumcols](#) , [ocicolumnname](#) et [ocicolumnsize](#) .

7.73.24 ocicolumntyperaw() : Lit directement le type de colonne Oracle

mixed **ocicolumntyperaw** (resource stmt , int col)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.25 ocicommit() : Valide les transactions Oracle en cours

bool **ocicommit** (resource connection)

[ocicommit](#) valide toutes les transactions en cours sur la connexion Oracle `connection` . Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

This example demonstrates how OCICommit is used.

OCICommit

```

<?php
// Connexion au serveur Oracle
$conn = OCILogon('scott', 'tiger');

// Analyse de la requête SQL
$stmt = OCIParse($conn, "INSERT INTO employees (name, surname) VALUES ('Maxim', 'Maletsky')");

// Exécution de la requête
OCIExecute($stmt);

// Validation de la transaction
$committed = OCICommit($conn);

// Vérification de la validation : si une erreur est survenue, afficher
// le message d'erreur
if (!$committed) {
    $error = OCIError($conn);
    echo "La validation a échoué. Oracle dit : " . $error['message'];
}

// Fermeture de la connexion
OCILogoff($conn);
?>

```

Voir aussi [ocirollback](#) .

7.73.26 ocidefinebyname() : Utilise une variable PHP pour la phase de définition, dans une commande SELECT Oracle

bool **ocidefinebyname** (resource stmt , string column_name , mixed &variable , int type)

ocidefinebyname copie les valeurs issues de colonnes SQL `column-name` dans les variables PHP.

Méfiez-vous des colonnes Oracle qui sont toutes en majuscule, tandis que dans les SELECT, vous pouvez aussi les écrire en minuscules. ocidefinebyname 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.

Si vous avez besoin de définir un type de données abstrait, tel que (LOB/ROWID/BFILE), vous devez lui allouer la mémoire avec ocinewdescriptor. Reportez-vous aussi à ocibindbyname.

Exemple avec ocidefinebyname

```
<?php
$conn = OCILogin("scott","tiger");

$stmt = OCIParse($conn,"select empno, ename from emp");

/* La définition DOIT intervenir AVANT l'exécution! */

OCIDefineByName($stmt,"EMPNO",$empno);
OCIDefineByName($stmt,"ENAME",$ename);

OCIExecute($stmt);

while (OCIFetch($stmt)) {
    echo "empno :".$empno."\n";
    echo "ename :".$ename."\n";
}

OCIFreeStatement($stmt);
OCILogoff($conn);
?>
```

7.73.27 ocierror() : Retourne la dernière erreur de stmt|conn|global sur Oracle

array **ocierror** (resource stmt|conn|global)

ocierror retourne la dernière erreur trouvée. Si l'option `stmt|conn` n'est pas fournie, la dernière erreur rencontrée est retournée. Si aucune erreur n'est trouvée, ocierror retourne `FALSE`. ocierror 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.

7.73.28 ociexecute() : Exécute une commande SQL Oracle

bool **ociexecute** (resource stmt , int mode)

ociexecute exécute une commande déjà préparée (voir ociparse). 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`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.73.29 ocifetch() : Lit la prochaine ligne dans le résultat Oracle

bool **ocifetch** (resource stmt)

ocifetch lit la prochaine ligne (d'une commande SELECT) et la place dans le buffer interne de résultat. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.73.30 ocifetchinto() : Retourne la ligne suivante d'un résultat Oracle dans un tableau

int **ocifetchinto** (resource stmt , array &result , int mode)

ocifetchinto retourne la ligne suivante (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_login("username","password");

$query = "SELECT pommes FROM arbre";

$stmt = OCIParse ($conn, $query);
OCIExecute ($stmt);

while (OCIFetchInto ($stmt, $row, OCI_ASSOC)) {
    print $row['pommes'];
}
?>
```

Voir aussi ocifetch et ociexecute .

7.73.31 ocifetchstatement() : Retourne toutes les lignes d'un résultat Oracle

int **ocifetchstatement** (resource stmt , array &output , int skip , int maxrows , int flags)

ocifetchstatement retourne toutes les lignes d'un résultat dans le tableau variable. ocifetchstatement retourne le nombre de lignes retournées. ^{skip} est le nombre de lignes initiales à ignorer (par défaut, 0). ^{maxrows} est le nombre de lignes à lire, en commençant à la ^{skip} -ième ligne (par défaut, -1, qui représente toute les lignes).

^{flags} représente les options disponibles pour cette fonction, et qui peuvent être la combinaison des constantes suivantes :

- OCI_FETCHSTATEMENT_BY_ROW

- OCI_FETCHSTATEMENT_BY_COLUMN (valeur par défaut)
- OCI_NUM
- OCI_ASSOC

Exemple avec ocifetchstatement

```
<?php
$conn = OCILogon("scott","tiger");

$stmt = OCIParse($conn,"select * from emp");

OCIExecute($stmt);

$rows = OCIFetchStatement($stmt,$results);
if ( $rows > 0 ) {
    print "<TABLE BORDER='1'\>\n";
    print "<TR>\n";
    while ( list( $key, $val ) = each( $results ) ) {
        print "<TH>$key</TH>\n";
    }
    print "</TR>\n";

    for ( $i = 0; $i < $rows; $i++ ) {
        reset($results);
        print "<TR>\n";
        while ( $column = each($results) ) {
            $data = $column['value'];
            print "<TD>$data[$i]</TD>\n";
        }
        print "</TR>\n";
    }
    print "</TABLE>\n";
} else {
    echo "Aucune données trouvées<BR>\n";
}
print "$rows lignes lues<BR>\n";

OCIFreeStatement($stmt);
OCILogoff($conn);
?>
```

7.73.32 ocifreecollection() : Supprime une collection Oracle

bool **ocifreecollection** ()

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.33 ocifreecursor() : Libère toutes les ressources réservées par un résultat Oracle

bool **ocifreecursor** (resource stmt)

ocifreecursor libère toutes les ressources réservées par le résultat Oracle ^{stmt}. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.73.34 ocifreedesc() : Détruit un pointeur de LOB Oracle

bool **ocifreedesc** ()

ocifreedesc détruit le pointeur de LOB ^{lob}. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.73.35 ocifreestatement() : Libère toutes les ressources occupées par une commande Oracle

bool **ocifreestatement** (resource stmt)

ocifreestatement libère toutes les ressources occupées par la commande `stmt`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.73.36 ociinternaldebug() : Active ou désactive l'affichage des données de débogage Oracle

void **ociinternaldebug** (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.

7.73.37 ociloadlob() : Charge un LOB Oracle

string **ociloadlob** ()

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.38 ocilogoff() : Déconnecte d'un serveur Oracle

bool **ocilogoff** (resource connection)

ocilogoff ferme la connexion Oracle `connection`.

Utiliser ocilogoff n'est pas nécessaire, car les connexions non persistantes seront automatiquement fermées à la fin du script. Voyez aussi le chapitre sur la libération des ressources.

7.73.39 ocilogon() : Etablit une connexion à un serveur Oracle

resource **ocilogon** (string username , string password , string db)

ocilogon retourne un identifiant de connexion, nécessaire à la plupart des fonctions oci. Le paramètre optionnel `db` peut contenir le nom local du serveur Oracle, ou bien son nom dans la table d'entrées `tnsnames.ora` à laquelle vous voulez vous connecter. Si le troisième paramètre n'est pas spécifié, PHP utilisera la variable d'environnement `ORACLE_SID` (instance Oracle) ou `TWO_TASK` (`tnsnames.ora`) pour déterminer la base à laquelle se connecter.

Les connexions sont partagées, à l'intérieur d'une même page avec ocilogon. Cela signifie que COMMIT et ROLLBACK s'appliquent à toutes les transactions commencées à l'intérieur d'une même page, même si vous avez créé de multiples connexions.

Les connexions sont partagées au niveau de la page, lorsque vous utilisez ocilogon. Cela signifie que les validations et annulations s'appliquent à toutes les transactions de la page, même si vous avez créé de multiples connexions à un même serveur Oracle.

Cet exemple montre comment les connexions sont partagées :

Exemple avec `oci_logon`

```
<?php
print "<HTML><PRE>";
$db = "";

$c1 = oci_logon("scott","tiger",$db);
$c2 = oci_logon("scott","tiger",$db);

function create_table($conn)
{ $stmt = ociparse($conn,"create table scott.hallo (test varchar2(64))");
  ociexecute($stmt);
  echo $conn." a créé une table\n\n";
}

function drop_table($conn)
{ $stmt = ociparse($conn,"drop table scott.hallo");
  ociexecute($stmt);
  echo $conn." a effacé une table\n\n";
}

function insert_data($conn)
{ $stmt = ociparse($conn,"insert into scott.hallo
    values(' $conn' || ' ' || to_char(sysdate,'DD-MON-YY HH24:MI:SS'))");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn." a inséré dans hallo\n\n";
}

function delete_data($conn)
{ $stmt = ociparse($conn,"delete from scott.hallo");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn." a effacé dans hallo\n\n";
}

function commit($conn)
{ ocicommit($conn);
  echo $conn." validé\n\n";
}

function rollback($conn)
{ ocirollback($conn);
  echo $conn." annulé\n\n";
}

function select_data($conn)
{ $stmt = ociparse($conn,"select * from scott.hallo");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn."----selection\n\n";
  while (ocifetch($stmt))
    echo $conn." <.ociresult($stmt,"TEST").">\n\n";
  echo $conn."----fin\n\n";
}

create_table($c1);
insert_data($c1); // Insertion avec c1
insert_data($c2); // Insertion avec c2

select_data($c1); // Resultats des insertions
select_data($c2);

rollback($c1); // Annulation sur c1

select_data($c1); // Les deux insertions ont été annulées
select_data($c2);

insert_data($c2); // Insertion avec c2
commit($c2); // Validation avec c2

select_data($c1); // Le résultat de l'insertion via c2

delete_data($c1); // Effacement de toutes les lignes avec c1
select_data($c1); // Aucune ligne lue
select_data($c2); // Aucune ligne lue
commit($c1); // Validation avec c1

select_data($c1); // Aucune ligne lue
select_data($c2); // Aucune ligne lue

drop_table($c1);
print "</PRE></HTML>";
?>
```

Voir aussi [oci_logon](#) et [oci_logon](#) .

7.73.40 ocinewcollection() : Initialise une nouvelle collection

object **ocinewcollection** (resource connection , string tdo , string schema)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.41 ocinewcursor() : Retourne un nouveau pointeur à utiliser pour lier les pointeurs de références

resource **ocinewcursor** (resource conn)

ocinewcursor alloue un nouveau pointeur de commande, pour la connexion `conn`.

Utiliser un REF CURSOR issue d'une procédure enregistrée

```
<?php
// supposons que votre procédure stockée info.output retourne un curseur de référence dans :data

$conn = OCILogon("scott","tiger");
$curs = OCINewCursor($conn);
$stmt = OCIParse($conn,"begin info.output(:data); end;");

ocibindbyname($stmt,"data",&$curs,-1,OCI_B_CURSOR);
ociexecute($stmt);
ociexecute($curs);

while (OCIFetchInto($curs,&$data)) {
    var_dump($data);
}

OCIFreeStatement($stmt);
OCIFreeCursor($curs);
OCILogoff($conn);
?>
```

Utiliser un REF CURSOR issue d'une commande SELECT

```
<?php
print "<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);
print "<TABLE BORDER='1'>";
print "<TR>";
print "<TH>DEPT NAME</TH>";
print "<TH>DEPT #</TH>";
print "<TH># EMPLOYEES</TH>";
print "</TR>";

while (OCIFetchInto($stmt,&$data,OCI_ASSOC)) {
    print "<TR>";
    $dname = $data["DNAME"];
    $deptno = $data["DEPTNO"];
    print "<TD>$dname</TD>";
    print "<TD>$deptno</TD>";
    ociexecute($data["EMPCNT"]);
    while (OCIFetchInto($data["EMPCNT"],&$subdata,OCI_ASSOC)) {
        $num_emps = $subdata["NUM_EMPS"];
        print "<TD>$num_emps</TD>";
    }
    print "</TR>";
}
print "</TABLE>";
print "</BODY></HTML>";
OCIFreeStatement($stmt);
OCILogoff($conn);
?>
```

7.73.42 ocinewdescriptor() : Initialise un nouveau pointeur vide de LOB/FILE

object **ocinewdescriptor** (resource connection , int type)

ocinewdescriptor alloue l'espace nécessaire pour stocker un descripteur, ou un pointeur de LOB. Les valeurs acceptées pour type sont OCI_D_FILE, OCI_D_LOB et OCI_D_ROWID. Pour les pointeurs de LOB, les méthodes load, save, et savefile sont associées avec le pointeur. Pour les pointeurs de type BFILE, seule la méthode load existe. Voyez le second exemple pour une illustration.

ocinewdescriptor

```
<?php
/* Ce script est fait pour être appelé dans un formulaire HTML
 * Il attend 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).
 */
$conn = OCILogon($user, $password);
$stmt = OCIParse($conn,"select rowid from $table $where");
$rowid = OCINewDescriptor($conn,OCI_D_ROWID);
OCIDefineByName($stmt,"ROWID",&$rowid);
OCIExecute($stmt);
while ( OCIFetch($stmt) ) {
    $nrows = OCIRowCount($stmt);
    $delete = OCIParse($conn,"delete from $table where ROWID = :rid");
    OCIBindByName($delete,":rid",&$rowid,-1,OCI_B_ROWID);
    OCIExecute($delete);
    print "$nrows\n";
    if ( ($nrows % $commitsize) == 0 ) {
        OCICommit($conn);
    }
}
$nrows = OCIRowCount($stmt);
print "$nrows deleted...\n";
OCIFreeStatement($stmt);
OCILogoff($conn);
?>

<?php
/* Ce script est fait pour être appelé depuis un formulaire HTML.
 * Il attend les variables $user, $password, $table, $where, et $commitsize,
 * données par le formulaire. Le script efface
 * les lignes sélectionnées avec ROWID est valide les transactions
 * à chaque jeu de $commitsize lignes. (Attention : il n'y plus d'annulation) */
if(!isset($lob_upload) || $lob_upload == 'none'){
?>
<form action="upload.php" method="post" enctype="multipart/form-data">
Fichier à télécharger : <input type="file" name="lob_upload"><br>
<input type="submit" value="Envoyer"> - <input type="reset">
</form>
<?php
} else {

    // $lob_upload contient le nom temporaire du fichier téléchargé

    // voir aussi la section fonctionnalités sur les téléchargements de
    // fichiers si vous voulez utiliser des téléchargements sécurisés

    $conn = OCILogon($user, $password);
    $lob = OCINewDescriptor($conn, OCI_D_LOB);
    $stmt = OCIParse($conn,"insert into $table (id, the_blob)
        values(my_seq.NEXTVAL, EMPTY_BLOB()) returning the_blob into :the_blob");
    OCIBindByName($stmt, ':the_blob', &$lob, -1, OCI_B_BLOB);
    OCIExecute($stmt, OCI_DEFAULT);
    if($lob->savefile($lob_upload)){
        OCICommit($conn);
        echo "Blob bien téléchargé\n";
    }else{
        echo "Impossible de télécharger le Blob\n";
    }
    OCIFreeDesc($lob);
    OCIFreeStatement($stmt);
    OCILogoff($conn);
}
?>
```

Exemple avec ocinewdescriptor

```

<?php
/* Appel d'une procédure PL/SQL stockée qui prend un clob
 * en entrée (PHP 4 >= 4.0.6).
 * Exemple de signature de procédure stockée PL/SQL :
 *
 * PROCEDURE save_data
 *   Nom de l'argument          Type          In/Out Default?
 *   -----
 *   KEY                         NUMBER(38)      IN
 *   DATA                       CLOB             IN
 *
 */

$conn = OCILogon($user, $password);
$stmt = OCIParse($conn, "begin save_data(:key, :data); end;");
$clob = OCINewDescriptor($conn, OCI_D_LOB);
OCIBindByName($stmt, ':key', $key);
OCIBindByName($stmt, ':data', $clob, -1, OCI_B_CLOB);
$clob->WriteTemporary($data);
OCIExecute($stmt, OCI_DEFAULT);
OCICommit($conn);
$clob->close();
$clob->free();
OCIFreeStatement($stmt);
?>

```

7.73.43 ocinlogon() : Etablit une nouvelle connexion avec un serveur Oracle

resource **ocinlogon** (string username , string password , string db)

ocinlogon crée une nouvelle connexion à un serveur Oracle et s'identifie. Le paramètre optionnel ^{db} peut contenir le nom local du serveur Oracle, ou bien son nom dans la table d'entrées tnsnames.ora à laquelle vous voulez vous connecter. Si le troisième paramètre n'est pas spécifié, ^{PHP} utilisera la variable d'environnement ^{ORACLE_SID} (instance Oracle) ou ^{TWO_TASK} (tnsnames.ora) pour déterminer la base à laquelle se connecter.

ocinlogon force le serveur à établir une nouvelle connexion. Cette fonction ne doit être utilisée que si vous voulez isoler un ensemble de transactions. Par défaut, les connexions sont partagées au niveau de la page, si vous utilisez la fonction ocinlogon ou bien au niveau du processus web, si vous utilisez oci_logon. Si vous avez de multiples connexions ouvertes avec ocinlogon, les validations et annulations ne s'appliquent qu'à la connexion spécifiée.

L'exemple ci-dessous montre l'utilisation des connexions séparées.

Exemple avec ocinlogon

```

<?php
print "<HTML><PRE>";
$db = "";

$c1 = ocinlogon("scott","tiger",$db);
$c2 = ocinlogon("scott","tiger",$db);

function create_table($conn)
{ $stmt = ociparse($conn,"create table scott.hallo (test varchar2(64))");
  ociexecute($stmt);
  echo $conn." a créé une table\n\n";
}

function drop_table($conn)
{ $stmt = ociparse($conn,"drop table scott.hallo");
  ociexecute($stmt);
  echo $conn." a effacé une table\n\n";
}

function insert_data($conn)
{ $stmt = ociparse($conn,"insert into scott.hallo
  values('$conn' || ' ' || to_char(sysdate,'DD-MON-YY HH24:MI:SS'))");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn." a inséré dans hallo\n\n";
}

```

```

function delete_data($conn)
{ $stmt = ociparse($conn,"delete from scott.hallo");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn." a effacé dans hallo\n\n";
}

function commit($conn)
{ ocicommit($conn);
  echo $conn." validé\n\n";
}

function rollback($conn)
{ ocirollback($conn);
  echo $conn." annulé\n\n";
}

function select_data($conn)
{ $stmt = ociparse($conn,"select * from scott.hallo");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn."----selection\n\n";
  while (ocifetch($stmt))
    echo $conn." <".ociresult($stmt,"TEST").">\n\n";
  echo $conn."----fin\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);
print "</PRE></HTML>";
?>

```

Voir aussi [ocilogon](#) et [ociplogon](#) .

7.73.44 ocinumcols() : Retourne le nombre de colonnes dans un résultat

int **ocinumcols** (resource stmt)

[ocinumcols](#) retourne le nombre de colonnes dans le résultat ^{stmt} .

Exemple avec [ocinumcols](#)

```

<?php
print "<HTML><PRE>\n";
$conn = OCILogon("scott", "tiger");
$stmt = OCIParse($conn,"select * from emp");
OCIExecute($stmt);
while ( OCIFetch($stmt) ) {
  print "\n";
  $ncols = OCINumCols($stmt);
  for ( $i = 1; $i <= $ncols; $i++ ) {
    $column_name = OCIColumnName($stmt,$i);
    $column_value = OCIResult($stmt,$i);
    print $column_name . ': ' . $column_value . "\n";
  }
  print "\n";
}
OCIFreeStatement($stmt);
OCILogoff($conn);
print "</PRE>";

```



```
?> print "</HTML>\n";
```

7.73.45 `ociparse()` : Analyse une requête SQL avec Oracle

`resource ociparse ( resource conn , string query )`

`ociparse` analyse la requête `query` sur la connexion `conn`, 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.

7.73.46 `ociologon()` : Ouvre une connexion persistante à un serveur Oracle

`resource ociologon ( string username , string password , string db )`

`ociologon` crée une connexion persistante à un serveur Oracle 8 et s'authentifie. Le paramètre optionnel `db` peut contenir le nom local du serveur Oracle, ou bien son nom dans la table d'entrées `tnsnames.ora` à laquelle vous voulez vous connecter. Si le troisième paramètre n'est pas spécifié, PHP utilisera la variable d'environnement `ORACLE_SID` (instance Oracle) ou `TWO_TASK` (`tnsnames.ora`) pour déterminer la base à laquelle se connecter.

Voir aussi [ociologon](#) et [ocinlogon](#).

7.73.47 `ociresult()` : Retourne la valeur d'une colonne dans un résultat Oracle

`mixed ociresult ( resource statement , mixed col )`

`ociresult` retourne les données de la colonne `column` dans la ligne courante (voir [ocifetch](#)). `ociresult` retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILES).

7.73.48 `ocirollback()` : Annule les transactions Oracle en cours

`bool ocirollback ( resource connection )`

`ocirollback` annule les transactions en cours sur la connexion Oracle `connection`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [ocicommit](#).

7.73.49 `ocirowcount()` : Retourne le nombre de lignes affectées par Oracle

`int ocirowcount ( resource stmt )`

`ocirowcount` retourne le nombre de lignes affectées par la commande de modification (UPDATE) `statement`. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que

les lignes aient été modifiées.

Exemple avec `ocirowcount`

```
<?php
print "<HTML><PRE>";
$conn = OCILogon("scott","tiger");
$stmt = OCIParse($conn,"create table emp2 as select * from emp");
OCIExecute($stmt);
print OCIRowCount($stmt) . " rows inserted.<BR>";
OCIFreeStatement($stmt);
$stmt = OCIParse($conn,"delete from emp2");
OCIExecute($stmt);
print OCIRowCount($stmt) . " rows deleted.<BR>";
OCICommit($conn);
OCIFreeStatement($stmt);
$stmt = OCIParse($conn,"drop table emp2");
OCIExecute($stmt);
OCIFreeStatement($stmt);
OCILogOff($conn);
print "</PRE></HTML>";
?>
```

7.73.50 `ocisavelob()` : Sauve un LOB Oracle

`bool ocisavelob ( )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.51 `ocisavelobfile()` : Sauve un fichier LOB Oracle

`bool ocisavelobfile ( )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.52 `ociserverversion()` : Retourne une chaîne contenant les informations de version du serveur Oracle

`string ociserverversion ( resource conn )`

`ociserverversion` retourne une chaîne contenant les informations de version du serveur Oracle, identifié par la connexion `conn`.

Exemple avec `ociserverversion`

```
<?php
$conn = OCILogon("scott","tiger");
print "Version du serveur Oracle : " . OCIServerVersion($conn);
OCILogOff($conn);
?>
```

7.73.53 `ocisetprefetch()` : Indique le nombre de lignes qui doivent être pré-lues par Oracle

`bool ocisetprefetch ( resource stmt , int rows )`

`ocietprefetch` spécifie le nombre des premières lignes qui doivent être pré-lues à ^{rows} pour la connexion Oracle ^{stmt}. La valeur par défaut de ^{rows} est 1 ligne.

7.73.54 ocistatementtype() : Retourne le type de commande OCI

string **ocistatementtype** (resource stmt)

`ocistatementtype` 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

Exemple avec `ocistatementtype`

```
<?php
print "<HTML><PRE>";
$conn = OCILogon("scott","tiger");
$sql = "delete from emp where deptno = 10";

$stmt = OCIParse($conn,$sql);
if ( OCISTatementType($stmt) == "DELETE" ) {
    die "Nous n'êtes pas autorisés à effacer des lignes dans cette table<BR>";
}

OCILogoff($conn);
print "</PRE></HTML>";
?>
```

7.73.55 ociwritelobtofile() : Ecrit un LOB Oracle dans un fichier

bool **ociwritelobtofile** (string filename , int start , int length)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.73.56 ociwritetemporarylob() : Ecrit un blob Oracle temporaire

bool **ociwritetemporarylob** (string var , int lob_type)

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.74 OpenSSL

7.74.1 Introduction

Cette extension utilise les fonctions de OpenSSL pour générer et vérifier les signatures, ainsi que pour sceller (chiffrer) et ouvrir (déchiffrer) les données. Vous avez besoin de OpenSSL >= 0.9.5 pour utiliser ce module.

7.74.2 Pré-requis

Afin de pouvoir utiliser les fonctions OpenSSL, vous devez installer les fonctions OpenSSL . PHP-4.0.4pl1 requiert OpenSSL >= 0.9.6, mais PHP-4.0.5 est plus récent fonctionneront aussi avec OpenSSL >= 0.9.5.

7.74.3 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 librairies **libeay32.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).

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 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 librairie 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 l'installer 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.

7.74.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.74.5 Types de ressources

7.74.6 Constantes prédefinies

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.

7.74.6.1 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

7.74.6.2 Options de remplissage (Padding)

OPENSSL_PKCS1_PADDING entier
 OPENSSL_SSLV23_PADDING entier
 OPENSSL_NO_PADDING entier
 OPENSSL_PKCS1_OAEP_PADDING entier

7.74.6.3 Types de clés

OPENSSL_KEYTYPE_RSA entier
 OPENSSL_KEYTYPE_DSA entier
 OPENSSL_KEYTYPE_DH entier

7.74.6.4 Constantes/options PKCS7

Les fonctions S/MIME utilisent des options qui sont spécifiés 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 spécification 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 utilisé pour rechercher le certificat de signature. Avec cette option, seul le certificat spécifié par le paramètre extracerts de la fonction <code>openssl_pkcs7_verify</code> est utilisé. Les certificats fournis peuvent toujours être utilisé, 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 local du certificat du signataire (passée au paramètre extracerts , 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 claire, avec le type MIME "multipart/signed" . C'est la valeur par défaut du paramètre flags 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.

7.74.7 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/privée
 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.

7.74.8 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 contenant les tiers de confiance. Si un dossier est spécifié, il doit être correct, car `openssl` va l'utiliser.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Paramètres clés/certificats](#)
- [Vérification de certificats](#)
- [`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_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](#)

7.74.10 [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. Si 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.

Le paramètre optionnel ^{notext} affecte la quantité de données affichées par l'opération : si ce paramètre vaut ^{FALSE}, alors des informations supplémentaires et compréhensibles sont ajoutées dans l'affichage. La valeur par défaut de ^{notext} est ^{TRUE}.

Voir aussi [openssl_csr_export_to_file](#), [openssl_csr_new](#) et [openssl_csr_sign](#).

7.74.11 [openssl_csr_new\(\)](#) : Génère une clé privée et un CSR

`bool openssl\_csr\_new ( array dn , resource privkey , array configargs , array extraattrs )`

[openssl_csr_new](#) génère une nouvelle CSR (Certificate Signing Request, requête de signature de certificat), basé sur les informations fournies par ^{dn}, qui représente le nom distingué (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 <code>openssl.cnf</code> valide pour que cette fonction opère correctement. Voyez les notes dans la section installation pour plus d'informations.

Par défaut, les informations du fichier `openssl.cnf` 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.cnf`, tel qu'indiqué ci-dessous.

Index configargs	Type	Equivalent openssl.cnf	Description
<code>digest_alg</code>	<u>chaîne de caractères</u>	<code>default_md</code>	Sélectionne la méthode "digest" à utiliser
<code>x509_extensions</code>	<u>chaîne de caractères</u>	<code>x509_extensions</code>	Sélectionne quelles extensions doivent être utilisées lors de la création du certificat x509
<code>req_extensions</code>	<u>chaîne de caractères</u>	<code>req_extensions</code>	Sélectionne quelles extensions doivent être utilisées lors de la création d'une CSR
<code>private_key_bits</code>	<u>integer</u>	<code>default_bits</code>	Spécifie le nombre de bits à utiliser lors de la génération de la clé privée
<code>private_key_type</code>	<u>integer</u>	<code>none</code>	Spécifie le type de clé privée à générer. Cela peut être l'une des constantes suivantes : <code>OPENSSL_KEYTYPE_DSA</code> , <code>OPENSSL_KEYTYPE_DH</code> ou <code>OPENSSL_KEYTYPE_RSA</code> . La valeur par défaut est <code>OPENSSL_KEYTYPE_RSA</code> , ce qui est le seul type de clé actuellement supporté.
<code>encrypt_key</code>	<u>boolean</u>	<code>encrypt_key</code>	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 <code>openssl_csr_new</code> - création d'un certificat auto chiffré
<pre>// 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 la clé privé 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 debug_zval_dump(\$csrout); openssl_x509_export(\$sscert, \$certout) and debug_zval_dump(\$certout); openssl_pkey_export(\$privkey, \$pkeyout, "mypassword") and debug_zval_dump(\$pkeyout);</pre>


```
// Affiche les erreurs qui sont survenues
while (($e = openssl_error_string()) !== false) {
    echo $e . "\n";
}
```

7.74.12 openssl_csr_sign() : Signe un certificat avec un autre certificat

resource **openssl_csr_sign** (mixed csr , mixed cacert , mixed priv_key , int days)

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 jour.

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)

```
// Supposent que ce script est configuré pour recevoir des CSR qui ont
// été collé 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'ai 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, ce façon à ce que l'utilisateur
// puisse le copier/coller dans leur configuration locale ( comme par exemple un
// fichier qui contient les certificats de leur serveur SSL)
openssl_x509_export($usercert, $certout) and echo $certout;

// Affiche toutes les erreurs survenues
while (($e = openssl_error_string()) !== false) {
    echo $e . "\n";
}
```

7.74.13 openssl_error_string() : Retourne le message d'erreur OpenSSL

mixed **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 librairie OpenSSL. Les messages d'erreurs sont empilés, et openssl_error_string doit être appelé plusieurs fois pour afficher toutes les erreurs.

Les paramètres et le type de retour de cette fonction risquent d'évoluer d'ici à la prochaine version de PHP.

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>";
}
?>
```

Note

Ces constantes ont été ajoutées en PHP 4.0.6.

7.74.14 `openssl_free_key()` : Libère les ressources

`void openssl_free_key ( resource key_identifiant )`

`openssl_free_key` libère les ressources associées à `key_identifiant`.

7.74.15 `openssl_get_privatekey()` : Prépare une clé privée au format PEM

`resource openssl_get_privatekey ( mixed key , string passphrase )`

`openssl_get_privatekey` retourne un identifiant de clé positif, ou `FALSE` en cas d'erreur.

`openssl_get_privatekey` analyse la clé privée `key`, au format PEM, et la prépare pour à être utilisée par d'autres fonctions. Le paramètre optionnel `passphrase` doit être utilisé si la clé est chiffrée (protégée par un mot de passe).

7.74.16 `openssl_get_publickey()` : Extrait une clé publique d'un certificat

`resource openssl_get_publickey ( mixed certificate )`

`openssl_get_publickey` retourne un identifiant de clé positif, ou `FALSE` en cas d'erreur.

`openssl_get_publickey` extrait la clé publique du certificat `certificate` (format X.509), et la prépare à être utilisée ultérieurement.

7.74.17 `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](#) .

7.74.18 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` .

Les paramètres et le type de retour de cette fonction risquent d'évoluer d'ici à la prochaine version de PHP.

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!";
?>

```

Note

Ces constantes ont été ajoutées en PHP 4.0.6.

7.74.19 openssl_pkcs7_encrypt() : Chiffre un message S/MIME

`bool openssl_pkcs7_encrypt ( string infile , string outfile , mixed recipcerts , array headers , int flags )`

`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 ajouté 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 une en-tête complète.

Les paramètres et le type de retour de cette fonction risquent d'évoluer d'ici à la prochaine

version de PHP.

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 vous votre lecture!
L'ennemi approche! Rendez-vous au café à 8h30,
pour votre faux passeport.
HQ
EOD;
// 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", "nighthawk.pem",
    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");
}
?>

```

Note

Ces constantes ont été ajoutées en PHP 4.0.6.

7.74.20 `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 ajouté au 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 certificat supplémentaires à inclure dans la signature, qui pourront aider le destinataire à vérifier les données que vous utilisez.

Les paramètres et le type de retour de cette fonction risquent d'évoluer d'ici à la prochaine version de PHP.

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("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");
}
?>

```

Note

Ces constantes ont été ajoutées en PHP 4.0.6.

7.74.21 openssl_pkcs7_verify() : Vérifie la signature d'un message S/MIME

bool **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 certificat utilisé comme certificats de peu de confiance.

Les paramètres et le type de retour de cette fonction risquent d'évoluer d'ici à la prochaine version de PHP.

Note

Ces constantes ont été ajoutées en PHP 4.0.6.

7.74.22 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.

7.74.23 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.

7.74.24 openssl_pkey_get_private() : Lit une clé privée

```
resource openssl_get_privatekey ( mixed key , string passphrase )
```

`openssl_get_privatekey` lit une ressource de clé privée en cas de succès, et `FALSE` sinon.

`openssl_get_privatekey` 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 privé 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).

7.74.25 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 être l'une des valeurs suivantes :

1. une chaîne au format `file://path/to/file.pem` . Le fichier ainsi désigné doit contenir une clé privé ou un certificat au format PEM (éventuellement, les deux).
2. Une clé privée au format PEM.

7.74.26 openssl_pkey_new() : Génère une nouvelle clé privée

```
resource openssl_pkey_new ( array configargs )
```

`openssl_pkey_new` génère une nouvelle paire de clé. La clé privée est accessible via la fonction `openssl_pkey_get_public`. Vous pouvez calibrer la génération de la clé (comme par exemple le nombre de bits) en utilisant le paramètre `configargs`. Voyez 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.

7.74.27 `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 chiffré 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`.

7.74.28 `openssl_private_encrypt()` : Chiffre des données avec une clé privée

```
bool openssl_private_encrypt ( string data , string &encrypted , mixed key , int padding )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.74.29 `openssl_public_decrypt()` : Déchiffre des données avec une clé publique

```
bool openssl_public_decrypt ( string data , string &decrypted , resource key , int padding )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.74.30 `openssl_public_encrypt()` : Chiffre des données avec une clé publique

```
bool openssl_public_encrypt ( string data , string crypt , mixed key , int padding )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.74.31 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 encryptée 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 encryptées et la clé d'enveloppe, qui a été encryptée 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 .

7.74.32 openssl_sign() : Signe les données

```
bool openssl_sign ( string data , string signature , mixed priv_key_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.

openssl_sign calcule la signature des données `data` en utilisant l'algorithme SHA1 (hashing) suivi du chiffage 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
$fpr = fopen("/src/openssl-0.9.6/demos/sign/key.pem", "r");
$priv_key = fread($fpr, 8192);
fclose($fpr);
$pkid = openssl_get_privatekey($priv_key);
// calcule de la signature
openssl_sign($data, $signature, $pkid);
// libère les clés de la mémoire
openssl_free_key($pkid);
?>
```

Voir aussi `openssl_verify` .

7.74.33 openssl_verify() : Vérifie une signature

```
int openssl_verify ( string data , string signature , mixed pub_key_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.

`openssl_verify` retourne 1 si la signature est correcte, 0 si la signature est incorrecte, et -1 en cas d'erreur.

`openssl_verify` vérifie que la signature ^{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
$fpr = fopen("/src/openssl-0.9.6/demos/sign/cert.pem", "r");
$cert = fread($fpr, 8192);
fclose($fpr);
$pubkeyid = openssl_get_publickey($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` .

7.74.34 openssl_x509_check_private_key() : Vérifie si une clé privée correspon à un CERT

```
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.

7.74.35 openssl_x509_checkpurpose() : Vérifie l'usage d'un certificat

`bool 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.

`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.

Les paramètres et le type de retour de cette fonction risquent d'évoluer d'ici à la prochaine version de PHP.

Constante	Description
<code>X509_PURPOSE_SSL_CLIENT</code>	Est ce que le certificat peut être utilisé avec le client d'une connexion SSL?
<code>X509_PURPOSE_SSL_SERVER</code>	Est ce que le certificat peut être utilisé avec le serveur d'une connexion SSL?
<code>X509_PURPOSE_NS_SSL_SERVER</code>	Est ce que le certificat peut être utilisé avec un serveur Netscape d'une connexion SSL?
<code>X509_PURPOSE_SMIME_SIGN</code>	Est ce que le certificat peut être utilisé pour signer des courrier à la norme S/MIME?
<code>X509_PURPOSE_SMIME_ENCRYPT</code>	Est-ce que le certificat peut être utilisé pour chiffrer un courrier au format S/MIME?
<code>X509_PURPOSE_CRL_SIGN</code>	Est-ce que le certificat peut être utilisé pour chiffrer une liste de revocation de certificat? (CRL)?
<code>X509_PURPOSE_ANY</code>	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.

Note
Ces constantes ont été ajoutées en PHP 4.0.6.

7.74.36 openssl_x509_export_to_file() : Exporte un CERT vers un fichier ou une variable

`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 `notext` vaut par défaut ^{TRUE}. Si il est mis à ^{FALSE}, un texte supplémentaire et lisible sera aussi stocké dans le fichier ^{output}.

Le paramètre optionnel ^{notext} affecte le niveau de détails du résultat. Si 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.

7.74.37 openssl_x509_export() : Exporte un CERT vers une variable

`bool openssl_x509_export ( mixed x509 , string &output , bool notext )`

`openssl_x509_export` stocke un certificat `x509` dans le fichier `outfilename` au format PRM. Le paramètre `notext` vaut par défaut `TRUE`. Si il est mis à `FALSE`, un texte supplémentaire et lisible sera aussi stocké dans le fichier `output`.

Le paramètre optionnel `notext` affecte le niveau de détails du résultat. Si 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.

7.74.38 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`

Note

Ces constantes ont été ajoutées en PHP 4.0.6.

7.74.39 openssl_x509_parse() : Analyse un certificat X509

`array openssl_x509_parse ( mixed x509cert , bool shortnames )`

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`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.

Note

Ces constantes ont été ajoutées en PHP 4.0.6.

7.74.40 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.

Note

Ces constantes ont été ajoutées en PHP 4.0.6.

7.75 Oracle

7.75.1 Introduction

Cette extension ajoute le support des bases de données Oracle. Voyez aussi [OCI8](#).

7.75.2 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.

7.75.3 Constantes prédefinies

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 prédefinies](#)
- [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](#)

7.75.5 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 .

7.75.6 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 majuscules.

7.75.7 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` .

7.75.8 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` . Le type retourné prendra une des valeurs suivantes :

- "VARCHAR2"
- "VARCHAR"
- "CHAR"
- "NUMBER"
- "LONG"
- "LONG RAW"
- "ROWID"
- "DATE"
- "CURSOR"

7.75.9 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 .

7.75.10 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.

7.75.11 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.

7.75.12 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 détails sur les erreurs est accessible avec les fonctions ora_error et ora_errorcode.

Voir aussi ora_parse, ora_exec et ora_fetch.

7.75.13 ora_error() : Retourne le message d'erreur Oracle

`string ora_error ( resource cursor_or_connection )`

ora_error retourne un messages 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'erreurs tels que : \$

```
Array
(
    [0] => 00001, 00000,
    [1] => "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
)
```

7.75.14 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 exécuté 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.

7.75.15 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 .

7.75.16 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} .

ora_fetch_into retourne le nombre de colonnes lues.

Exemple avec ora_fetch_into

```
<?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'];
?>
```

Notez que vous devez passer le tableau par référence;

Voir aussi ora_parse , ora_exec , ora_fetch et ora_do .

7.75.17 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, sa valeur sera disponible dans les fonctions ora_error et ora_errorcode.

Voir aussi ora_parse, ora_exec et ora_do.

7.75.18 ora_getcolumn() : Retourne une donnée d'une ligne de résultat Oracle

mixed **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".

7.75.19 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 détails sur les erreurs est accessible avec les fonctions ora_error et ora_errorcode.

Voir aussi ora_logon.

7.75.20 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 les noms d'utilisateur ^{user} et le mot de passe ^{password}.

Les connexions peut être faites 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 configuré 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.

7.75.21 ora_numcols() : Retourne le nombre de colonnes

int **ora_numcols** (resource cursor)

ora_numcols retourne le nombre de colonne 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.

7.75.22 ora_numrows() : Retourne le nombre de colonnes

int **ora_numrows** (resource cursor)

ora_numrows retourne le nombre de colonnes dans le résultat `cursor_ind`.

7.75.23 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.

7.75.24 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_ind`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec..

Voir aussi ora_exec, ora_fetch et ora_do.

7.75.25 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.

7.75.26 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](#) .

7.76 Ovrimos SQL

7.76.1 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 n'est pas disponible sur les plate-formes Windows.

7.76.2 Pré-requis

Ovrimos SQL Server est disponible à www.ovrimos.com . Vous devez installer la librairie sqlcli, qui est disponible dans la distribution de Ovrimos SQL Server.

7.76.3 Installation

Pour activer le support d'Ovrimos support en ^{PHP}, vous devez compiler ^{PHP} avec l'option `--with-ovrimos[=DIR]` . DIR est le chemin jusqu'au dossier d'Ovrimos qui contient la librairie libsqlcli.

7.76.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.76.5 Types de ressources

7.76.6 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.

- [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](#)

7.76.8 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.

7.76.9 ovrimos_connect() : Connexion à un serveur

int **ovrimos_connect** (string host , string db , string user , string password)

[ovrimos_connect](#) sert à se connecter 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.

7.76.10 ovrimos_cursor() : Retourne le nom du curseur Ovrimos

```
string ovrimos_cursor ( int result_id )
```

ovrimos_cursor sert à lire le nom du curseur

ovrimos_cursor retourne le nom du curseur. Pratique, lorsqu'on veut faire des modifications ou des effacements avec des curseurs déjà positionnés.

7.76.11 ovrimos_exec() : Exécute une requête SQL

```
int ovrimos_exec ( int connection_id , string query )
```

ovrimos_exec sert à exécuter une requête SQL.

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.

7.76.12 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'interrogations 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.

7.76.13 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 un résultat SQL.

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.

7.76.14 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.

7.76.15 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 ^{field_number}, dans le résultat ^{field_number}.

7.76.16 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).

7.76.17 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`.

7.76.18 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 `field_number`.

7.76.19 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`.

7.76.20 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`.

7.76.21 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`.

7.76.22 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.

7.76.23 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.

7.76.24 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);
}
?>

```

7.76.25 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.

7.76.26 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.

7.77 Bufferisation de sortie

7.77.1 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 envoyées par la fonction `header` ou les cookies envoyés par `setcookie`. Seules les fonctions telles que `echo` et les données entre blocs `PHP` sont affectées.

7.77.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.77.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.77.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>output_buffering</code>	<code>"0"</code>	<code>PHP_INI_PERDIR PHP_INI_SYSTEM</code>
<code>output_handler</code>	<code>NULL</code>	<code>PHP_INI_PERDIR PHP_INI_SYSTEM</code>
<code>implicit_flush</code>	<code>"0"</code>	<code>PHP_INI_PERDIR PHP_INI_SYSTEM</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

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 de `0`. 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 `0`. Par exemple, `output_buffering=4096`).

`output_handler` chaîne de caractères

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 adaptée 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_inconv_handler` et non plus `ob_gzhandler` et `zlib.output_compression`.

`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 le débogage. Cette valeur est par défaut à `TRUE` lorsque PHP fonctionne en mode `CLI SAPI`.

Voir aussi [ob_implicit_flush](#).

7.77.5 Types de ressources

Cette extension ne définit aucune ressource.

7.77.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.77.7 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.

7.77.8 Voir aussi

Voir aussi [header](#) et [setcookie](#).

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [Voir aussi](#)
- [flush](#)
- [ob_clean](#)
- [ob_end_clean](#)
- [ob_end_flush](#)
- [ob_flush](#)
- [ob_get_clean](#)
- [ob_get_contents](#)

- [ob_get_length](#)
- [ob_get_level](#)
- [ob_get_status](#)
- [ob_gzhandler](#)
- [ob_implicit_flush](#)
- [ob_start](#)

7.77.10 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](#) .

7.77.11 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_flush](#) . La fonction ^{TRUE} lorsqu'elle a réussi à détruire correctement le buffer, et ^{FALSE} sinon. Les raisons d'un tel échec sont la bufferisation de sortie pouvaient ne pas être activé, 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](#)

```
<?php
while (@ob_end_clean());
?>
```

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](#) .

7.77.12 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, et ^{FALSE} otherwise. 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).

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_flush et ob_end_clean .

7.77.13 ob_flush() : Envoie le buffer de sortie

void **ob_flush** (void)

ob_flush envoie le contenu du buffer de sortie (si 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_clean .

Voir aussi ob_get_contents , ob_clean , ob_end_flush et ob_end_clean .

7.77.14 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 la bufferisation n'était 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);
/* Notre exemple va afficher :
string(16) "Bonjour le monde"
*/
?>
```

Voir aussi [ob_start](#) et [ob_get_contents](#) .

7.77.15 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](#) .

7.77.16 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](#) .

7.77.17 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](#) .

7.77.18 ob_get_status() : Lit le statut du buffer de sortie

array **ob_get_status** (bool full_status)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[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](#) .

7.77.19 ob_gzhandler() : Fonction de callback pour la compression automatique des buffers

string **ob_gzhandler** (string buffer , int mode)

[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 une en-tête indiquant les types de pages supportés.

Note

Vous ne pouvez pas utiliser simultanément [ob_gzhandler](#) et [ini.zlib.output_compression](#). De plus, notez bien que [ini.zlib.output_compression](#) est préférable à [ob_gzhandler](#).

Exemple d'envoi de page compressée avec [ob_gzhandler](#)

```
<?php
  ob_start("ob_gzhandler");
?>
<HTML>
<BODY>
<P>Ceci devrait être une page compressée.
</HTML>
</BODY>
```

Voir aussi [ob_start](#) et [ob_end_flush](#).

7.77.20 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](#).

7.77.21 ob_start() : Enclenche la bufferisation de sortie

bool **ob_start** (callback output_callback)

[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.

Note

En PHP 4.0.4, [ob_gzhandler](#) a été introduit pour faciliter l'envoi de fichier compressé 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.

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.

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.
</body>
</html>
<?php
ob_end_flush();
?>
```

va afficher :

```
<html>
<body>
<p>C'est comme comparer des carottes et des carottes.
</body>
</html>
```

Voir aussi [ob_get_contents](#) , [ob_end_flush](#) , [ob_end_clean](#) , [ob_implicit_flush](#) et [ob_gzhandler](#) .

7.78 Overload

7.78.1 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 supporte cet 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 et modifier un membre, et 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.

7.78.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.78.3 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 librairie supplémentaire pour disposer de ces fonctions.

Note

Le support par défaut de l'extension overload a été ajouté en PHP 4.3.0.

7.78.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.78.5 Types de ressources

Cette extension ne définit aucune ressource.

7.78.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.78.7 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;
print "\$o->a: $o->a\n"; // print: $o->a:
print "\$o->b: $o->b\n"; // print: $o->b: 9
print "\$o->c: $o->c\n"; // print: $o->c: 42
print "\$o->d: $o->d\n"; // print: $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);

?>
```

Attention

Comme c'est une extension expérimentale, tout ne fonctionne pas encore. Il n'y a pas de support pour `__call()` actuellement, et nous ne pouvez overloader que des accesseurs. Vous ne pouvez pas appeler les fonctions d'overload de la classe, et `__set()` ne fonctionne que pour un seul niveau d'accesseur.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [overload](#)

7.79 PDF

7.79.1 Introduction

Les fonctions PDF permettent de créer des fichiers PDF en utilisant la librairie 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 intéressés par des alternatives gratuites pour générer des PDF, sans passer par des librairies PDF, voyez cette entrée de la FAQ .

7.79.2 Pré-requis

PDFlib est disponible en téléchargement à <http://www.pdflib.com/products/pdflib/index.html> , mais impose l'achat d'une license pour l'utiliser commercialement. Les librairies [JPEG](#) et [TIFF](#) sont nécessaires pour faire fonctionner cette librairie.

7.79.2.1 Problèmes avec les anciennes versions de PDFlib

Toutes les versions des PHP 4 éditées après le 9 Mars 2000 ne supportent pas les versions de PDFlib plus anciennes que PDFlib 3.0.

PDFlib 3.0 est plus récent est supporté par PHP 3.0.19 et plus récent.

7.79.3 Installation

Pour inclure le support de la PDFlib dans votre `PHP` , 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 librairies 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-pdf-lib`.

7.79.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.79.5 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 adoptée cette version comme le minimum nécessaire pour supporter PDF. En conséquence de quoi, un grand nombre de fonction vont disparaître, ou être remplacée. 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
<code>pdf_put_image</code>	Désormais inutile.
<code>pdf_execute_image</code>	Désormais inutile.
<code>pdf_get_annotation</code>	<code>pdf_get_bookmark</code> en utilisant les mêmes paramètres.
<code>pdf_get_font</code>	<code>pdf_get_value</code> en passant "font" comme second paramètre.
<code>pdf_get_fontsize</code>	<code>pdf_get_value</code> en passant "fontsize" comme second paramètre.
<code>pdf_get_fontname</code>	<code>pdf_get_parameter</code> en passant "fontname" comme second paramètre.
<code>pdf_set_info_creator</code>	<code>pdf_set_info</code> en passant "Creator" comme second paramètre.
<code>pdf_set_info_title</code>	<code>pdf_set_info</code> en passant "Title" comme second paramètre.
<code>pdf_set_info_subject</code>	<code>pdf_set_info</code> en passant "Subject" comme second paramètre.
<code>pdf_set_info_author</code>	<code>pdf_set_info</code> en passant "Author" comme second paramètre.
<code>pdf_set_info_keywords</code>	<code>pdf_set_info</code> en passant "Keywords" comme second paramètre.
<code>pdf_set_leading</code>	<code>pdf_set_value</code> en passant "leading" comme second paramètre.
<code>pdf_set_text_rendering</code>	<code>pdf_set_value</code> en passant "textrendering" comme second paramètre.
<code>pdf_set_text_rise</code>	<code>pdf_set_value</code> en passant "textrise" comme second paramètre.
<code>pdf_set_horiz_scaling</code>	<code>pdf_set_value</code> en passant "horizscaling" comme second paramètre.
<code>pdf_set_text_matrix</code>	Désormais abandonnée.
<code>pdf_set_char_spacing</code>	<code>pdf_set_value</code> en passant "charspacing" comme second paramètre.
<code>pdf_set_word_spacing</code>	<code>pdf_set_value</code> en passant "wordspacing" comme second paramètre.
<code>pdf_set_transition</code>	<code>pdf_set_parameter</code> en passant "transition" comme second paramètre.
<code>pdf_open</code>	<code>pdf_new</code> suivi d'un appel à <code>pdf_open_file</code>
<code>pdf_set_font</code>	<code>pdf_findfont</code> suivi d'un appel à <code>pdf_setfont</code>
<code>pdf_set_duration</code>	<code>pdf_set_value</code> en passant "duration" comme second paramètre.
<code>pdf_open_gif</code>	<code>pdf_open_image_file</code> en passant "gif" comme second paramètre.
<code>pdf_open_jpeg</code>	<code>pdf_open_image_file</code> en passant "jpeg" comme second paramètre.
<code>pdf_open_tiff</code>	<code>pdf_open_image_file</code> en passant "tiff" comme second paramètre.
<code>pdf_open_png</code>	<code>pdf_open_image_file</code> en passant "png" comme second paramètre.
<code>pdf_get_image_width</code>	<code>pdf_get_value</code> en passant "imagewidth" comme second paramètre et l'image en troisième paramètre.
<code>pdf_get_image_height</code>	<code>pdf_get_value</code> en passant "imageheight" comme second paramètre et l'image en troisième paramètre.

7.79.6 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", de taille de 30pt. Le texte est aussi souligné.

Création d'un document PDF avec PDFlib
<pre><?php \$pdf = pdf_new(); pdf_open_file(\$pdf, "test.pdf"); pdf_set_info(\$pdf, "Author", "Uwe Steinmann");</pre>

```
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($pdf, "Page 1");
pdf_set_font($pdf, "Times-Roman", 30, "host");
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.

```
<?php
$len = filesize($filename);
header("Content-type: application/pdf");
header("Content-Length: $len");
header("Content-Disposition: inline; filename=foo.pdf");
readfile($filename);
?>
```

La distribution de la PDFlib contient des exemples plus complexes, qui créent un 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, "")) {
    print 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);
    }
}
```

```

$stime = getdate();

/* l'aiguille des heures */
pdf_save($pdf);
pdf_rotate($pdf, -(($stime['minutes']/60.0)+$stime['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, -(($stime['seconds']/60.0)+$stime['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, -(($stime['seconds'] - 15.0) * 6.0));
pdf_moveto($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");
print $buf;

pdf_delete($pdf);
?>

```

7.79.7 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_locallink](#)
- [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](#)

7.79.9 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} = 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.

Exemple avec [pdf_add_bookmark](#)

```
<?php
// create a new PDF

$pdf = pdf_new();
pdf_open_file($pdf);
pdf_set_info($pdf, "Author", "Bob Nijman");

// begin a new page
pdf_begin_page($pdf, 300, 300);

// add a top-level bookmark
$bookmark = pdf_add_bookmark($pdf, "People");

// add a nested bookmark
pdf_add_bookmark($pdf, "Rasmus", $bookmark);

// and some text
pdf_set_font($pdf, "Helvetica", 20, "host");
$text = "This is R's page";
$width = pdf_stringwidth($pdf, $text);
pdf_set_text_pos($pdf, (300-$width)/2, 100);
pdf_show($pdf, $text);

// close the page and the PDF
pdf_end_page($pdf);
pdf_close($pdf);

?>
```

7.79.10 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](#) .

7.79.11 pdf_add_loclink() : Ajoute une annotation de lien dans la page PDF courante

`bool pdf_add_loclink ( resource pdfdoc , float lowerleftx , float lowerlefty , float upperrightx , float upperrighty , int page , string dest )`

[pdf_add_loclink](#) 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`, `fitheight` ou `fitbbox`.

Voir aussi [pdf_add_launchlink](#).

7.79.12 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.

7.79.13 pdf_add_outline() : Obsolète : Ajoute un signet dans la page courante

Obsolète.

Voir [pdf_add_bookmark](#).

7.79.14 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_loclink](#) et [pdf_add_weblink](#).

7.79.15 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. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.79.16 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.

7.79.17 pdf_arc() : Dessine un arc de cercle PDF dans le sens anti-horaire

```
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 anti-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. Cette fonction retourne **TRUE** en cas de succès, **FALSE** en cas d'échec.

Voir aussi [pdf_arcn](#) et [pdf_circle](#) .

7.79.18 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_arcn 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. Cette fonction retourne **TRUE** en cas de succès, **FALSE** en cas d'échec.

Voir aussi [pdf_arc](#) et [pdf_circle](#) .

7.79.19 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.

7.79.20 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](#) .

7.79.21 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](#) .

7.79.22 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.

7.79.23 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` . Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [pdf_arc](#) et [pdf_arcn](#) .

7.79.24 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.

7.79.25 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](#) .

7.79.26 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 cette page. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.79.27 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 utilisée en entrée. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

Voir aussi pdf_open_pdi .

7.79.28 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 .

7.79.29 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 .

7.79.30 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_fill_stroke .

7.79.31 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](#).

7.79.32 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.

7.79.33 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. L'espacement entre les lignes est déterminé avec le paramètre `leading`. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.79.34 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.

7.79.35 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](#).

7.79.36 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](#).

7.79.37 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 .

7.79.38 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.

7.79.39 pdf_endpath() : Obsolète : Termine le chemin courant

Obsolète. Utilisez une des fonctions de pinceau, remplissage ou de clip.

7.79.40 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 courant, en appliquant le pinceau courant (stroke). Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi pdf_setcolor .

7.79.41 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 .

7.79.42 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.

`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);
}
?>
```

7.79.43 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.

7.79.44 pdf_get_font() : Obsolète: charge une police

Obsolète.

Voir `pdf_get_value` .

7.79.45 pdf_get_fontname() : Obsolète : lit le nom de la police

Obsolète.

Voir `pdf_get_parameter` .

7.79.46 pdf_get_fontsize() : Obsolète : gère les polices

Obsolète.

Voir `pdf_get_value` .

7.79.47 pdf_get_image_height() : Retourne la hauteur d'une image

string **pdf_get_image_height** (int pdf_object , int image)

`pdf_get_image_height` est obsolète. Utilisez plutôt `pdf_get_value` .

7.79.48 pdf_get_image_width() : Retourne la largeur d'une image

```
string pdf_get_image_width ( int pdf_object , int image )
```

pdf_get_image_width est obsolète. Utilisez plutôt pdf_get_value .

7.79.49 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 .

7.79.50 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 .

7.79.51 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.

7.79.52 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.

7.79.53 pdf_get_pdi_value() : Lit des paramètres numériques dans le document PDF d'entrée

```
string 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.

7.79.54 pdf_get_value() : Lit des paramètres numériques

```
float pdf_get_value ( resource pdfdoc , string key , float modifier )
```

pdf_get_value lit des paramètres numériques du document `pdf_object` , identifié par `key` .

7.79.55 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.

7.79.56 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.

7.79.57 pdf_makespotcolor() : Place un point de couleur PDF

```
bool 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](#) .

7.79.58 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.

7.79.59 pdf_new() : Crée un nouvel objet PDF

```
resource pdf_new ( )
```

[pdf_new](#) crée un nouvel objet PDF, en utilisant le système de gestion d'erreur et celui de gestion de la mémoire.

Voir aussi [pdf_close](#) .

7.79.60 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 blackls1 )
```

[pdf_open_ccitt](#) ouvre une image brute CCITT. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.79.61 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](#) fonction. 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
<pre><?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, "A PDF document created in memory!", 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; ?></pre>

7.79.62 pdf_open_gif() : Obsolète : Ouvre une image GIF

Obsolète.

Voir [pdf_open_image](#) ,

7.79.63 pdf_open_image_file() : Lit une image depuis un fichier

```
int pdf_open_image_file ( resource pdfdoc , string imagetype , string filename , string stringparam , string intparam )
```

[pdf_open_image_file](#) ouvre une image depuis le fichier `filename`. Les formats supportés sont `jpeg`, `tiff`, `gif`, et `png`. `stringparam` vaut soit `""`, `mask`, `masked` ou `page`. `intparam` vaut soit 0, soit la ressource d'image appliquée au masque, soit la ressource de page.

7.79.64 pdf_open_image() : Ouvre une image

```
int pdf_open_image ( resource pdf-document , string imagetype , string source , string data , long 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`.

7.79.65 pdf_open_jpeg() : Obsolète : Ouvre une image JPEG

Cette fonction est obsolète, utilisez plutôt [pdf_open_image](#).

7.79.66 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](#).

7.79.67 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](#)

7.79.68 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](#) .

7.79.69 pdf_open_png() : Obsolète : ouvre une image PNG

Obsolète.

Reportez-vous à [pdf_open_image](#) .

7.79.70 pdf_open_tiff() : Obsolète : Ouvre une image TIFF

Cette fonction est obsolète, utilisez plutôt [pdf_open_image](#) .

7.79.71 pdf_open() : Obsolète : Ouvre un nouvel objet PDF

Cette fonction est obsolète, utilisez de préférence [pdf_new](#) avec [pdf_open_file](#) .

7.79.72 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.

7.79.73 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 PDF, 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.

7.79.74 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.

7.79.75 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.

7.79.76 pdf_rotate() : Configure la rotation

```
bool pdf_rotate ( resource pdfdoc , float phi )
```

pdf_rotate fait pivoter le système de coordonnées de ^{phi} degrés. Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.79.77 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.

7.79.78 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.

7.79.79 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.

7.79.80 pdf_set_border_dash() : Configure le style des lignes autour des liens at 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 .

7.79.81 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.

7.79.82 pdf_set_char_spacing() : Obsolète : Configure l'espacement de caractères character spacing

Cette fonction est obsolète. Utilisez plutôt [pdf_set_value](#) .

7.79.83 pdf_set_duration() : Obsolète : Configure la durée entre deux pages

Obsolète.

Voir [pdf_set_value](#) .

7.79.84 pdf_set_font() : Obsolète : Sélectionne la taille et le nom d'une police

Obsolète. Utilisez plutôt [pdf_findfont](#) suivie de [pdf_setfont](#) .

Voir [pdf_findfont](#) et [pdf_setfont](#) .

7.79.85 pdf_set_horiz_scaling() : Obsolète : Configure l'échelle horizontale du texte

```
void pdf_set_horiz_scaling ( int pdf_object , float scale )
```

Cette fonction est bsolète. Utilisez plutôt [pdf_set_value](#) .

7.79.86 pdf_set_info_author() : Obsolète : remplit le champ d'auteur du document

```
bool pdf_set_info_author ( int pdfdoc , string author )
```

Obsolète. Utilisez [pdf_set_info](#) .

7.79.87 pdf_set_info_creator() : Obsolète : remplit le champ de créateur du document

```
bool pdf_set_info_creator ( int pdfdoc , string creator )
```

Obsolète. Utilisez [pdf_set_info](#) .

7.79.88 pdf_set_info_keywords() : Obsolète : remplit le champ de mots clés du document

```
bool pdf_set_info_keywords ( int pdfdoc , string keywords )
```


Obsolète. Utilisez [pdf_set_info](#) .

7.79.89 pdf_set_info_subject() : Obsolète : remplit le champ de sujet du document

```
bool pdf_set_info_subject ( int pdfdoc , string subject )
```

Obsolète. Utilisez [pdf_set_info](#) .

7.79.90 pdf_set_info_title() : Obsolète : remplit le champ de titre du document

```
bool pdf_set_info_title ( int pdfdoc , string title )
```

Obsolète. Utilisez [pdf_set_info](#) .

7.79.91 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.

7.79.92 pdf_set_leading() : Obsolète : Configure la distance entre deux lignes de texte

Cette fonction est obsolète, utilisez de préférence [pdf_set_value](#) .

7.79.93 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.

7.79.94 pdf_set_text_matrix() : Obsolète : configure la matrice de texte

Voir [pdf_set_paramter](#) .

7.79.95 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. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.79.96 pdf_set_text_rendering() : Obsolète : Détermine le rendu du texte

Obsolète.

Voir [pdf_set_value](#).

7.79.97 pdf_set_text_rise() : Obsolète : configure l'élévation de texte

Obsolète.

Reportez vous à [pdf_set_value](#).

7.79.98 pdf_set_value() : Modifie les 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.

7.79.99 pdf_set_word_spacing() : Obsolète : configure l'espace entre deux mots

Obsolète.

Reportez vous à [pdf_set_value](#).

7.79.100 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`, `cmk`, `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.

Pour les gris, seul `c1` est utilisé.

Pour les couleurs `rgb`, les paramètres `c1`, `c2` et `c3` spécifie le rouge, le vert, et le bleu, respectivement.

Exemple avec <u>pdf_setcolor</u> : couleurs RGB

<pre>// Configure le blanc pour le dessin et le remplissage pdf_setcolor(\$pdf, "both", "rgb", 1, 1, 1);</pre>

Pour les couleurs `cmk`, les paramètres `c1`, `c2`, `c3` et `c4` représentent le cyan, le magenta, le jaune et le noir, respectivement

Exemple avec <u>pdf_setcolor</u> : couleurs CMYK

<pre>// Configure le blanc pour le dessin et le remplissage pdf_setcolor(\$pdf, "both", "cmyk", 0, 0, 0, 1);</pre>

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.

7.79.101 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.

7.79.102 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.

7.79.103 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.

7.79.104 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 .

7.79.105 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 .

7.79.106 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 .

7.79.107 pdf_setlinecap() : Configure le paramètre de linecap

```
void pdf_setlinecap ( resource pdfdoc , int linecap )
```

[pdf_setlinecap](#) configure le paramètre ^{linecap} à une valeur entre 0 et 2 inclusivement.

7.79.108 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.

7.79.109 pdf_setlinewidth() : Configure la largeur de ligne

```
void 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.

7.79.110 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.

7.79.111 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.

7.79.112 pdf_setpolydash() : Configure des pointillés complexes

```
void pdf_setpolydash ( int pdf_object , float * dasharray )
```

pdf_setpolydash configure des modèles de pointillés complexes.

7.79.113 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` 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 pdf_setcolor .

7.79.114 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 pdf_setcolor .

7.79.115 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 pdf_setcolor .

7.79.116 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 et la hauteur sont nulles, une simple ligne sera dessinée.

pdf_show_boxed retourne le nombre de caractères qui n'ont pas pu être dessiné dans le rectangle définit. pdf_show_boxed retourne 0 si tous les caractères ont pu être dessiné, ou si width et height sont nuls.

7.79.117 pdf_show_xy() : Affiche un text 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.

7.79.118 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, et avec la police et sa taille courante.

7.79.119 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 coordonnées. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.79.120 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 la police courante, configurée avec `pdf_setfont`. Si les paramètres optionnels de `font` et `size` sont spécifiés, la largeur sera calculée en utilisant cette police et cette taille. Notez que `font` est une ressource de police, retournée par `pdf_findfont`.

Note

Les paramètres <code>font</code> et <code>size</code> doivent être utilisé ensemble.

Voir aussi `pdf_setfont` et `pdf_findfont`.

7.79.121 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 clot le chemin. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.79.122 pdf_translate() : Translate l'origine du système de coordonnées

`bool pdf_translate ( resource pdfdoc , float tx , float ty )`

`pdf_translate` translate l'origine du système de coordonnées d'un vecteur de (`tx` , `ty`).

7.80 Paiement par Verisign

7.80.1 Introduction

Cette extension vous permet d'effectuer des transactions avec des cartes de crédits en utilisant les services Verisign Payment Services, anciennement connu sous le nom de Signio (<http://www.verisign.com/products/payflow/pro/index.html>).

Ces fonctions sont utilisables dès que PHP a été compilé avec l'option `--with-pfpro[=DIR]`. 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. Si vous avez l'intention d'utiliser cette extension sur un serveur web SSL ou avec d'autres composants SSL (tels que l'extension CURL et SSL) vous DEVEZ utiliser le SDK beta.

Ces fonctions ont été ajoutée 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 n'est pas disponible sur les plate-formes Windows.

7.80.2 Pré-requis

Une fois que vous avez téléchargé le SDK vous devez copier les fichiers depuis le dossier `lib` de la distribution. Copier le fichier d'en-têtes `pfpro.h` dans `/usr/local/include` et la librairie `libpfpro.so` dans `/usr/local/lib`.

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 transaction à traiter, ou que vous souhaiteriez un contrôle plus fin de la librairie. Vous pouvez effectuer autant de transaction que vous le souhaitez avec `pfpro_process` lors d'une connexion.

7.80.3 Installation

Ces fonctions ne sont disponible que si PHP a été compilé avec l'option `--with-pfpro[=DIR]`.

7.80.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>pfpro.defaulthost/PFPRO_VERSION < 3</code>	<code>"test.signio.com"</code>	<code>PHP_INI_ALL</code>
<code>pfpro.defaulthost</code>	<code>"test-payflow.verisign.com"</code>	<code>PHP_INI_ALL</code>
<code>pfpro.defaultport</code>	<code>"443"</code>	<code>PHP_INI_ALL</code>
<code>pfpro.defaulttimeout</code>	<code>"30"</code>	<code>PHP_INI_ALL</code>
<code>pfpro.proxyaddress</code>	<code>" "</code>	<code>PHP_INI_ALL</code>
<code>pfpro.proxyport</code>	<code>" "</code>	<code>PHP_INI_ALL</code>

<code>pfpro.proxylogon</code>	<code>""</code>	<code>PHP_INI_ALL</code>
<code>pfpro.proxypassword</code>	<code>""</code>	<code>PHP_INI_ALL</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

7.80.5 Types de ressources

Cette extension ne définit aucune ressource.

7.80.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [pfpro_cleanup](#)
- [pfpro_init](#)
- [pfpro_process_raw](#)
- [pfpro_process](#)
- [pfpro_version](#)

7.80.8 pfpro_init() : Initialise la librairie Payflow Pro

```
void pfpro_init ( void )
```

[pfpro_init](#) initialise la librairie Payflow Pro library. Vous pouvez omettre cet appel : dans ce cas, elle sera appelée automatiquement [pfpro_init](#) avant la première transaction.

Voir aussi [pfpro_cleanup](#).

7.80.9 pfpro_process_raw() : Envoie une transaction brute à Payflow Pro

```
string pfpro_process_raw ( string parameters, string address, int port, int timeout, string proxy_address, int proxy_port, string proxy_logon, string proxy_password )
```

[pfpro_process_raw](#) retourne une chaîne avec une réponse.

[pfpro_process_raw](#) envoie une transaction brute au serveur Payflow Pro. Il est vivement recommandé d'utiliser [pfpro_process](#) à la place, car les règles de codage sont non standard.

Le premier argument est une chaîne contenant la transaction brute. Tous les autres paramètres sont les mêmes que ceux de [pfpro_process](#). La valeur de retour est une chaîne contenant la réponse brute.

Note
Lisez attentivement le "Payflow Pro Developers Guide" pour connaître tous les détails des paramètres et leur règle d'encodage. Il est recommandé d'utiliser plutôt pfpro_process .

Exemple avec `pfpro_process_raw`

```
<?php
pfpro_init();
$response = pfpro_process("USER=mylogin&PWD[5]=m&ndy&TRXTYPE=S&TENDER=C&AMT=1.50&ACCT=4111111111111111&EXPDATE=0904");
if (!$response) {
    die("Impossible de contacter Verisign.\n");
}
echo "La réponse brute de Verisign est ".$response;
pfpro_cleanup();
?>
```

7.80.10 pfpro_process() : Effectue une transaction avec Payflow Pro

array **pfpro_process** (array parameters , string address , int port , int timeout , string proxy_address , int proxy_port , string proxy_logon , string proxy_password)

`pfpro_process` retourne un tableau associatif, contenant la réponse à la transaction.

`pfpro_process` effectue une transaction avec Payflow Pro. Le premier paramètre 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 un événement DNS ou un problème de 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` indique le nom de compte et le mot de passe à utiliser sur le proxy.

`pfpro_process` retourne un tableau associatif.

Note

Lisez attentivement le "Payflow Pro Developers Guide" pour connaître les détails des autres paramètres.

Exemple de transaction Payflow Pro

```
<?php
pfpro_init();
$transaction = array(USER => 'monlogin',
    PWD => 'mmotdepasse',
    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];
echo ", c'est-à-dire : ".$response[RESPMSG]."\n";
echo "\nLa requête de transaction: ";
print_r($transaction);
```

```
echo "\nLa réponse: ";
print_r($response);
pfpro_cleanup();
?>
```

7.80.11 pfpro_version() : Lit le numéro de version de Payflow Pro

string **pfpro_version** (void)

pfpro_version lit la version de la librairie Payflow Pro. Au moment de rédaction de la doc, c'était L211.

7.81 Options PHP et informations

7.81.1 Introduction

Ces fonctions vous donnent accès à de nombreuses informations sur PHP lui-même, comme par exemple les configurations d'exécution, les extensions chargées, les versions, etc... Vous trouverez aussi des fonctions pour modifier des options. Et vous trouverez aussi la star des fonctions PHP [phpinfo](#) .

7.81.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.81.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.81.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<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>magic_quotes_gpc</code>	"1"	PHP_INI_PERDIR PHP_INI_SYSTEM
<code>magic_quotes_runtime</code>	"0"	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#) .

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 PHP. Vous

pouvez activer le chargement dynamique d'extension (avec la fonction [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` integer

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 affectés 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`.

`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. Si `magic_quotes_sybase` fonctionne aussi, les guillemets simples seront échappés avec un autre guillemet simple, plutôt qu'un antislash.

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.

7.81.5 Types de ressources

Cette extension ne définit aucune ressource.

7.81.6 Constantes prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

Constant	Description
CREDITS_ALL	Tous les crédits. C'est l'équivalent de : CREDITS_DOCS + CREDITS_GENERAL + CREDITS_GROUP + CREDITS_MODULES + CREDITS_FULLPAGE. Elle génère une page HTML complète et autonome.
CREDITS_DOCS	Les crédits de l'équipe de documentation
CREDITS_FULLPAGE	Généralement utilisé combiné avec d'autres options. Cette option indique qu'une page HTML complète doit être générée.
CREDITS_GENERAL	Crédits généraux. Design du langage, concepts, auteurs de PHP 4.0 et 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 API de serveurs, et leurs auteurs.

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 4. Voir aussi phpcredits .
INFO_CONFIGURATION	4	Valeurs locales et serveurs des directives PHP. Voyez aussi ini_get .
INFO_MODULES	8	Les modules chargés et leur configuration respective.
INFO_ENVIRONMENT	16	Les variables d'environnement, qui sont aussi disponibles dans <code>\$ENV</code> .
INFO_VARIABLES	32	Toutes les <u>variables prédéfinies</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 prédéfinies](#)
- [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](#)

7.81.8 assert() : Vérifie si une assertion est fausse

int **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 par exemple 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 `.ini-settings`.

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 appelés 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 (si 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

```
<?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'<br>
    Line '$line'<br>
    Code '$code'<br><hr>";
}
```

```
// Configuration de la méthode de callback
assert_options (ASSERT_CALLBACK, 'mon_assert_handler');

// Utilisation d'une assertion qui va échouer
assert ('mysql_query ("")');
?>
```

7.81.9 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 désactivé (soit en inactivant 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 librairie n'a pu être trouvée, `dl` retournera `FALSE` et émettra un message d'alerte.

Utilisez la fonction `extension_loaded` pour vérifier qu'une extension est chargée ou non. Cette fonction travaille aussi bien avec les extensions intégrées qu'avec les extensions dynamiquement chargées (via le `php.ini`, ou via `dl`).

Exemple avec `dl`

```
<?php
if (!extension_loaded('gd')) {
    if (!dl('gd.so')) {
        exit;
    }
}
```

Le dossier où sont rangées les extensions dépend de votre plate forme :

Windows – Si il n'est pas explicitement indiqué dans le fichier `php.ini`, le dossier des extensions est `c:\php4\extensions\`.

Unix – Si 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`
- Si PHP a été compilé avec le support (expérimental) de ZTS (Zend Thread Safety)
- 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, comme 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** supporté 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** !

Note

`dl` est sensible à la casse sur les plate-formes Unix.

Voir aussi [directives de chargement d'extensions](#) et [extension_loaded](#).

7.81.10 `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.

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 lister toutes les extensions disponibles :

```
$ php -m
[PHP Modules]
xml
tokenizer
standard
sockets
session
posix
pcrc
overload
mysql
mbstring
ctype

[Zend Modules]
```

Note

`extension_loaded` utilisent 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 des extensions 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`.

7.81.11 `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.

7.81.12 `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](#) , [getmypid](#) , [getmyinode](#) et [getlastmod](#) .

7.81.13 `get_defined_constants()` : Retourne la liste des constantes et leur valeur

array `get_defined_constants` (void)

`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());
?>
```

affichera

```
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
)
```

Voir aussi [get_loaded_extensions](#) , [get_defined_functions](#) et [get_defined_vars](#) .

7.81.14 `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` .

Par exemple, les lignes suivantes :

Exemple avec `get_extension_funcs`

```
<?php
print_r(get_extension_funcs("xml"));
print_r(get_extension_funcs("gd"));
?>
```

vont afficher la liste des fonctions disponibles avec les modules `xml` et `gd` .

Voir aussi [get_loaded_extensions](#) .

7.81.15 `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
print get_include_path();

// Fonctionne sur toutes les versions
print ini_get('include_path');
?>
```

Voir aussi [ini_get](#) , [restore_include_path](#) , [set_include_path](#) et [include](#) .

7.81.16 `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](#) ou [include_once](#) .

L'exemple ci-dessous

Affichage des fichiers inclus et requis

```
<?php
require_once("local.php");
require_once("../inc/global.php");
for ($i=1; $i<5; $i++)
    include "util".$i.".php";
echo "Fichiers appelés avec required_once/included_once\n";
print_r(get_required_files());
?>
```

va afficher :

```
Fichiers appelés avec required_once/included_once
Array
(
    [0] => local.php
    [1] => /full/path/to/inc/global.php
    [2] => util1.php
    [3] => util2.php
    [4] => util3.php
    [5] => util4.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 cette fonction n'était pas un alias de [get_included_files](#)

Voir aussi [include](#) , [include_once](#) , [require](#) , [require_once](#) , et [get_required_files](#) .

7.81.17 [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'interpréteur PHP courant.

Par exemple, la ligne ci-dessous

Exemple avec [get_loaded_extensions](#)

```
<?php
print_r(get_loaded_extensions());
?>
```

affichera la liste suivante :

```
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](#) .

7.81.18 [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 [magic_quotes_gpc](#) ne peut être configuré durant l'exécution du script.

Exemple avec [get_magic_quotes_gpc](#)

```
<?php
echo get_magic_quotes_gpc();           // 1
echo $_POST['lastname'];               // 0'reilly
echo addslashes($_POST['lastname']);  // 0\\'reilly

if (!get_magic_quotes_gpc()) {
    $lastname = addslashes($_POST['lastname']);
} else {
    $lastname = $_POST['lastname'];
}

echo $lastname; // 0'reilly
$sql = "INSERT INTO lastnames (lastname) VALUES ('$lastname')";
?>
```

Voir aussi [addslashes](#) , [stripslashes](#) , [get_magic_quotes_runtime](#) et [ini_get](#) .

7.81.19 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](#) .

7.81.20 get_required_files() : Alias de [get_included_files](#)

[get_required_files](#) est un alias de [get_included_files](#) .

7.81.21 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
$ip = getenv("REMOTE_ADDR");
// retourne l'adresse IP de l'utilisateur
?>
```

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](#) .

7.81.22 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 marqueur de temps UNIX, utilisable comme paramètre avec la fonction date. getlastmod retourne ^{FALSE} en cas d'erreur.

Exemple avec getlastmod

```
<?php
// affiche 'Dernière modification: March 04 1998 20:43:59.'
echo "Dernière modification : ".date( "F d Y H:i:s.", getlastmod() );
?>
```

Voir aussi date, getmyuid, getmygid, get_current_user, getmyinode, getmypid et filemtime.

7.81.23 getmygid() : Retourne le GID du propriétaire du script

int **getmygid** (void)

getmygid retourne le GID du propriétaire du script, ou ^{FALSE} en cas d'erreur.

Voir aussi getmyuid, getmypid, get_current_user, getmyinode et getlastmod.

7.81.24 getmyinode() : Retourne l'inode du script

int **getmyinode** (void)

getmyinode retourne l'inode du script, ou ^{FALSE} en cas d'erreur.

Voir aussi getmyuid, getmygid, get_current_user, getmypid et getlastmod.

Note

Cette fonction n'est pas implémentée sous Windows.

7.81.25 getmypid() : Retourne le numéro de processus courant

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 getmyuid, getmygid, get_current_user, getmyinode et getlastmod.

7.81.26 getmyuid() : Retourne l'UID du propriétaire du script actuel

int **getmyuid** (void)

getmyuid retourne l'UID du propriétaire du script actuel ou ^{FALSE} en cas d'erreur.

Voir aussi getmypid, get_current_user, getmyinode et getlastmod.

7.81.27 getopt() : Lit des options passés dans la ligne de commande

string **getopt** (string options)

getopt retourne un tableau associatif avec la liste des paire option/valeur, lues dans la ligne de commande, et basée sur le format d'option ^{options} ou bien ^{FALSE} en cas d'erreur.

Exemple avec getopt

```
$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

getopt requiert la directive register_argc_argv activée.

Note

Cette fonction n'est pas implémentée sous Windows.

7.81.28 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 is 1", getrusage sera appelé avec le paramètre RUSAGE_CHILDREN.

Toutes les valeurs du tableau sont accessibles en utilisant leur nom dans le tableau.

Exemple 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_etime.tv_sec"]; // Temps utilisateur (en secondes)
echo $dat["ru_etime.tv_usec"]; // Temps utilisateur (en microsecondes)
?>
```

Consultez le manuel "man" pour plus de détails.

Note

Cette fonction n'est pas implémentée sous Windows.

7.81.29 ini_alter() : Alias de ini_set

Note

ini_alter est un alias de ini_set.

7.81.30 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 changement 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 champs de bits ^{access}.

Exemple avec ini_get_all

```
<?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](#)

7.81.31 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}.

Note

Lecture de valeurs booléennes

Une directive de configuration ayant la valeur de ^{off} sera retournée sous la forme d'une chaîne vide, tandis qu'une directive de configuration ayant une valeur de ^{on} sera retournée sous la forme de la chaîne "1".

Note

Lors de la lecture des tailles de mémoire

Plusieurs directives traitant de taille mémoire, comme `upload_max_filesize` sont stockées dans le fichier `php.ini` avec une notation courte. `ini_get` 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.

Exemple avec `ini_get`

```
<?php
/*
Notre fichier php.ini contient les directives suivantes :

display_errors = On
register_globals = Off
post_max_size = 8M
*/

print 'display_errors = ' . ini_get('display_errors') . "\n";
print 'register_globals = ' . ini_get('register_globals') . "\n";
print 'post_max_size = ' . ini_get('post_max_size') . "\n";
print 'post_max_size+1 = ' . (ini_get('post_max_size')+1) . "\n";

/*
Ce script va afficher :

display_errors = 1
register_globals =
post_max_size = 8M
post_max_size+1 = 9
*/
?>
```

Voir aussi `get_cfg_var`, `ini_get_all`, `ini_restore` et `ini_set`.

7.81.32 ini_restore() : Restaure la valeur de l'option de configuration

string **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`.

7.81.33 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 la valeur 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.

Toutes les options disponibles ne peuvent pas être toutes modifiées avec `ini_set`. Ci-dessous, vous trouverez une liste de toutes les options (disponibles en PHP 4.0.5-dev), et si elles peuvent être modifiées.

Nom	Par défaut	Modifiable
com.allow_dcom	"0"	PHP_INI_SYSTEM
com.autoregister_typelib	"0"	PHP_INI_SYSTEM
com.autoregister_verbose	"0"	PHP_INI_SYSTEM
com.autoregister_casesensitive	"1"	PHP_INI_SYSTEM
com.typelib_file	""	PHP_INI_SYSTEM
crack.default_dictionary	NULL	PHP_INI_SYSTEM
exif.encode_unicode	"ISO-8859-15"	PHP_INI_ALL
exif.decode_unicode_motorola	"UCS-2BE"	PHP_INI_ALL

exif.decode_unicode_intel	"UCS-2LE"	PHP_INI_ALL
exif.encode_jis	""	PHP_INI_ALL
exif.decode_jis_motorola	"JIS"	PHP_INI_ALL
exif.decode_jis_intel	"JIS"	PHP_INI_ALL
fbsql.allow_persistent	"1"	PHP_INI_SYSTEM
fbsql.generate_warnings	"0"	PHP_INI_SYSTEM
fbsql.autocommit	"1"	PHP_INI_SYSTEM
fbsql.max_persistent	"-1"	PHP_INI_SYSTEM
fbsql.max_links	"128"	PHP_INI_SYSTEM
fbsql.max_connections	"128"	PHP_INI_SYSTEM
fbsql.max_results	"128"	PHP_INI_SYSTEM
fbsql.batchSize	"1000"	PHP_INI_SYSTEM
fbsql.default_host	NULL	PHP_INI_SYSTEM
fbsql.default_user	"_SYSTEM"	PHP_INI_SYSTEM
fbsql.default_password	""	PHP_INI_SYSTEM
fbsql.default_database	""	PHP_INI_SYSTEM
fbsql.default_database_password	""	PHP_INI_SYSTEM
hwapi.allow_persistent	"0"	PHP_INI_SYSTEM
hyperwave.allow_persistent	"0"	PHP_INI_SYSTEM
hyperwave.default_port	"418"	PHP_INI_ALL
iconv.input_encoding	ICONV_INPUT_ENCODING	PHP_INI_ALL
iconv.output_encoding	ICONV_OUTPUT_ENCODING	PHP_INI_ALL
iconv.internal_encoding	ICONV_INTERNAL_ENCODING	PHP_INI_ALL
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
ingres.allow_persistent	"1"	PHP_INI_SYSTEM
ingres.max_persistent	"-1"	PHP_INI_SYSTEM
ingres.max_links	"-1"	PHP_INI_SYSTEM
ingres.default_database	NULL	PHP_INI_ALL
ingres.default_user	NULL	PHP_INI_ALL
ingres.default_password	NULL	PHP_INI_ALL
ibase.allow_persistent	"1"	PHP_INI_SYSTEM
ibase.max_persistent	"-1"	PHP_INI_SYSTEM
ibase.max_links	"-1"	PHP_INI_SYSTEM
ibase.default_user	NULL	PHP_INI_ALL
ibase.default_password	NULL	PHP_INI_ALL
ibase.timestampformat	"%m/%d/%Y%H:%M:%S"	PHP_INI_ALL
ibase.dateformat	"%m/%d/%Y"	PHP_INI_ALL
ibase.timeformat	"%H:%M:%S"	PHP_INI_ALL
java.class.path	NULL	PHP_INI_ALL
java.home	NULL	PHP_INI_ALL
java.library.path	NULL	PHP_INI_ALL
java.library	JAVAILIB	PHP_INI_ALL
java.library	NULL	PHP_INI_ALL
ldap.max_links	"-1"	PHP_INI_SYSTEM
mbstring.detect_order	NULL	PHP_INI_ALL
mbstring.http_input	NULL	PHP_INI_ALL
mbstring.http_output	NULL	PHP_INI_ALL
mbstring.internal_encoding	NULL	PHP_INI_ALL
mbstring.substitute_character	NULL	PHP_INI_ALL
mbstring.func_overload	"0"	PHP_INI_SYSTEM
mcrypt.algorithms_dir	NULL	PHP_INI_ALL
mcrypt.modes_dir	NULL	PHP_INI_ALL
mime_magic.magicfile	"/usr/share/misc/magic.mime"	PHP_INI_SYSTEM
mssql.allow_persistent	"1"	PHP_INI_SYSTEM
mssql.max_persistent	"-1"	PHP_INI_SYSTEM
mssql.max_links	"-1"	PHP_INI_SYSTEM
mssql.max_procs	"25"	PHP_INI_ALL
mssql.min_error_severity	"10"	PHP_INI_ALL
mssql.min_message_severity	"10"	PHP_INI_ALL
mssql.compatability_mode	"0"	PHP_INI_ALL
mssql.connect_timeout	"5"	PHP_INI_ALL

Manuel PHP 4.3.0 – Version Française

mssql.timeout	"60"	PHP_INI_ALL
mssql.textsize	"-1"	PHP_INI_ALL
mssql.textlimit	"-1"	PHP_INI_ALL
mssql.batchsize	"0"	PHP_INI_ALL
mssql.datetimeconvert	"1"	PHP_INI_ALL
mysql.allow_persistent	"1"	PHP_INI_SYSTEM
mysql.max_persistent	"-1"	PHP_INI_SYSTEM
mysql.max_links	"-1"	PHP_INI_SYSTEM
mysql.default_host	NULL	PHP_INI_ALL
mysql.default_user	NULL	PHP_INI_ALL
mysql.default_password	NULL	PHP_INI_ALL
mysql.default_port	NULL	PHP_INI_ALL
mysql.default_socket	NULL	PHP_INI_ALL
ncurses.value	"42"	PHP_INI_ALL
ncurses.string	"foobar"	PHP_INI_ALL
odbc.allow_persistent	"1"	PHP_INI_SYSTEM
odbc.max_persistent	"-1"	PHP_INI_SYSTEM
odbc.max_links	"-1"	PHP_INI_SYSTEM
odbc.default_db	NULL	PHP_INI_ALL
odbc.default_user	NULL	PHP_INI_ALL
odbc.default_pw	NULL	PHP_INI_ALL
odbc.defaultlrl	"4096"	PHP_INI_ALL
odbc.defaultbinmode	"1"	PHP_INI_ALL
odbc.check_persistent	"1"	PHP_INI_SYSTEM
pfpro.defaulthost	"test.signio.com"	
pfpro.defaulthost	"test-payflow.verisign.com"	
pfpro.defaultport	"443"	PHP_INI_ALL
pfpro.defaulttimeout	"30"	PHP_INI_ALL
pfpro.proxyaddress	""	PHP_INI_ALL
pfpro.proxyport	""	PHP_INI_ALL
pfpro.proxylogon	""	PHP_INI_ALL
pfpro.proxypassword	""	PHP_INI_ALL
pgsql.allow_persistent	"1"	PHP_INI_SYSTEM
pgsql.max_persistent	"-1"	PHP_INI_SYSTEM
pgsql.max_links	"-1"	PHP_INI_SYSTEM
pgsql.auto_reset_persistent	"0"	PHP_INI_SYSTEM
pgsql.ignore_notice	"0"	PHP_INI_ALL
pgsql.log_notice	"0"	PHP_INI_ALL
session.save_path	"/tmp"	PHP_INI_ALL
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_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
session.use_cookies	"1"	PHP_INI_ALL
session.use_only_cookies	"0"	PHP_INI_ALL
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	"1"	PHP_INI_SYSTEM PHP_INI_PERDIR
session.encode_sources	"globals,track"	PHP_INI_ALL
extname.global_value	"42"	PHP_INI_ALL
extname.global_string	"foobar"	PHP_INI_ALL
assert.active	"1"	PHP_INI_ALL
assert.bail	"0"	PHP_INI_ALL
assert.warning	"1"	PHP_INI_ALL
assert.callback	NULL	PHP_INI_ALL
assert.quiet_eval	"0"	PHP_INI_ALL
safe_mode_protected_env_vars	SAFE_MODE_PROTECTED_ENV_VARS	PHP_INI_SYSTEM
safe_mode_allowed_env_vars	SAFE_MODE_ALLOWED_ENV_VARS	PHP_INI_SYSTEM
url_rewriter.tags	"a=href,area=href,frame=src,form=fakeentry"	PHP_INI_ALL
sybct.allow_persistent	"1"	PHP_INI_SYSTEM
sybct.max_persistent	"-1"	PHP_INI_SYSTEM
sybct.max_links	"-1"	PHP_INI_SYSTEM

Manuel PHP 4.3.0 – Version Française

sybct.min_server_severity	"10"	PHP_INI_ALL
sybct.min_client_severity	"10"	PHP_INI_ALL
sybct.hostname	NULL	PHP_INI_ALL
tokenizer.global_value	"42"	PHP_INI_ALL
tokenizer.global_string	"foobar"	PHP_INI_ALL
vpopmail.directory	""	PHP_INI_ALL
zlib.output_compression	"0"	PHP_INI_SYSTEM PHP_INI_PERDIR
zlib.output_compression_level	"-1"	PHP_INI_ALL
define_syslog_variables	"0"	PHP_INI_ALL
highlight.bg	HL_BG_COLOR	PHP_INI_ALL
highlight.comment	HL_COMMENT_COLOR	PHP_INI_ALL
highlight.default	HL_DEFAULT_COLOR	PHP_INI_ALL
highlight.html	HL_HTML_COLOR	PHP_INI_ALL
highlight.keyword	HL_KEYWORD_COLOR	PHP_INI_ALL
highlight.string	HL_STRING_COLOR	PHP_INI_ALL
allow_call_time_pass_reference	"1"	PHP_INI_SYSTEM PHP_INI_PERDIR
asp_tags	"0"	PHP_INI_SYSTEM PHP_INI_PERDIR
display_errors	"1"	PHP_INI_ALL
display_startup_errors	"0"	PHP_INI_ALL
enable_dl	"1"	PHP_INI_SYSTEM
expose_php	"1"	PHP_INI_SYSTEM
html_errors	"1"	PHP_INI_SYSTEM
xmlrpc_errors	"0"	PHP_INI_SYSTEM
xmlrpc_error_number	"0"	PHP_INI_ALL
ignore_user_abort	"0"	PHP_INI_ALL
implicit_flush	"0"	PHP_INI_PERDIR PHP_INI_SYSTEM
log_errors	"0"	PHP_INI_ALL
log_errors_max_len	"1024"	PHP_INI_ALL
ignore_repeated_errors	"0"	PHP_INI_ALL
ignore_repeated_source	"0"	PHP_INI_ALL
magic_quotes_gpc	"1"	PHP_INI_PERDIR PHP_INI_SYSTEM
magic_quotes_runtime	"0"	PHP_INI_ALL
magic_quotes_sybase	"0"	PHP_INI_ALL
output_buffering	"0"	PHP_INI_PERDIR PHP_INI_SYSTEM
output_handler	NULL	PHP_INI_PERDIR PHP_INI_SYSTEM
register_argc_argv	"1"	PHP_INI_PERDIR PHP_INI_SYSTEM
register_globals	"0"	PHP_INI_PERDIR PHP_INI_SYSTEM
safe_mode	"1"	PHP_INI_SYSTEM
safe_mode	"0"	PHP_INI_SYSTEM
safe_mode_include_dir	NULL	PHP_INI_SYSTEM
safe_mode_gid	"0"	PHP_INI_SYSTEM
short_open_tag	DEFAULT_SHORT_OPEN_TAG	PHP_INI_SYSTEM PHP_INI_PERDIR
sql.safe_mode	"0"	PHP_INI_SYSTEM
track_errors	"0"	PHP_INI_ALL
y2k_compliance	"0"	PHP_INI_ALL
unserialize_callback_func	NULL	PHP_INI_ALL
arg_separator.output	"&"	PHP_INI_ALL
arg_separator.input	"&"	PHP_INI_SYSTEM PHP_INI_PERDIR
auto_append_file	NULL	PHP_INI_SYSTEM PHP_INI_PERDIR
auto_prepend_file	NULL	PHP_INI_SYSTEM PHP_INI_PERDIR
doc_root	NULL	PHP_INI_SYSTEM
default_charset	SAPI_DEFAULT_CHARSET	PHP_INI_ALL
default_mimetype	SAPI_DEFAULT_MIMETYPE	PHP_INI_ALL
error_log	NULL	PHP_INI_ALL
extension_dir	PHP_EXTENSION_DIR	PHP_INI_SYSTEM
gpc_order	"GPC"	PHP_INI_ALL
include_path	PHP_INCLUDE_PATH	PHP_INI_ALL
max_execution_time	"30"	PHP_INI_ALL
open_basedir	NULL	PHP_INI_SYSTEM
safe_mode_exec_dir	"1"	PHP_INI_SYSTEM
upload_max_filesize	"2M"	PHP_INI_SYSTEM
file_uploads	"1"	PHP_INI_SYSTEM
post_max_size	"8M"	PHP_INI_SYSTEM
upload_tmp_dir	NULL	PHP_INI_SYSTEM
user_dir	NULL	PHP_INI_SYSTEM
variables_order	NULL	PHP_INI_ALL
error_append_string	NULL	PHP_INI_ALL
error_prepend_string	NULL	PHP_INI_ALL
SMTP	"localhost"	PHP_INI_ALL
smtp_port	25	PHP_INI_ALL
browscap	NULL	PHP_INI_SYSTEM

error_reporting	NULL	PHP_INI_ALL
memory_limit	"8M"	PHP_INI_ALL
precision	"14"	PHP_INI_ALL
sendmail_from	NULL	PHP_INI_ALL
sendmail_path	DEFAULT_SENDMAIL_PATH	PHP_INI_SYSTEM
disable_functions	""	PHP_INI_SYSTEM
allow_url_fopen	"1"	PHP_INI_ALL
always_populate_raw_post_data	"0"	PHP_INI_ALL
xbithack	"0"	PHP_INI_ALL
engine	"1"	PHP_INI_ALL
last_modified	"0"	PHP_INI_ALL
child_terminate	"0"	PHP_INI_ALL
async_send	"0"	PHP_INI_ALL

Constante	Valeur	Signification
PHP_INI_USER	1	La valeur peut être modifiée dans un script
PHP_INI_PERDIR	2	La valeur peut être modifiée dans le fichier .htaccess et les directives d'un vhost
PHP_INI_SYSTEM	4	La valeur peut être modifiée dans php.ini ou httpd.conf (mais pas dans les blocs vhost de httpd.conf)
PHP_INI_ALL	7	La valeur peut être modifiée n'importe où

Voir aussi [get_cfg_var](#) , [ini_get_all](#) , [ini_get](#) et [ini_restore](#)

7.81.34 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 fonctionnalité était de fournir un lien vers la page de manuel de ^{PHP} lorsque la directive `html_errors` était activée (par défaut).

Certains messages d'erreur faisaient référence à une page de manuel pour la fonction [main](#) alors que cette page n'existe pas.

Nom de la fonction	Ne pointe plus sur cette page depuis la version
include	4.3.1
include_once	4.3.1
require	4.3.1
require_once	4.3.1

Voir aussi [html_errors](#) et [display_errors](#) .

7.81.35 `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 <code>memory_get_usage</code>
<pre> <?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); </pre>

```
echo memory_get_usage() . "\n"; // 36744
?>
```

Voir aussi [memory_limit](#) .

7.81.36 `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 nom 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, et 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 simple de liste retournée par [php_ini_scanned_files](#)

```
<?php
if ($filelist = php_ini_scanned_files()) {
    if (strlen($filelist) > 0) {
        $files = explode(',', $filelist);
        foreach ($files as $file) {
            echo "<li>" . trim($file) . "</li>\n";
        }
    }
}
?>
```

Voir aussi [ini_set](#) et [phpinfo](#) .

7.81.37 `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.

Note

Cette fonctionnalité a été ajoutée dans PHP 4 Beta 4.

Exemple avec [php_logo_guid](#)

```
<?php
echo '';
?>
```

Voir aussi [phpinfo](#) , [phpversion](#) , [phpcredits](#) et [zend_logo_guid](#) .

7.81.38 `php_sapi_name()` : Retourne le type d'interface utilisé 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 utilisé en 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
$inter_type = php_sapi_name();
if ($inter_type == "cgi")
    print "Vous utilisez CGI PHP\n";
else
    print "Vous n'utilisez pas CGI PHP\n";
?>
```

7.81.39 `php_uname()` : Retourne les informations sur le système d'exploitation

string `php_uname` (void)

`php_uname` retourne une chaîne de caractères avec des informations 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` .

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 prédéfinies liées, qui peuvent s'avérer utiles, par exemple :

Exemples avec quelques constantes liées au système

```
<?php
// *nix
echo DIRECTORY_SEPARATOR; // :
echo PHP_SHLIB_SUFFIX;     // so
```

```
// Win*
echo DIRECTORY_SEPARATOR; // ;
echo PHP_SHLIB_SUFFIX;    // dll
?>
```

Voir aussi [phpversion](#) , [php_sapi_name](#) et [phpinfo](#) .

7.81.40 `phpcredits()` : Imprime les crédits de ^{PHP}

```
void 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 :

Afficher tous les crédits de PHP

```
<?php
phpcredits(CREDITS_GENERAL);
?>
```

Et pour afficher la liste des développeurs et du groupe de documentation dans une page séparée, vous utiliserez

Afficher les développeurs et le groupe de documentation PHP

```
<?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);
// Un autre 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	Cette constante est définie, mais elle n'est toujours pas utilisée sous PHP 4.0.1p12.

Voir aussi [phpinfo](#) , [phpversion](#) et [php_logo_guid](#) .

7.81.41 phpinfo() : Affiche de nombreuses informations sur le PHP

```
int phpinfo ( int what )
```

`phpinfo` affiche de nombreuses informations sur le PHP, concernant sa configuration courante : options de compilation, extensions, version, informations sur le serveur, et environnement (lorsque compilé comme module), environnement PHP, chemins, utilisateur, en-têtes HTTP, et licence GNU Public License.

Comme tous les systèmes sont configurés différemment, `phpinfo` sert généralement à vérifier la configuration ainsi que les variables prédéfinies, pour une plate-forme donnée. De plus, `phpinfo` est un 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 les additionner.

Nom (constant)	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 4. Voir aussi <code>phpcredits</code> .
INFO_CONFIGURATION	4	Valeurs courantes locales et générales des directives PHP. Voyez aussi la fonction <code>ini_get</code> .
INFO_MODULES	8	Modules chargées et leur configuration spécifique.
INFO_ENVIRONMENT	16	Informations sur les environnements de variables, qui sont disponibles dans la variable <code>\$_ENV</code> .
INFO_VARIABLES	32	Affiche toutes les <u>variables prédéfinies</u> , issues de l'environnement, la méthode GET, la méthode POST, les cookies et le serveur.
INFO_LICENSE	64	La licence PHP. Voir aussi <u>la FAQ de la licence</u> .
INFO_ALL	-1	Affiche toutes les informations sus-citées. C'est la valeur par défaut.

Exemple avec `phpinfo`

```
<?php
// Affiche toutes les information, 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 inclut les logos PHP et Zend, ainsi que les crédits.

Voir aussi `get_loaded_modules`, `phpversion`, `phpcredits`, `php_logo_guid`, `ini_get`, `ini_set` et la section sur les variables prédéfinies.

7.81.42 phpversion() : Retourne le numéro de la version courante de PHP

```
string phpversion ( void )
```

`phpversion` retourne le numéro de la version courante de PHP.

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 "PHP Version : ".phpversion();
?>
```

Voir aussi [version_compare](#), [phpinfo](#), [phpcredits](#), [php_logo_guid](#) et [zend_version](#).

7.81.43 putenv() : Fixe la valeur d'une variable d'environnement

void **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 qui 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.

Modification d'une variable d'environnement

```
<?php
putenv("UNIQID=$unqid");
?>
```

Voir aussi [getenv](#).

7.81.44 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

print get_include_path(); // ../usr/local/lib/php

set_include_path('/inc');

print get_include_path(); // /inc

// Fonctionne depuis PHP 4.3.0
restore_include_path();

// Fonctionne sur toutes les versions
ini_restore('include_path');

print get_include_path(); // ../usr/local/lib/php
```

```
?>
```

Voir aussi [ini_restore](#) , [set_include_path](#) , [get_include_path](#) et [include](#) .

7.81.45 [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. [set_include_path](#) 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');
?>
```

Voir aussi [ini_set](#) , [get_include_path](#) , [restore_include_path](#) et [include](#) .

7.81.46 [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](#) .

7.81.47 [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.

Lorsqu'elle est appelée, la fonction [set_time_limit](#) remet le compteur de 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.

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` .

7.81.48 version_compare() : Compare deux versions de PHP

int **version_compare** (string version1 , string version2 , string operator)

version_compare compare les deux versions de PHP standardisée. 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.

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, version_compare retournera 1 si la relation est vérifiée et 0 sinon.

Exemple avec version_compare

```
<?php
echo version_compare("4.0.4", "4.0.6");
// affiche -1
echo version_compare("4.0.4", "4.0.6", "<");
echo version_compare("4.0.6", "4.0.6", "eq");
// affichent tous 1
?>
```

7.81.49 zend_logo_guid() : Retourne le logo de Zend

string **zend_logo_guid** (void)

Note

Cette fonctionnalité a été ajoutée en PHP 4 Beta 4.

7.81.50 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, pour l'exécutable PHP courant.

Exemple avec zend_version

```
<?php
// affiche e.g. 'Version du moteur Zend: 1.0.4'
// 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](#) .

7.82 POSIX

7.82.1 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. Certains 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 par exemple `posix_getpwnam` et compagnie. Aucune des fonctions POSIX n'est surveillée lorsque le safe mode est activé. Il est donc **fortement** recommandé de désactiver les fonctions POSIX (à l'aide de l'option de configuration `--disable-posix`) si vous travaillez dans un tel environnement.

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.82.2 Installation

Le support des fonctions POSIX est activé par défaut. Vous pouvez désactiver ces fonctions en compilant ^{PHP} avec l'option `--disable-posix`.

7.82.3 Voir aussi

La section sur la contrôle des processus peut aussi vous intéresser.

- [Introduction](#)
- [Installation](#)
- [Voir aussi](#)
- [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_setegid](#)
- [posix_seteuid](#)
- [posix_setgid](#)
- [posix_setpgid](#)
- [posix_setsid](#)
- [posix_setuid](#)
- [posix_strerror](#)
- [posix_times](#)
- [posix_ttyname](#)
- [posix_uname](#)

7.82.5 [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](#) .

7.82.6 [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.

7.82.7 [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.

7.82.8 [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.

7.82.9 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.

7.82.10 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_getgid();
$groupinfo = posix_getgrgid($groupid);
print_r($groupinfo);
/* Exemple d'affichage :
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.

7.82.11 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.

7.82.12 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.

7.82.13 `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.

7.82.14 `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`.

7.82.15 `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.

7.82.16 `posix_getpid()` : Retourne l'identifiant du processus courant

`int posix_getpid ( void )`

[`posix\_getpid`](#) retourne l'identifiant du processus courant.

7.82.17 `posix_getppid()` : Retourne l'identifiant du processus parent

`int posix_getppid ( void )`

[`posix\_getppid`](#) retourne l'identifiant du processus parent du processus courant.

7.82.18 `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 :

Élément	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	GEOS 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.

7.82.19 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 :

Élément	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 passes "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	GEOS 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 professionnelle, 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.

7.82.20 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.

7.82.21 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`.

7.82.22 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.

7.82.23 `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.

7.82.24 `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.

7.82.25 `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 fichier, 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)`.

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.

7.82.26 `posix_setegid()` : Modifie le GID réel du processus courant

```
bool posix_setegid ( int gid )
```

`posix_setegid` modifie l'identifiant de group 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.

`posix_setegid` retourne `TRUE` en cas de succès, et `FALSE` sinon.

7.82.27 posix_seteuid() : Set the effective UID of the current process

```
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.

posix_seteuid retourne `TRUE` en cas de succès, et `FALSE` sinon.

7.82.28 posix_setgid() : Fixe le GID effective du processus courant.

```
bool posix_setgid ( int gid )
```

posix_setgid fixe le GID effective 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.

posix_setgid retourne `TRUE` en cas de succès, `FALSE` sinon.

7.82.29 posix_setpgid() : Fixe l'identifiant de group de processus

```
int posix_setpgid ( int pid , int pgid )
```

posix_setpgid ajoute le processus `pid` au groupe d'id `pgid`. Reportez-vous à POSIX.1 et `setpgid(2)` dans le manuel de votre système POSIX pour plus d'informations sur le contrôle de tâche. posix_setpgid retourne `TRUE` en cas de succès, et `FALSE` sinon.

7.82.30 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.

7.82.31 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 faire ceci.

posix_setuid retourne `TRUE` en cas de succès, `FALSE` sinon. Voir aussi posix_setgid.

7.82.32 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 avec cette fonction pour lire le dernier message d'erreur POSIX.

Voir aussi `posix_get_last_error`.

7.82.33 `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.

7.82.34 `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.

7.82.35 `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 (e.g. 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)

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 3 chiffres pour la version, par exemple.

7.83 PostgreSQL

7.83.1 Introduction

La base de données PostgreSQL est un produit Open Source, disponible sans frais. Postgres, développé au département de Science informatique, à UC Berkeley, mis 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 originale de Berkeley.

7.83.2 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 <http://techdocs.postgresql.org/>.

7.83.3 Installation

Afin d'activer le support PostgreSQL, l'option `--with-pgsql[=DIR]` est nécessaire lors de la compilation de PHP. DIR est le path du dossier d'installation de PostgreSQL, et par défaut il vaut `/usr/local/pgsql`. Si le module shared object est disponible, le module PostgreSQL est chargeable avec la directive du fichier `php.ini` ou via la fonction `dl`.

7.83.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<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
<code>pgsql.ignore_notice</code>	"0"	PHP_INI_ALL
<code>pgsql.log_notice</code>	"0"	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`pgsql.allow_persistent` bool

Whether to allow persistent Postgres connections.

`pgsql.max_persistent` integer

Le nombre maximum de connexions persistantes Postgre par processus.

`pgsql.max_links` integer

Le nombre maximum de connexions Postgre par processus, y compris les connexions persistantes.

7.83.5 Trucs et astuces d'utilisation

Attention

Utiliser le module PostgreSQL avec PHP 4.0.6 n'est pas recommandé à cause d'un bug dans le gestionnaire d'alerte. Utilisez plutôt la version 4.1.0 ou plus récent.

Attention

Le nom des fonctions PostgreSQL va changer dans la version 4.2.0 pour prendre en compte les standards actuels de programmation. La plupart des nouveaux noms recevront des soulignés, comme par exemple `pg_lo_open()`. Certaines fonctions sont renommées différemment, comme par exemple `pg_exec()` en `pg_query()`. Les anciens noms sont toujours utilisés pour encore quelques versions, mais ils seront bientôt supprimés définitivement.

Ancien nom	Nouveau nom
<code>pg_cmdtuples</code>	<code>pg_affected_rows</code>
<code>pg_errormessage</code>	<code>pg_last_error</code>
<code>pg_exec</code>	<code>pg_query</code>
<code>pg_fieldname</code>	<code>pg_field_name</code>
<code>pg_fieldsize</code>	<code>pg_field_size</code>
<code>pg_fieldnum</code>	<code>pg_field_num</code>
<code>pg_fielddortlen</code>	<code>pg_field_ortlen</code>
<code>pg_fieldisnull</code>	<code>pg_field_is_null</code>
<code>pg_freeresult</code>	<code>pg_free_result</code>
<code>pg_getlastoid</code>	<code>pg_last_oid</code>
<code>pg_loreadall</code>	<code>pg_lo_read_all</code>
<code>pg_locreate</code>	<code>pg_lo_create</code>
<code>pg_lounlink</code>	<code>pg_lo_unlink</code>
<code>pg_loopen</code>	<code>pg_lo_open</code>
<code>pg_loclose</code>	<code>pg_lo_close</code>
<code>pg_loread</code>	<code>pg_lo_read</code>
<code>pg_lowrite</code>	<code>pg_lo_write</code>
<code>pg_loimport</code>	<code>pg_lo_import</code>
<code>pg_loexport</code>	<code>pg_lo_export</code>
<code>pg_numrows</code>	<code>pg_num_rows</code>
<code>pg_numfields</code>	<code>pg_num_fields</code>
<code>pg_result</code>	<code>pg_fetch_result</code>

Les anciennes syntaxes de `pg_connect` / `pg_pconnect` seront rendues obsolètes pour supporter les nouvelles connexions asynchrones. Utilisez la chaîne de connexion avec `pg_connect` et `pg_pconnect` .

Toutes les fonctions ne sont pas supportées par toutes les compilations. Cela dépend de votre librairie libpq (la librairie C client de PostgreSQL), et comment libpq est compilé. Si il y a des fonctions qui manquent, libpq ne supporte pas la fonctionnalité sur laquelle reposait la fonction n'est pas disponible.

Il est aussi important que vous utilisiez une version de libpq qui soit plus récente que le serveur sur lequel vous vous connectez. Si vous utilisez une version de libpq plus ancienne que le serveur, vous aurez des problèmes.

Depuis la version 6.3 (03/02/1998) PostgreSQL utilise les sockets UNIX, et une table est dédiée à ces nouvelles capacités. La socket est située dans le dossier `/tmp/.s.PGSQL.5432` . Cette option peut être activée avec `-i` passé au `postmaster` et cela s'interprète: "écoute sur les sockets TCP/IP et sur les sockets Unix".

Postmaster	PHP	Statut
<code>postmaster &</code>	<code>pg_connect("dbname=MonDbName");</code>	OK
<code>postmaster -i &</code>	<code>pg_connect("dbname=MonDbName");</code>	OK
<code>postmaster &</code>	<code>pg_connect("host=localhost dbname=MonDbName");</code>	Unable to connect to PostgreSQL server: connectDB() failed: Impossible de se connecter au serveur PostgreSQL: connectDB() a échoué. Est ce que le postmaster fonctionne, et accepte les TCP/IP (option -i) sur le port '5432'?

postmaster -i &	pg_connect("host=localhost dbname=MonDbName");	OK

Il est possible de se connecter avec la commande suivante : `$conn = pg_Connect("host=monHote port=monPort
tty=monTTY options=myOptions dbname=myDB user=myUser password=myPassword");`

L'ancienne syntaxe : `$conn = pg_connect("host", "port", "options", "tty", "dbname")` est obsolète.

Les variables environnementales affectent le comportement de PostgreSQL. Par exemple, le module PostgreSQL va rechercher PGHOST dans les variables d'environnement, si le nom du serveur hôte est omis dans la chaîne de connexion. Les variables d'environnement supportées sont différentes suivant les versions. Reportez vous au manuel du programmeur PostgreSQL (libpq – Environment Variables) pour plus de détails.

Assurez vous que vous avez bien configuré vos variables d'environnement pour le bon utilisateur. Utilisez `$_ENV` ou `getenv` pour vérifier quelles variables d'environnement sont disponibles pour le processus courant.

Configuration par défaut des paramètres
<pre>PGHOST=pgsql.example.com PGPORT=7890 PGDATABASE=web-system PGUSER=web-user PGPASSWORD=secret PGDATESTYLE=ISO PGTZ=JST PGCLIENTENCODING=EUC-JP export PGHOST PGPORT PGDATABASE PGUSER PGPASSWORD PGDATESTYLE PGTZ PGCLIENTENCODING</pre>

7.83.6 Constantes prédefinies

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)
PGSQL_NUM (entier)
PGSQL_BOTH (entier)
PGSQL_CONNECTION_BAD (entier)
PGSQL_CONNECTION_OK (entier)
PGSQL_SEEK_SET (entier)
PGSQL_SEEK_CUR (entier)
PGSQL_SEEK_END (entier)
PGSQL_ESCAPE_STRING (entier)
PGSQL_ESCAPE_BYTEA (entier)
PGSQL_EMPTY_QUERY (entier)
PGSQL_COMMAND_OK (entier)
PGSQL_TUPLES_OK (entier)
PGSQL_COPY_OUT (entier)
PGSQL_COPY_IN (entier)
PGSQL_BAD_RESPONSE (entier)
PGSQL_NONFATAL_ERROR (entier)
PGSQL_FATAL_ERROR (entier)
```

7.83.7 Exemples

Depuis PostgreSQL 7.1.0, vous pouvez stocker jusqu'à 1Go dans un champ de type text. Dans les anciennes versions, vous étiez limité à la taille maximale d'un bloc (qui, par défaut, valait 8 ko, et au mieux, 32 ko, suivant le choix au moment de la compilation).

Pour utiliser l'interface des grands objets (large object (lo) interface), il est nécessaire de les placer dans un bloc de transaction. Un bloc de transaction commence avec `begin` et si la transaction se termine avec un `commit` et `end`. Si la transaction échoue, elle doit être conclue par un `abort` et `rollback`.

Utilisation des objets de grande taille (Large Objects)
<pre> <?php \$database = pg_connect ("dbname=jacarta"); pg_query (\$database, "begin"); \$oid = pg_lo_create (\$database); echo "\$oid\n"; \$handle = pg_lo_open (\$database, \$oid, "w"); echo "\$handle\n"; pg_lo_write (\$handle, "large object data"); pg_lo_close (\$handle); pg_query (\$database, "commit"); ?> </pre>

Vous ne devez pas fermer la connexion au serveur PostgreSQL avant de fermer l'objet de grande taille.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Trucs et astuces d'utilisation](#)
- [Constantes prédefinies](#)
- [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_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](#)
- [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_pconnect](#)
- [pg_ping](#)
- [pg_port](#)
- [pg_put_line](#)
- [pg_query](#)
- [pg_result_error](#)
- [pg_result_seek](#)
- [pg_result_status](#)
- [pg_select](#)
- [pg_send_query](#)
- [pg_set_client_encoding](#)
- [pg_trace](#)
- [pg_tty](#)
- [pg_unescape_bytea](#)
- [pg_untrace](#)
- [pg_update](#)

7.83.9 [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](#) , sur la connexion `connection` .
Vous ne pouvez pas annuler une requête démarrée avec [pg_query](#) .

Voir aussi [pg_send_query](#) et [pg_connection_busy](#) .

7.83.10 pg_client_encoding() : Lit l'encodage du client

string **pg_client_encoding** (resource connection)

pg_client_encoding retourne l'encodage du client. Elle retourne une des valeurs suivantes : SQL_ASCII, EUC_JP, EUC_CN, EUC_KR, EUC_TW, UNICODE, MULE_INTERNAL, LATINX (X=1...9), KOI8, WIN, ALT, SJIS, BIG5, WIN1250.

Note

Cette fonction requiert PHP-4.0.2 ou plus récent et PostgreSQL-7.0 ou plus récent.

Jadis, pg_client_encoding s'appelait pg_clientencoding().

Voir aussi pg_set_client_encoding .

7.83.11 pg_close() : Termine une connexion PostgreSQL

bool **pg_close** (resource connection)

pg_close ferme la connexion au serveur PostgreSQL associé à connection . Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

Note

Il n'est généralement pas nécessaire de fermer une connexion non persistante, car elles sont automatiquement fermées à la fin d'un script.

pg_close ne ferme pas les connexions persistantes ouvertes avec pg_pconnect .

Exemple avec pg_close

```
<?php
$dbconn = $dbconn = pg_connect ("host=localhost port=5432 dbname=mary")
    or die("Impossible de se connecter");
print ("Connexion à PostgreSQL");
pg_close($dbconn);
?>
```

Si des objets de grande taille ont été ouverts avec cette connexion, ne fermez pas la connexion avant d'avoir refermé les objets.

7.83.12 pg_connect() : Ouvre une connexion PostgreSQL

resource **pg_connect** (string connection_string)

pg_connect retourne un index de connexion en cas de succès, et ^{FALSE} sinon.

pg_connect ouvre une connexion à un serveur PostgreSQL, grâce à la chaîne de connexion connection_string . Les arguments doivent être placés entre guillemets.

Exemples avec pg_connect

```
<?php
$dbconn = pg_connect("dbname=marie");
//connexion à une base de données nommée "marie"
$dbconn2 = pg_connect("host=localhost port=5432 dbname=marie");
//connexion à une base de données nommée "marie" sur l'hôte "localhost" sur le port "5432"
$dbconn3 = pg_connect ("host=sheep port=5432 dbname=marie user=mouton password=baaaa");
```

```
//connexion à une base de données nommée "marie" sur le serveur "mouton" avec
// un nom d'utilisateur et le mot de passe associé
$conn_string = "host=sheep port=5432 dbname=test user=lamb password=bar";
$dbconn4 = pg_connect ($conn_string);
//connexion à une base de données appelée "test" sur l'hôte "sheep" avec le nom d'utilisateur
?>
```

Les arguments disponibles comprennent notamment `connection_string` `includes` `host` , `port` , `tty` , `options` , `dbname` , `user` et `password` .

`pg_connect` retourne un index de connexion qui sera nécessaire aux autres fonctions PostgreSQL. Vous pouvez ouvrir plusieurs connexions simultanées.

Si un deuxième appel à `pg_connect` est fait avec les mêmes arguments, aucune nouvelle connexion ne sera établie, mais la connexion précédente sera retournée.

L'ancienne syntaxe `$conn = pg_connect("host", "port", "options", "tty", "dbname")` est obsolète.

Voir aussi `pg_pconnect` , `pg_close` , `pg_host` , `pg_port` , `pg_tty` , `pg_options` et `pg_dbname` .

7.83.13 pg_connection_busy() : Vérifie si la connexion PostgreSQL est occupée

`bool pg_connection_busy ( resource connection )`

`pg_connection_busy` retourne `TRUE` si la connexion est occupée. Si elle est occupée, une requête a déjà été lancée, et est en cours. Si `pg_get_result` est appelé, elle sera alors bloquée.

Exemple avec `pg_connection_busy`

```
<?php
$dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");
$bs = pg_connection_busy($dbconn);
if ($bs) {
    echo 'La connexion est occupée';
}
else {
    echo 'La connexion est libre';
}
?>
```

Voir aussi `pg_connection_status` et `pg_get_result` .

7.83.14 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. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec `pg_connection_reset`

```
<?php
$dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");
$dbconn2 = pg_connection_reset($dbconn);
if ($dbconn2) {
    echo 'Redémarrage réussi';
}
}
```

```

    else {
        echo 'Redémarrage échoué';
    }
?>

```

Voir aussi [pg_connect](#) , [pg_pconnect](#) et [pg_connection_status](#) .

7.83.15 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` . Les valeurs possibles sont `PGSQL_CONNECTION_OK` (connexion correcte) et `PGSQL_CONNECTION_BAD` (connexion abandonnée). La valeur de retour entière 0 indique une connexion valide.

Exemple avec [pg_connection_status](#)

```

<?php
$dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");
$stat = pg_connection_status($dbconn);
if ($stat === 0) {
    echo 'Connexion ok';
} else {
    echo 'Connexion ko';
}
?>

```

Voir aussi [pg_connection_busy](#) .

7.83.16 pg_convert() : Convertit un tableau associatif 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` doit correspondre aux index du tableau `assoc_array` . [pg_convert](#) retourne un tableau avec les valeurs converties, en cas de succès, et sinon, `FALSE` .

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 [pg_meta_data](#) .

7.83.17 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 la table `table_name` . Cette fonction utilise la

commande SQL interne `COPY FROM` . Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [pg_copy_to](#) .

7.83.18 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, et le retourne. Cette fonction utilise la commande interne SQL `COPY TO` pour insérer les tableaux. En cas d'erreur, la fonction retourne `FALSE` .

Voir aussi [pg_copy_from](#)

7.83.19 pg_dbname() : Nom de la base de données

string **pg_dbname** (resource connection)

[pg_dbname](#) retourne le nom de la base de données PostgreSQL associée à l'index de connexion `connection`, ou `FALSE` si connexion n'est pas valide.

Exemple avec [pg_dbname](#)

```
<?php
    error_reporting(E_ALL);

    pg_connect ("host=localhost port=5432 dbname=mary");
    echo pg_dbname();
?>
```

7.83.20 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` , qui est bâti sous la forme `champ=>valeur` . Si `option` est spécifié, [pg_convert](#) est appliqué à `assoc_array` avec cette option.

Exemple avec [pg_delete](#)

```
<?php
    $db = pg_connect ('dbname=foo');
    // C'est sans probleme, car $_POST est convertit automatiquement
    $res = pg_delete($db, 'post_log', $_POST);
    if ($res) {
        echo "Les données POSTées ont été effacées : $res\n";
    }
    else {
        echo "Les données d'entrées sont erronées.\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.

Voir aussi [pg_convert](#) .

7.83.21 pg_end_copy() : Synchronise avec le serveur PostgreSQL

bool **pg_end_copy** (resource connection)

[pg_end_copy](#) synchronise le client PostgreSQL (ici PHP) avec le serveur, après une opération de copie. Il faut utiliser cette fonction, sous peine de recevoir une erreur "out of sync" (désynchronisé). Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Pour plus de détails et un exemple voyez : [pg_put_line](#) .

7.83.22 pg_escape_bytea() : Protège le caractères d'une chaîne binaire en mode 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 \ (e.g. \032). Les utilisateurs doivent effectuer la conversion en format binaire eux-même.

[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.

Voir aussi [pg_unescape_bytea](#) et [pg_escape_string](#) .

7.83.23 pg_escape_string() : Protège une chaîne de caractères

string **pg_escape_string** (string data)

[pg_escape_string](#) protège la chaîne de caractères `data` . Elle retourne la chaîne ainsi protégée. L'utilisation de cette fonction est recommandée, à la place de [addslashes](#) .

Note

Cette fonction requiert PostgreSQL 7.2 ou plus récent.

Voir aussi [pg_escape_bytea](#) .

7.83.24 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` . Elle retourne `FALSE` s'il n'y a plus de lignes disponibles.

Exemple avec `pg_fetch_all`

```

<?php
$conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

$result = pg_query ($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Une erreur est survenue.\n";
    exit;
}

$arr = pg_fetch_all ($result, 0, PGSQL_NUM);
var_dump($arr);
?>

```

Voir aussi `pg_fetch_row` , `pg_fetch_array` , `pg_fetch_object` et `pg_fetch_result` .

7.83.25 pg_fetch_array() : Lit une ligne de résultat PGSQL dans un tableau

array **pg_fetch_array** (resource result , int row , int result_type)

`pg_fetch_array` retourne un tableau qui contient à la ligne demandée, dans le résultat identifiée par `result` , et `FALSE` , s'il ne reste plus de lignes.

`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.

L'argument optionnel `result_type` de `pg_fetch_array` est une constante, qui peut prendre les valeurs suivantes : `PGSQL_ASSOC` , `PGSQL_NUM` et `PGSQL_BOTH` . `pg_fetch_array` retourne un tableau associatif dont les clés sont les noms de colonnes, avec la constante `PGSQL_ASSOC` ; les index de colonnes avec `PGSQL_NUM` , et les deux avec la constante `PGSQL_BOTH` . La valeur par défaut est `PGSQL_BOTH` .

Note

`result_type` a été ajoutée en PHP 4.0.

Il est important de noter que `pg_fetch_array` n'est pas significativement plus lent que `pg_fetch_row` , tandis qu'elle fournit un confort d'utilisation notable.

Pour plus de détails, reportez-vous à `pg_fetch_row` .

Exemple avec `pg_fetch_array`

```

<?php
$conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Erreur de connexion.\n";
    exit;
}

$result = pg_query ($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Erreur durant la requête.\n";
    exit;
}

$arr = pg_fetch_array ($result, 0, PGSQL_NUM);
echo $arr[0] . " <- array\n";

$arr = pg_fetch_array ($result, 1, PGSQL_ASSOC);
echo $arr["author"] . " <- array\n";
?>

```

Voir aussi [pg_fetch_row](#) , [pg_fetch_object](#) et [pg_fetch_result](#) .

Note

Depuis 4.1.0, `row` est devenu optionnel. Appeler [pg_fetch_array](#) avec incrémenter le pointeur interne de 1.

7.83.26 pg_fetch_assoc() : Lit une ligne de résultat PGSQL 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](#) retourne `FALSE` si il n'y a plus de lignes.

[pg_fetch_assoc](#) est une version améliorée de [pg_fetch_row](#) . En plus de stocker les valeurs dans un tableau a index numérique, elle stocke aussi les données dans un tableau associatif, en utilisant les noms de colonnes comme index.

`row` est le numéro de ligne à lire. Par défaut, il vaut 0.

[pg_fetch_assoc](#) n'est pas significativement plus lente que [pg_fetch_row](#) , et elle apporte un confort d'utilisation appréciable.

Exemple avec [pg_fetch_assoc](#)

```
<?php
$conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

$result = pg_query ($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Une erreur est survenue.\n";
    exit;
}

$arr = pg_fetch_assoc ($result, 1, PGSQL_ASSOC);
echo $arr["author"] . " <- array\n";
?>
```

Voir aussi [pg_fetch_row](#) , [pg_fetch_array](#) , [pg_fetch_object](#) et [pg_fetch_result](#) .

7.83.27 pg_fetch_object() : Lit une ligne de résultat PGSQL dans un objet

object **pg_fetch_object** (resource result , int row , int result_type)

[pg_fetch_object](#) retourne un objet dont les membres sont les champs de la ligne demandée, ou `FALSE` , si il n'y a plus de lignes.

[pg_fetch_object](#) est similaire à [pg_fetch_array](#) , avec une différence majeure : c'est un objet qui est retourné, au lieu d'un tableau. Par conséquent, cela signifie que vous ne pouvez accéder aux membres qu'avec leur nom, et non plus leur offset (les nombres ne sont pas autorisés comme nom de membre).

L'argument optionnel `result_type` de `result_type` est une constante qui peut prendre les valeurs suivantes : `PGSQL_ASSOC`, `PGSQL_NUM` et `PGSQL_BOTH`.

Note

`result_type` a été ajouté dans PHP 4.0.

Au niveau vitesse, `pg_fetch_object` est aussi rapide que `pg_fetch_row` et presque aussi rapide que `pg_fetch_row` (la différence est non significative).

Lecture d'un objet Postgres

```
<?php
$dbbase = "verlag";
$db_conn = pg_connect ("host=localhost port=5432 dbname=$dbbase");
if (!$db_conn): ?>
    <H1>Impossible de se connecter à la base PostgreSQL<?php echo $dbbase ?></H1> <?php
    exit;
endif;

$squ = pg_query ($db_conn, "SELECT * FROM verlag ORDER BY autor");
$row = 0; // PostgreSQL requiert un compteur de ligne que d'autres n'ont pas

while ($data = pg_fetch_object ($squ, $row)) {
    echo $data->autor." (";
    echo $data->jahr."): ";
    echo $data->titel."<BR>";
    $row++;
}
?>
<PRE>
<?php
$fields[] = Array ("autor", "Author");
$fields[] = Array ("jahr", " Year");
$fields[] = Array ("titel", " Title");

$row= 0; // PostgreSQL requiert un compteur de ligne que d'autres n'ont pas
while ($data = pg_fetch_object ($squ, $row)) {
    echo "-----\n";
    reset ($fields);
    while (list (,$item) = each ($fields)):
        echo $item[1].": ".$data->$item[0]."\n";
    endwhile;
    $row++;
}
echo "-----\n";
?>
</PRE>
<?php
pg_free_result ($squ);
pg_close ($db_conn);
?>
```

Note

Depuis 4.1.0, `row` est devenu optionnel. Appeler `pg_fetch_array` avec incrémenter le pointeur interne de 1.

Voir aussi `pg_query`, `pg_fetch_array`, `pg_fetch_assoc`, `pg_fetch_row` et `pg_fetch_result`.

7.83.28 pg_fetch_result() : Retourne les valeurs d'un résultat

`mixed pg_fetch_result ( resource result , int row , mixed field )`

`pg_fetch_result` retourne les valeurs stockées dans la ressource de résultat `result`, retournée par `pg_query`. `row` est un entier. `field` est le nom du champ (sous forme de chaîne) ou un index de colonne (sous forme d'entier). `row` et `field` spécifie quelles cellules de la table de résultat doivent être lu. Les lignes sont numérotées à partir de 0. Au lieu de nommer le champ, vous pouvez utiliser le numéro de colonne. L'indexation des colonnes commence à 0.

PostgreSQL dispose de nombreux types, et seuls les types basiques sont supportés ici. Tous les

types ^{integer} sont retournées sous forme de ^{integer}. Tous les types de float, et nombres réels sont retournées sous forme de ^{float}. Les booléens sont retournées comme des "t" ou "f". Tous les autres types, y compris les tableaux, sont retournées sous forme de chaînes formatées, de la même manière que PostgreSQL vous les afficherait dans le client ^{psql}.

7.83.29 pg_fetch_row() : Lit une ligne dans un tableau

array **pg_fetch_row** (resource result , int row)

pg_fetch_row retourne un tableau qui contient les données de la ligne demandée, ou ^{FALSE}, si il ne reste plus de lignes.

pg_fetch_row lit une ligne dans le résultat associé à l'index ^{result}. La ligne est retournée sous la forme d'un tableau. La ligne est retournée sous la forme d'un tableau, qui commence à l'index 0.

Les appels ultérieurs à pg_fetch_row retourneront la ligne d'après, ou bien ^{FALSE}, lorsqu'il n'y aura plus de lignes.

Postgres retourne une ligne
<pre><?php \$conn = pg_pconnect("dbname=publisher"); if (!\$conn) { echo "Une erreur est survenue.\n"; exit; } \$result = pg_exec(\$conn, "SELECT * FROM authors"); if (!\$result) { echo "Une erreur est survenue.\n"; exit; } \$num = pg_numrows(\$result); for (\$i=0; \$i<\$num; \$i++) { \$r = pg_fetch_row(\$result, \$i); for (\$j=0; \$j<count(\$r); \$j++) { echo "\$r[\$j]&nbsp;"; } echo "
"; } ?></pre>

Voir aussi pg_query, pg_fetch_array, pg_fetch_object et pg_fetch_result.

Note
Depuis ^{PHP} 4.1.0, ^{row} est devenu un paramètre optionnel. Appeler <u>pg_fetch_row</u> en l'omettant va incrémenter le pointeur interne de résultat de 1.

7.83.30 pg_field_is_null() : Teste si un champ PostGreSQL est à ^{NULL}

int **pg_field_is_null** (resource result , int row , mixed field)

pg_field_is_null teste si un champ est à ^{NULL}. pg_field_is_null retourne 0 si le champ n'est pas ^{NULL}.

pg_field_is_null retourne 1 si le champ est à ^{NULL}. Le champ peut être identifié avec son nom ou son index numérique (commençant à 0).

Exemple avec <u>pg_field_is_null</u>
<pre><?php \$dbconn = pg_connect("dbname=publisher") or die ("Impossible de se connecter"); \$res = pg_query(\$dbconn, "select * from authors where author = 'Orwell'"); if (\$res) { if (pg_field_is_null(\$res, 0, "annee") == 1) { echo "La valeur du champ \"annee\" est null.\n"; } } </pre>

```

    }
    if (pg_field_is_null($res, 0, "year") == 0) {
        echo "La valeur du champ \"annee\" n'est pas null.\n";
    }
}
?>

```

Note

Auparavant, cette fonction s'appelait `pg_fieldisnull()` .

7.83.31 pg_field_name() : Retourne le nom d'un champ PostgreSQL

string **pg_field_name** (resource result , int field_number)

`pg_field_name` va retourner le nom du champ qui occupe la colonne numéro `field_number` dans le résultat `result` . La numérotation des champs commence à 0.

Lire des informations sur les champs PostgreSQL

```

<?php
$dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");

$res = pg_query($dbconn, "select * from authors where author = 'Orwell'");
$i = pg_num_fields($res);
for ($j = 0; $j < $i; $j++) {
    echo "column $j\n";
    $fieldname = pg_field_name($res, $j);
    echo "Champ : $fieldname\n";
    echo "Taille affichée : ".pg_field_prtlen($res, $fieldname)." characters\n";
    echo "Taille de stockage : ".pg_field_size($res, $j)." bytes\n";
    echo "Type de champ : ".pg_field_type($res, $j)." \n\n";
}
?>

```

The above example would produce the following output:

```

column 0
Champ : author
Taille affichée : 6 characters
Taille de stockage : -1 bytes
Type de champ : varchar

column 1
Champ : year
Taille affichée : 4 characters
Taille de stockage : 2 bytes
Type de champ : int2

column 2
Champ : title
Taille affichée : 24 characters
Taille de stockage : -1 bytes
Type de champ : varchar

```

Note

Auparavant, cette fonction s'appelait `pg_fieldname()` .

Voir aussi [pg_field_num](#) .

7.83.32 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` . La numérotation des champs commence à 0. `pg_field_num` retournera -1 en cas d'erreur.

Note
Auparavant, cette fonction s'appelait <code>pg_fieldnum()</code> .

Voir aussi [pg_field_name](#) .

7.83.33 pg_field_ptrlen() : Retourne la taille imprimée

```
int pg_field_ptrlen ( resource result , int row_number , string field_name )
```

[pg_field_ptrlen](#) retourne la taille imprimée (nombre de caractères) d'une valeur donnée dans un résultat PostgreSQL. La numérotation des lignes commence à 0. [pg_field_ptrlen](#) retourne `-1` en cas d'erreur.

Note
Auparavant, cette fonction s'appelait <code>pg_fieldprtlen()</code> .

Voir aussi [pg_field_size](#) .

7.83.34 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. [pg_field_size](#) retourne `-1` si la taille est variable. [pg_field_size](#) retourne `FALSE` en cas d'erreur. La numérotation des colonnes commence à 0.

Note
Auparavant, cette fonction s'appelait <code>pg_fieldsize()</code> .

Voir aussi [pg_field_ptrlen](#) et [pg_field_type](#) .

7.83.35 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 du champ donné par son index `field_number` . La numérotation des champ commence à 0.

Note
Auparavant, cette fonction s'appelait <code>pg_fielddtype()</code> .

Voir aussi [pg_field_ptrlen](#) et [pg_field_name](#) .

7.83.36 pg_free_result() : Libère la mémoire

```
bool pg_free_result ( resource result )
```

`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. Mais, si vous êtes sûr de ne pas avoir besoin du résultat ultérieurement, vous pouvez appeler `pg_free_result` avec l'index de résultat comme argument, et la mémoire sera libérée. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note

Auparavant, cette fonction s'appelait `pg_freeresult()`.

Voir aussi `pg_query`.

7.83.37 pg_get_notify() : Lit le message de 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`. Si il y a un message de notification avec cette connexion, un tableau sera retourné avec le message et le PID du serveur. Si il n'y a pas de message, `pg_get_notify` retourne `FALSE`.

Exemple avec `pg_get_notify`

```
<?php
$conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

// ecoute le message 'author_updated' des autres processus
pq_query($conn, 'LISTEN author_updated;');
$notify = pg_get_notify($conn);
if (!$notify)
    print("Aucun message\n");
else
    print_r($notify);
?>
```

Voir aussi `pg_get_pid`.

7.83.38 pg_get_pid() : Lit l'identifiant de processus du serveur

int `pg_get_pid` (resource connection)

`pg_get_pid` lit l'identifiant de processus du serveur PostgreSQL. Le PID est pratique pour vérifier si un message de `NOTIFY` a été envoyé par un autre processus ou pas.

Exemple avec `pg_get_pid`

```
<?php
$conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

// PID du serveur. Utilisez alors le PID avec pg_get_notify()
$pid = pg_get_pid($conn);
?>
```

Voir aussi `pg_get_notify`.

7.83.39 pg_get_result() : Lit un résultat asynchrone

resource **pg_get_result** (resource connection)

pg_get_result lit le résultat asynchrone d'une requête exécutée par pg_send_query. pg_send_query peut envoyer plusieurs requêtes à PostgreSQL et pg_get_result sert à lire les résultats un par un. pg_get_result retourne une ressource de résultat. Si il n'y a pas de résultat, pg_get_result retourne `FALSE`.

7.83.40 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.

Voir aussi pg_connect et pg_pconnect.

7.83.41 pg_insert() : Insère un tableau dans une table

bool **pg_insert** (resource connection , string table_name , array assoc_array , int options)

pg_insert insère le tableau `assoc_array` (bâtit sur le modèle `champ=>valeur`) dans la table `table_name`. Si `options` est spécifié, pg_convert est appliqué à `assoc_array` avec cette option. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec. Si `options` est spécifiée, pg_insert s'applique à `assoc_array` avec l'option spécifiée.

Exemple avec pg_insert

```
<?php
$db = pg_connect ('dbname=foo');
// C'est correct, car $_POST est convertit automatiquement
$res = pg_insert($db, 'post_log', $_POST);
if ($res) {
    echo "Les données POSTées ont pu être enregistrée avec succès.\n";
}
else {
    echo "Il y a un problème avec les données.\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.

Voir aussi pg_convert.

7.83.42 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 qui a été enregistré sur la connexion `connection`.

Les messages d'erreur peuvent être écrasé 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_status](#) et [pg_connection_status](#) pour améliorer la gestion des erreurs.

Note
pg_last_error s'appelait <code>pg_errormessage()</code> .

Voir aussi [pg_result_error](#) .

7.83.43 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, spécifiée par `connection` . Le serveur PostgreSQL envoie des notes dans plusieurs cas, comme par exemple si la transaction ne peut être continuée. Avec [pg_last_notice](#) , vous pouvez éviter des requêtes inutiles en vérifiant si des notes liées à votre transaction ont été émises.

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 ancienne de PHP ignore 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 `php.ini` à partir de 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.

Voir aussi [pg_query](#) et [pg_last_error](#) .

7.83.44 pg_last_oid() : Retourne l'identifiant du dernier objet

```
int pg_last_oid ( resource result )
```

[pg_last_oid](#) sert à lire le dernier `oid` assigné à un ligne, qui est stocké dans la ressource `result` , obtenu lors de la dernière commande [pg_query](#) . Il faut que cette commande ait été un INSERT. [pg_last_oid](#) retourne un entier positif si il y a eu un `oid` valide. [pg_last_oid](#) retourne `FALSE` si une erreur est survenue,, ou si la commande utilisée avec [pg_query](#) n'était pas un INSERT, ou que cette commande a échoué.

Le champ OID est devenu optionnel depuis PostgreSQL 7.2. Lorsque le champ OID n'est pas défini dans la table, le programmeur doit utiliser [pg_result_status](#) pour vérifier si la ligne a été correctement insérée.

Note
pg_last_oid s'appelait <code>pg_getlastoid()</code> .

Voir aussi [pg_query](#) et [pg_result_status](#) .

7.83.45 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. ^{fd} 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 <code>pg_loclose()</code> .

Voir aussi pg_lo_open, pg_lo_create et pg_lo_import.

7.83.46 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. ^{conn} doit être une connexion valide avec une base de données PostgreSQL. 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.

Note
Auparavant, cette fonction s'appelait <code>pg_locreate()</code> .

7.83.47 pg_lo_export() : Exporte un objet de grande taille vers un fichier

```
bool pg_lo_export ( int oid , string pathname , resource connection )
```

pg_lo_export exporte un objet de grande taille dans un fichier. ^{oid} est un identifiant d'objet de grande taille qui sera exporté dans le fichier ^{filename}, qui spécifie son chemin. Retourne FALSE si une erreur survient, et TRUE en cas de succès. N'oubliez pas que la manipulation d'un objet de grande taille dans PostgreSQL doit intervenir dans une transaction.

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> .

Voir aussi pg_lo_import.

7.83.48 pg_lo_import() : Importe un objet de grande taille depuis un fichier

```
int pg_lo_import ( resource connection , 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 )
```

`pathname` est le chemin jusqu'à un fichier qui servira de source pour créer un objet de grande taille. `pg_lo_import` retourne `FALSE` en cas d'erreur, et sinon un identifiant d'objet, créé directement à la bonne taille.

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_loimport()</code> .

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 `pg_lo_export` et `pg_lo_open` .

7.83.49 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. Le descripteur de fichier contient les informations de connexion. Ne refermez pas la connexion avant d'avoir fermé l'objet. `objoid` est un Oid valide de Large Object, et `mode` peut prendre es valeurs suivantes : "r", "w", ou "rw".

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_loopen()</code> .

Voir aussi `pg_lo_close` et `pg_lo_create` .

7.83.50 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 <code>pg_loreadall()</code> .

Voir aussi `pg_lo_read` .

7.83.51 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. ^{large_object} est un identifiant valide d'objet de grande taille, et ^{len} indique la taille maximale de mémoire alloué à l'objet de grande taille.

Note

Auparavant, cette fonction s'appelait <code>pg_loread()</code> .

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

Voir aussi pg_lo_read_all .

7.83.52 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} . Le paramètre ^{whence} peut prendre les valeurs de `PGSQL_SEEK_SET` , `PGSQL_SEEK_CUR` ou `PGSQL_SEEK_END` .

Voir aussi pg_lo_tell .

7.83.53 pg_lo_tell() : Retourne la position courante dans un objet de grande taille

int **pg_lo_tell** (resource large_object)

pg_lo_tell retourne la position courante du pointeur de lecture sur l'objet de grande taille ^{large_object} . C'est le nombre d'octets depuis le début du fichier.

Voir aussi pg_lo_seek .

7.83.54 pg_lo_unlink() : Efface un objet de grande taille

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> .

Voir aussi pg_lo_create et pg_lo_import .

7.83.55 pg_lo_write() : Ecrit un objet de grande taille PGSQL

```
int pg_lo_write ( resource large_object , string data )
```

pg_lo_write écrit dans l'objet de grande taille autant de données possible, issues de la variable ^{buf} et retourne le nombre d'octets réellement écrits, ou ^{FALSE} en cas d'erreur. ^{fd} est un descripteur d'objet de grande taille, obtenu avec pg_lo_open .

Pour manipuler un objet de grande taille (lo), il est nécessaire de placer les opérations dans un bloc de transaction.

Voir aussi pg_lo_create et pg_lo_open .

Note

Auparavant, cette fonction s'appelait pg_lowrite() .

7.83.56 pg_meta_data() : Lit les données méta 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. Si une erreur survient, pg_meta_data retourne ^{FALSE} .

Lire les méta-données d'une table PostgreSQL

```
<?php
$dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");

$meta = pg_meta_data($dbconn, 'authors');
if (is_array($meta)) {
    echo '<pre>';
    var_dump ($meta);
    echo '</pre>';
}
?>
```

L'exemple ci-dessus pourrait afficher :

```
array(3) {
  ["author"]=>
  array(5) {
    ["num"]=>
    int(1)
    ["type"]=>
    string(7) "varchar"
    ["len"]=>
    int(-1)
    ["not null"]=>
    bool(false)
    ["has default"]=>
    bool(false)
  }
  ["year"]=>
  array(5) {
    ["num"]=>
    int(2)
    ["type"]=>
    string(4) "int2"
    ["len"]=>
    int(2)
    ["not null"]=>
    bool(false)
    ["has default"]=>
    bool(false)
  }
  ["title"]=>
  array(5) {
    ["num"]=>
    int(3)
    ["type"]=>
    string(7) "varchar"
    ["len"]=>
```

```
int(-1)
["not null"]=>
bool(false)
["has default"]=>
bool(false)
}
}
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 [pg_convert](#) .

7.83.57 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. L'argument doit être un identifiant de résultat valide retourné par [pg_exec](#) . Cette fonction retournera -1 en cas d'erreur.

Note

Auparavant, cette fonction s'appelait [pg_numfields\(\)](#) .

Voir aussi [pg_num_rows](#) et [pg_affected_rows](#) .

7.83.58 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. L'argument doit être un identifiant de résultat valide retourné par [pg_exec](#) . Cette fonction retournera -1 en cas d'erreur.

Note

Utilisez la fonction [pg_affected_rows](#) pour connaître le nombre de lignes affectées par les commandes SQL INSERT, UPDATE et DELETE.

Note

Auparavant, cette fonction s'appelait [pg_numrows\(\)](#) .

Voir aussi [pg_num_fields](#) et [pg_affected_rows](#) .

7.83.59 pg_options() : Retourne les options PostgreSQL

```
string pg_options ( resource connection )
```

[pg_options](#) retourne une chaîne contenant les options de la connexion PostgreSQL [connection](#) .

7.83.60 pg_pconnect() : Etablit une connexion PostgreSQL persistante

resource **pg_pconnect** (string connection_string)

pg_pconnect retourne une ressource de connexion persistante en cas de succès, ou `FALSE` en cas d'erreur.

Pour une description détaillée de la chaîne `connection_string`, voyez pg_connect.

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 referemera pas les connexions persistantes générées par pg_pconnect.

Voir aussi pg_connect et la section sur les connexions persistante pour plus d'informations.

7.83.61 pg_ping() : Pingue la connexion à la base

bool **pg_ping** (resource connection)

pg_ping pingue la connexion à la base de données, et essaie de se reconnecter si la connexion est perdue. pg_ping retourne `TRUE` si la connexion est active, et `FALSE` sinon.

Exemple avec pg_ping

```
<?php
$conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

if (!pg_ping($conn))
    die("La connexion est morte\n");
?>
```

Voir aussi pg_connection_status et pg_connection_reset.

7.83.62 pg_port() : Retourne le numéro de port

int **pg_port** (resource connection)

pg_port retourne le numéro de port de la connexion identifiée `connection`.

7.83.63 pg_put_line() : Envoie une chaîne au serveur PostgreSQL

bool **pg_put_line** (resource connection , string data)

pg_put_line envoie une chaîne (terminée par `NULL`) au serveur PostgreSQL. Ceci est pratique pour effectuer des insertions très rapides dans une table, initiée par une opération de copie PostgreSQL copy-operation. Le caractère final `NULL` est automatiquement ajouté. Cette fonction retourne `TRUE` en

cas de succès, `FALSE` en cas d'échec..

Note

Notez que l'application doit explicitement ajouter les deux caractères "\" à la fin de la chaîne pour indiquer au serveur qu'elle a fini d'envoyer des données.

Insertion à grande vitesse dans une table

```
<?php
$conn = pg_connect("dbname=foo");
pg_exec($conn, "create table bar (a int4, b char(16), d float8)");
pg_exec($conn, "copy bar from stdin");
pg_put_line($conn, "3\tBonjour le monde\t4.5\n");
pg_put_line($conn, "4\tAu revoir le monde\t7.11\n");
pg_put_line($conn, "\\.\n");
pg_end_copy($conn);
?>
```

Voir aussi [pg_end_copy](#) .

7.83.64 pg_query() : Exécute une requête

resource **pg_query** (resource connection , string query)

[pg_query](#) retourne une ressource de résultat si la requête a pu être exécutée. Elle retourne `FALSE` en cas d'échec, ou si la connexion n'est pas valide. Les détails sur les erreurs rencontrées est accessible via la fonction [pg_last_error](#) , si la connexion est valide. [pg_last_error](#) émet une requête SQL au serveur PostgreSQL, spécifié avec le paramètre `connection` . `connection` doit être une connexion valide, retournée par la fonction [pg_connect](#) ou [pg_pconnect](#) . La valeur retournée par cette fonction est une ressource de résultat. Pour accéder aux lignes lues, il faudra utiliser les autres fonctions PostgreSQL, telles que [pg_fetch_array](#) .

Note

`connection` est un paramètre optionnel de [pg_query](#) . Si `connection` est omis, la connexion par défaut sera utilisée. La connexion par défaut est la dernière connexion ouverte avec [pg_connect](#) ou [pg_pconnect](#) .

Bien que `connection` puisse être omis, il n'est pas recommandé de le faire, car il peut se révéler difficile de retrouver la connexion dans un script.

Note

Cette fonction s'appelait [pg_exec\(\)](#) . [pg_exec\(\)](#) est toujours disponible pour des raisons de compatibilité, mais les utilisateurs sont encouragés à utiliser le nouveau nom.

Voir aussi [pg_connect](#) , [pg_pconnect](#) , [pg_fetch_array](#) , [pg_fetch_object](#) , [pg_num_rows](#) et [pg_affected_rows](#) .

7.83.65 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](#) .

Voir aussi [pg_query](#) , [pg_send_query](#) , [pg_get_result](#) , [pg_last_error](#) et [pg_last_notice](#) .

7.83.66 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`. pg_result_seek retourne `FALSE`, en cas d'erreur.

Voir aussi pg_fetch_row, pg_fetch_assoc, pg_fetch_array, pg_fetch_object et pg_fetch_result.

7.83.67 pg_result_status() : Lit le statut du résultat

int **pg_result_status** (resource result)

pg_result_status lit le statut du résultat `result`. Les valeurs retournées possibles sont `PGSQL_EMPTY_QUERY`, `PGSQL_COMMAND_OK`, `PGSQL_TUPLES_OK`, `PGSQL_COPY_TO`, `PGSQL_COPY_FROM`, `PGSQL_BAD_RESPONSE`, `PGSQL_NONFATAL_ERROR` et `PGSQL_FATAL_ERROR`.

Voir aussi pg_connection_status.

7.83.68 pg_select() : Effectue une selection

array **pg_select** (resource connection , string table_name , array assoc_array , int options)

pg_select effectue une requête de SELECT avec les lignes spécifiées par le tableau `assoc_array`, au format `champ=>valeur`. Lorsque la requête réussit, pg_select retourne un tableau contenant toutes les lignes et champs qui vérifient la condition spécifiée par `assoc_array`. Si `options` est spécifiée, pg_convert est appliquée à `assoc_array` avec l'option spécifiée.

Exemple avec pg_select

```
<?php
$db = pg_connect ('dbname=foo');
// C'est correct, car $_POST est convertit automatiquement
$rec = pg_select($db, 'post_log', $_POST);
if ($rec) {
    echo "Lignes lues\n";
    var_dump($rec);
}
else {
    echo "Problème dans les données utilisateur\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.

Voir aussi pg_convert.

7.83.69 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 exécute la requête `query` sur la connexion `connection`, de manière asynchrone. Contrairement à pg_query, cette fonction permet d'envoyer plusieurs requêtes à un serveur PostgreSQL et de lire les

résultats un par un, avec la fonction [pg_get_result](#) . L'exécution du script n'est pas bloquée durant l'exécution de la requête. Utilisez [pg_connection_busy](#) pour vérifier si la connexion est encore occupée. 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 finie.

Exécute une requête PostgreSQL asynchrone

```
<?php
    $dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");

    if (!pg_connection_busy($dbconn)) {
        pg_send_query($dbconn,"select * from authors; select count(*) from authors;");
    }

    $res1 = pg_get_result($dbconn);
    echo "Premier appel à pg_get_result() : $res1\n";
    $rows1 = pg_num_rows($res1);
    echo "$res1 has $rows1 records\n\n";

    $res2 = pg_get_result($dbconn);
    echo "Second appel à pg_get_result() : $res2\n";
    $rows2 = pg_num_rows($res2);
    echo "$res2 has $rows2 records\n";
?>
```

L'exemple ci-dessus pourrait afficher :

Premier appel à pg_get_result() : Resource id #3
Resource id #3 has 3 records

Second appel à pg_get_result() : Resource id #4
Resource id #4 has 1 records

Voir aussi [pg_query](#) , [pg_cancel_query](#) , [pg_get_result](#) et [pg_connection_busy](#) .

7.83.70 pg_set_client_encoding() : Choisit l'encodage du client

```
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 sinon.

`encoding` est l'encodage du client, et peut être SQL_ASCII, EUC_JP, EUC_CN, EUC_KR, EUC_TW, UNICODE, MULE_INTERNAL, LATINX (X=1...9), KOI8, WIN, ALT, SJIS, BIG5, WIN1250.

Note

Cette fonction requiert PHP-4.0.2 ou plus récent et PostgreSQL-7.0 ou plus récent.

Jadis, [pg_set_client_encoding](#) s'appelait `pg_setclientencoding()`.

Voir aussi [pg_client_encoding](#) .

7.83.71 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.

`pathname` et `mode` sont les mêmes arguments que pour la fonction `fopen` (`mode` par défaut à 'w'), `connection` indique la connexion à suivre. Par défaut, c'est la dernière ouverte.

`pg_trace` retourne `TRUE` si `pathname` a pu être ouvert en écriture, et `FALSE` sinon.

Voir aussi `fopen` et `pg_untrace`.

7.83.72 pg_tty() : Retourne le nom de tty

string `pg_tty` (resource connection)

`pg_tty` retourne le nom de tty de la connexion associée à `connection`.

7.83.73 pg_unescape_bytea() : Supprime la protection d'une chaîne de type bytea

string `pg_unescape_bytea` (string data)

`pg_unescape_bytea` supprime la protection des caractères de type bytea, mis dans la chaîne `data` par la fonction `pg_escape_bytea`. La chaîne originale est ainsi retournée.

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-même.

`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 jeux de caractères, il peut arriver des erreurs. Il faut alors forcer le transtypage manuellement.

Voir aussi `pg_escape_bytea` et `pg_escape_string`

7.83.74 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`. `connection` indique la connexion à suivre. Par défaut, c'est la dernière ouverte.

`pg_untrace` retourne toujours `TRUE`.

Voir aussi `pg_trace`.

7.83.75 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.

Exemple avec pg_update

```
<?php
$db = pg_connect ('dbname=foo');
$data = array('field1'=>'AA', 'field2'=>'BB');
// C'est correct, car $_POST est convertit automatiquement
$res = pg_update($db, 'post_log', $_POST, $data);
if ($res) {
    echo "Les données ont été modifiées : $res\n";
}
else {
    echo "Problème dans les données utilisateur\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.

Voir aussi pg_convert .

7.84 Contrôle des processus

7.84.1 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)` and `signal(2)`, ou bien à consulter un ouvrage de référence comme "Advanced Programming in the UNIX Environment" par W. Richard Stevens (Addison-Wesley).

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.84.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.84.3 Installation

Le support du contrôle des processus n'est pas activé par défaut en ^{PHP}. Pour cela, vous devez utiliser l'option de compilation `--enable-pcntl`.

Note

Actuellement, ce module ne fonctionne pas sur les environnement non Unix comme Windows.

7.84.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.84.5 Types de ressources

Cette extension ne définit aucune ressource.

7.84.6 Constantes prédefinies

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*)

7.84.7 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");
} else if ($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");
    }

    // configuration des gestionnaires de signaux
    pcntl_signal(SIGTERM, "sig_handler");
    pcntl_signal(SIGHUP, "sig_handler");
  
```

```
// 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
    }
}

?>
```

7.84.8 Voir aussi

Jetez un oeil à la section sur les fonctions [POSIX](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [Voir aussi](#)
- [pcntl_exec](#)
- [pcntl_fork](#)
- [pcntl_signal](#)
- [pcntl_waitpid](#)
- [pcntl_wexitstatus](#)
- [pcntl_wifexited](#)
- [pcntl_wifsignaled](#)
- [pcntl_wifstopped](#)
- [pcntl_wstopsig](#)
- [pcntl_wtermsig](#)

7.84.10 pcntl_fork() : Forks the currently running process

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("could not fork");
} else if ($pid) {
    // we are the parent
} else {
    // we are the child
}
?>

```

Voir aussi [pcntl_waitpid](#) et [pcntl_signal](#).

7.84.11 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

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 tick 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:
            print "Reçu le signe SIGUSR1...\n";
            break;
        default:
            // gestion des autres signaux
    }
}

print "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_quelquechose");

print "Génération d'un signal SIGTERM à moi même...\n";

// envoi de SIGUSR1 à l'identifiant de processus courant
posix_kill(posix_getpid(), SIGUSR1);

print "Done\n"

```

```
?>
```

Voir aussi [pcntl fork](#) et [pcntl waitpid](#) .

7.84.12 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 que un processus fils spécifié par le paramètre `pid` ait terminé, ou bien 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 utilisé 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 :

<code>< -1</code>	attend que tous les processus fils dont l'identifiant de groupe est égal à la valeur absolue de <code>pid</code> soient terminés.
<code>-1</code>	attend que tous les processus fils soient terminés. Ceci est le même comportement que celui de la fonction <code>wait</code> .
<code>0</code>	attend que tous les processus fils dont l'identifiant de groupe est égal à celui du processus courant soient terminés.
<code>> 0</code>	attend que le processus fils dont l'identifiant est égal à <code>pid</code> soit terminé.

[pcntl_waitpid](#) enregistrera des informations sur le statut courant du processus dans le paramètre `status` , qui peut être accédé avec les fonctions suivantes : [pcntl_wifexited](#) , [pcntl_wifstopped](#) , [pcntl_wifsignaled](#) , [pcntl_wexitstatus](#) , [pcntl_wtermsig](#) et [pcntl_wstopsig](#) .

Le paramètre `options` peut prendre la valeur de zéro, ou plusieurs des constantes globales suivantes (combinez les avec l'opérateur `OR`) :

<code>WNOHANG</code>	retourne immédiatement si aucun processus fils ne s'est terminé.
<code>WUNTRACED</code>	retourne lorsque les processus fils sont arrêtés et que leur status 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](#) .

7.84.13 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 fourni à la fonction [pcntl_waitpid](#) , qui avait réussi.

Voir aussi [pcntl_waitpid](#) et [pcntl_wifexited](#) .

7.84.14 `pcntl_wifexited()` : Retourne ^{TRUE} si le code de retour représente une fin normale

```
int 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 fourni à la fonction `pcntl_waitpid`, qui avait réussi.

Voir aussi `pcntl_waitpid` et `pcntl_wexitstatus`.

7.84.15 `pcntl_wifsignaled()` : Returns ^{TRUE} if status code represents a termination due to a signal

```
int pcntl_wifsignaled ( int status )
```

Returns ^{TRUE} if the child process exited because of a signal which was not caught.

Le paramètre `status` est le paramètre fourni à la fonction `pcntl_waitpid`, qui avait réussi.

Voir aussi `pcntl_waitpid` et `pcntl_signal`.

7.84.16 `pcntl_wifstopped()` : Returns ^{TRUE} if child process is currently stopped

```
int pcntl_wifstopped ( int status )
```

Returns ^{TRUE} if the child process which caused the return is currently stopped; this is only possible if the call to `pcntl_waitpid` was done using the option `WUNTRACED`.

Le paramètre `status` est le paramètre fourni à la fonction `pcntl_waitpid`, qui avait réussi.

Voir aussi `pcntl_waitpid`.

7.84.17 `pcntl_wstopsig()` : Returns the signal which caused the child to stop

```
int pcntl_wstopsig ( int status )
```

Returns the number of the signal which caused the child to stop. This function is only useful if `pcntl_wifstopped` returned ^{TRUE}.

Le paramètre `status` est le paramètre fourni à la fonction `pcntl_waitpid`, qui avait réussi.

Voir aussi `pcntl_waitpid` et `pcntl_wifstopped`.

7.84.18 `pcntl_wtermsig()` : Returns the signal which caused the child to terminate

`int` **pcntl_wtermsig** (`int` *status*)

Returns the number of the signal that caused the child process to terminate. This function is only useful if `pcntl_wifsignaled` returned `TRUE` .

Le paramètre `status` est le paramètre fourni à la fonction `pcntl_waitpid` , qui avait réussi.

Voir aussi `pcntl_waitpid` , `pcntl_signal` et `pcntl_wifsignaled` .

7.85 Exécution de programmes externes

7.85.1 Introduction

Ces fonctions fournissent la possibilité de passer directement des commandes au système, mais aussi de protéger le système des commandes passées.

7.85.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.85.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.85.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.85.5 Types de ressources

Cette extension ne définit aucune ressource.

7.85.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.85.7 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 prédefinies](#)
- [Voir aussi](#)
- [escapeshellarg](#)
- [escapeshellcmd](#)
- [exec](#)
- [passthru](#)
- [proc_close](#)
- [proc_get_status](#)

- [proc_nice](#)
- [proc_open](#)
- [proc_terminate](#)
- [shell_exec](#)
- [system](#)

7.85.9 `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. Généralement, cette fonction est utilisée comme ceci :

Exemple avec [escapeshellcmd](#)

```
<?php
system(escapeshellcmd($cmd));
?>
```

Voir aussi [escapeshellarg](#), [exec](#), [popen](#) et [system](#).

7.85.10 `exec()` : Exécute un programme externe

`string` **exec** (`string` *command*, `array` *output*, `int` *return_var*)

[exec](#) exécute la commande *command*, mais ne renvoie rien comme retour, hormis la dernière ligne du résultat de la commande. Pour exécuter une commande et obtenir le résultat sans aucun traitement, il faut utiliser la fonction [passthru](#).

Si l'argument *array* est présent, alors ce tableau sera rempli par les lignes retournées par la commande. Il faut noter que si ce tableau contient des éléments, [exec](#) ajoutera les nouvelles lignes à la fin du tableau. Si vous ne voulez pas que les nouveaux éléments soient concaténés, utilisez la fonction [unset](#) avec ce tableau avant de le passer à [exec](#).

Si l'argument *return_var* est présent en plus du tableau *array*, alors le statut de retour d'exécution sera inscrit dans cette variable.

Notez que si vous allez fournir des commandes qui proviennent d'un utilisateur, il est avisé d'utiliser la fonction [escapeshellcmd](#) pour s'assurer que l'utilisateur n'essaie pas de profiter des caractères spéciaux pour tromper le système.

Note

Si vous démarrez un programme en utilisant cette fonction, et que vous voulez qu'il continue de fonctionner en tâche de fond, vous devez vous assurer que le résultat de ce programme est redirigé vers un fichier ou une autre méthode d'archivage, car sinon, ^{PHP} va attendre la fin de l'exécution du programme.

Voir aussi [system](#), [passthru](#), [popen](#) et [escapeshellcmd](#).

7.85.11 passthru() : Exécute un programme externe et affiche le résultat brut.

```
void passthru ( string command , int return_var )
```

La fonction `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.

Voir aussi `exec`, `system`, `popen` et `escapeshellcmd`.

7.85.12 proc_close() : Ferme un processus ouvert par proc_open() et retourne le code de sortie du processus.

```
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 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.

7.85.13 proc_get_status() : Lit les informations concernant un processus

```
array proc_get_status ( resource process )
```

`proc_get_status` lit les données concernant le processus `process`, créé avec la fonction `proc_open`. Les informations collectées sont retournées dans un tableau, avec les éléments suivants :

élément	type	description
command	string	
pid	int	identifiant du processus
running	bool	TRUE si le processus fonctionne toujours, et FALSE si il est terminé
signaled	bool	Non documenté
stopper	bool	Non documenté
exitcode	int	Le code retourné par le processus (uniquement si l'élément running vaut FALSE)
termsig	int	Non documenté
stopsig	int	Non documenté

Voir aussi `proc_open`.

7.85.14 proc_nice() : Change la priorité d'exécution du processus courant

```
bool proc_nice ( int priority )
```

`proc_nice` modifie la priorité d'exécution du processus courant en lui donnant la nouvelle valeur de

`priority` . Si une erreur survient, comme par exemple l'impossibilité pour l'utilisateur de changer la priorité, une erreur de niveau `E_WARNING` est générée et `FALSE` est retournée. Sinon, `TRUE` est retournée.

Note

`proc_nice` n'est disponible que sur les systèmes qui disposent de capacités NICE. NICE est compatible avec : SVr4, SVID EXT, AT&T, X/OPEN, BSD 4.3. Par exemple, `proc_nice` n'est pas disponible sous Windows.

Voir aussi `proc_open` et `proc_terminate` .

7.85.15 `proc_open()` : Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties

resource `proc_open` (string `cmd` , array `descriptorspec` , array `pipes`)

`proc_open` est similaire à `pcopen` mais fournit un plus grand degré de contrôle sur l'exécution du programme. `cmd` est la commande qui doit être exécutée par le shell. `descriptorspec` est un tableau indexé, dont les clés représentent le numéro de descripteur, et la valeur représente la méthode avec laquelle `PHP` va passer ce descripteur au processus fils. `pipes` recevra un tableau indexé de pointeurs de fichiers, qui correspondent aux sorties dans `PHP` de tous les pipes qui sont créés. La valeur retournée est une ressource, qui représente le processus. Il faudra la détruire avec la fonction `proc_close` lorsque vous aurez fini.

Exemple avec `proc_open`

```
<?php
$descriptorspec = array(
    0 => array("pipe", "r"), // stdin est un pipe où le processus va lire
    1 => array("pipe", "w"), // stdout est un pipe où le processus va écrire
    2 => array("file", "/tmp/error-output.txt", "a"), // stderr est un fichier
);
$process = proc_open("php", $descriptorspec, $pipes);
if (is_resource($process)) {
    // $pipes ressemble à :
    // 0 => fichier accessible en écriture, connecté à l'entrée standard du processus fils
    // 1 => fichier accessible en lecture, connecté à la sortie standard du processus fils
    // Toute erreur sera ajoutée au fichier /tmp/error-output.txt

    fwrite($pipes[0], "<?php echo \"Bonjour le monde!\"; ?>");
    fclose($pipes[0]);

    while(!feof($pipes[1])) {
        echo fgets($pipes[1], 1024);
    }
    fclose($pipes[1]);
    // Il est important que vous fermiez les pipes avant d'appeler
    // proc_close() afin d'éviter un verrouillage.
    $return_value = proc_close($process);

    echo "La commande a retournée $return_value\n";
}
?>
```

Les numéros de descripteurs de fichiers de `descriptorspec` ne sont pas limités à 0, 1 et 2 : vous pouvez spécifier n'importe quel numéro de descripteur valide, et il sera passé au processus fils. Cela permettra à votre script d'inter opérer avec d'autres scripts, et d'être exécuté comme "co-processus". En particulier, c'est très pratique pour passer des mots de passes à des programmes comme PGP, GPG et openssl, avec une méthode très protégée. C'est aussi pratique pour lire des informations de statut fournies par ces programmes, sur des descripteurs auxiliaires.

Note

Compatibilité Windows : les descripteurs au-delà de 2 (stderr) sont accessibles au processus fils, sous la forme de pointeurs héritables, mais comme l'architecture Windows n'associe pas de nombre aux descripteurs de bas niveau, le processus fils n'a (actuellement) aucun moyen d'y

accéder. Stdin, stdout et stderr fonctionnent comme d'habitude.

Note

Cette fonction a été introduite en PHP 4.3.0.

Note

Si vous n'avez besoin que d'un processus uni-directionnel, [popen](#) sera plus pratique, car plus simple à utiliser.

Voir aussi [stream_select](#), [exec](#), [system](#), [passthru](#), [popen](#) et [escapeshellcmd](#).

7.85.16 `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 par [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 plate-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](#).

Voir aussi [proc_open](#), [proc_close](#) et [proc_get_status](#).

7.85.17 `shell_exec()` : Exécute une commande via le shell et retourne le résultat sous forme de chaîne.

```
string shell_exec ( string cmd )
```

[shell_exec](#) exécute la commande `cmd` via le shell et retourne le résultat sous forme de chaîne.

[shell_exec](#) est identique aux [opérateurs de backtick](#).

7.85.18 `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. Si une variable est fournie comme second argument, alors le code de statut de la commande y sera affecté.

Notez que si vous allez fournir des commandes qui proviennent d'un utilisateur, il est avisé d'utiliser la fonction [escapeshellcmd](#) pour s'assurer que l'utilisateur n'essaie pas de profiter des caractères spéciaux pour tromper le système.

system essaie automatiquement de vider les tampons du serveur web après chaque ligne de résultat PHP, lorsque ce dernier fonctionne comme un module.

system retourne la dernière ligne du retour, en cas de succès, et ^{FALSE} en cas d'échec.

Si vous devez exécuter une commande et récupérer tout le résultat sans aucune intervention, utilisez la fonction passthru .

Voir aussi exec , passthru , popen et escapeshellcmd .

7.86 Impression

7.86.1 Introduction

Ces fonctions ne sont disponibles que sous Windows 9.x, ME, NT4 et 2000. Elles ont été ajoutées en PHP 4.0.4.

7.86.2 Installation

Ajoutez la ligne `extension=php_printer.dll` dans votre fichier `php.ini`.

7.86.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>printer.default_printer</code>	<code>""</code>	<code>PHP_INI_ALL</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

- [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_elipse](#)
- [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](#)

7.86.5 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);
?>
```

7.86.6 printer_create_brush() : Crée une nouvelle brosse

mixed **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.

7.86.7 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](#)

```
$handle = printer_open();
printer_start_doc($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_endpage($handle);
printer_end_doc($handle);
printer_close($handle);

```

7.86.8 printer_create_font() : Crée une nouvelle police

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](#) .

7.86.9 printer_create_pen() : Crée un nouveau stylo

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.

7.86.10 `printer_delete_brush()` : Efface une brosse

`bool 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`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec. `handle` doit être une ressource de brosse valide.

7.86.11 `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.

7.86.12 `printer_delete_font()` : Efface une police

`bool 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`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec. `handle` doit être une ressource de police valide.

7.86.13 `printer_delete_pen()` : Efface un stylo

`bool 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`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec. `handle` doit être une ressource de stylo valide.

7.86.14 `printer_draw_bmp()` : Dessine un fichier BMP

`void printer_draw_bmp ( resource handle , string filename , int x , int y )`

`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);
?>

```

7.86.15 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);
?>

```

7.86.16 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);
?>
```

7.86.17 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);
?>
```

7.86.18 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);
?>
```

7.86.19 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);
?>

```

7.86.20 `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);
?>

```

7.86.21 `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);
?>

```

7.86.22 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.

7.86.23 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.

7.86.24 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
$handle = printer_open();
print printer_get_option($handle, PRINTER_DRIVERVERSION);
printer_close($handle);
?>

```

7.86.25 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) );
?>
```

7.86.26 `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
$handle = printer_open();
print printer_logical_fontheight($handle, 72);
printer_close($handle);
?>
```

7.86.27 `printer_open()` : Ouvre une connexion à une imprimante

`mixed 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();
?>
```

7.86.28 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);
?>
```

7.86.29 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);
?>
```

7.86.30 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, le style 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);
?>
```

7.86.31 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 copie à imprimer. `value` doit être un entier.
- `PRINTER_MODE` : spécifie le type de data ("text", "raw" ou "emf"), `value` doit être une chaîne de caractères.
- `PRINTER_TITLE` : spécifie le nom du document, `value` doit être une chaîne de caractères.
- `PRINTER_ORIENTATION` : spécifie l'orientation du papier, `value` peut être `PRINTER_ORIENTATION_PORTRAIT` ou `PRINTER_ORIENTATION_LANDSCAPE`.
- `PRINTER_RESOLUTION_Y` : spécifie la résolution en ordonnées, en DPI, `value` doit être un entier.
- `PRINTER_RESOLUTION_X` : spécifie la résolution en abscisse, en DPI, `value` doit être un entier.
- `PRINTER_PAPER_FORMAT` : spécifie un format de papier prédéfini : donnez à `value` 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`. `value` peut alors être l'une des constantes suivantes :
 - ◆ `PRINTER_FORMAT_CUSTOM` : vous laissez 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. `value` 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. `value` 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 `value` /100. Par exemple, si vous

donnez un facteur d'échelle de 50, l'impression sera de la moitié de la taille du document original. ^{value} doit être un entier.

- `PRINTER_BACKGROUND_COLOR` : spécifie la couleur de fond pour le contexte actuel. ^{value} 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. ^{value} 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. ^{value} 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);
?>
```

7.86.32 `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);
?>
```

7.86.33 `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.

7.86.34 `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);
?>
```

7.87 Pspell

7.87.1 Introduction

La librairie pspell vous permet de vérifier l'orthographe d'un mot, et suggérer des corrections.

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.87.2 Pré-requis

Vous aurez besoin des librairies aspell et pspell, disponibles à <http://aspell.sourceforge.net/> et <http://aspell.net/> (respectivement). Il faut aussi ajouter l'option `--with-pspell[=dir]` lors de la compilation de PHP.

7.87.3 Installation

Si vous avez installé les librairies PSPELL, vous devez simplement ajouter l'option `--with-pspell[=dir]`, lors de la compilation de PHP.

7.87.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.87.5 Types de ressources

Cette extension ne définit aucune ressource.

7.87.6 Constantes prédefinies

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 prédefinies](#)
- [pspell_add_to_personal](#)
- [pspell_add_to_session](#)

- [pspell_check](#)
- [pspell_clear_session](#)
- [pspell_config_create](#)
- [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](#)

7.87.8 [pspell_add_to_session\(\)](#) : Ajoute le mot au dictionnaire personnel de la session courante

```
int 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](#).

7.87.9 [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.

[pspell_check](#)

```
<?php
$pspell_link = pspell_new ("french");
if (pspell_check ($pspell_link, "testt")) {
    echo "L'orthographe est exacte";
} else {
    echo "Désolé, mauvaise orthographe";
}
?>
```

7.87.10 [pspell_clear_session\(\)](#) : Remet à zéro la session courante

```
int 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
```

```
?>
```

7.87.11 `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 `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 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 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 mode (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);
?>
```

7.87.12 `pspell_config_ignore()` : Ignore les mots des moins de N caractères

```
int pspell_config_ignore ( int dictionary_link , int n )
```

`pspell_config_ignore` doit être utilisé avec une configuration avant d'appeler `pspell_new_config`. Cette fonction permet au vérificateur d'ignorer les mots trop courts.

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
?>
```

7.87.13 `pspell_config_mode()` : Change le mode de suggestion

```
int pspell_config_mode ( int dictionary_link , int mode )
```

`pspell_config_mode` doit être appelé avant `pspell_new_config`. Cette fonction détermine le nombre de suggestions qui seront retournés 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 mode (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");
?>
```

7.87.14 `pspell_config_personal()` : Choisit le fichier qui contient le dictionnaire personnel

```
int pspell_config_personal ( int dictionary_link , string file )
```

`pspell_config_personal` doit être appelé 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. 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_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");
?>
```

7.87.15 `pspell_config_repl()` : Choisit le fichier qui contient les paires de remplacement

```
int pspell_config_repl ( int dictionary_link , string file )
```

[pspell_config_repl](#) doit être appelé 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 utilisé 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");
?>
```

7.87.16 pspell_config_runtogether() : Considère deux mots accolés comme un composé.

```
int pspell_config_runtogether ( int dictionary_link , bool flag )
```

[pspell_config_runtogether](#) doit être appelé dans une configuration avant d'appeler [pspell_new_config](#). Cette fonction indique si deux mots accolés doivent être traités comme un composé valide, même si 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,
<TT>TRUE</TT>
);
$pspell_link = pspell_new_config ($pspell_config);
pspell_check ($pspell_link, "thecat");
?>
```

7.87.17 pspell_config_save_repl() : Active la sauvegarde des paires de remplacement

```
int pspell_config_save_repl ( int dictionary_link , bool flag )
```

[pspell_config_save_repl](#) doit être appelé 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 toutes façons, et si ce n'est pas le cas, elles ne seront pas sauvées. 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.

7.87.18 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 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);
?>
```

7.87.19 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 modifiée 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 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 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 mode (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_new_personal`

```
<?php
$spell_link = pspell_new_personal ( "/var/dictionaries/custom.pws",
                                   "en", "", "", "",
                                   PSPELL_FAST | PSPELL_RUN_TOGETHER );
?>
```

7.87.20 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 utiliser 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, séparé 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 ``american'', ``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 mode (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, construits 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
$spell_link = pspell_new("en", "", "", "", PSPELL_FAST|PSPELL_RUN_TOGETHER);
?>
```

7.87.21 pspell_save_wordlist() : Sauve le dictionnaire personnel dans un fichier

```
int 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 spécifié 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);
?>
```

7.87.22 pspell_store_replacement() : Enregistre une paire de remplacement pour un mot

int **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);
?>
```

7.87.23 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
$pspell_link = pspell_new ("english");
if (!pspell_check ($pspell_link, "testt")){
    $suggestions = pspell_suggest ($pspell_link, "testt");
    for ($i=0; $i < count ($suggestions); $i++) {
        echo "Orthographes suggérées : " . $suggestions[$i] . "<br>";
    }
}
?>
```

7.88 Readline (GNU)

7.88.1 Introduction

Les fonctions readline implémente une interface avec la librairie 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 commande shell.

Note

Cette extension n'est pas disponible sur les plate-formes Windows.

7.88.2 Pré-requis

Pour utiliser les fonctions readline, vous devez installer la librairie libreadline. Vous pouvez la trouver sur la page du projet GNU, à <http://cnswww.cns.cwru.edu/~chet/readline/rltop.html> . Elle est maintenue par Chet Ramey, qui est l'auteur de Bash.

Vous pouvez aussi utiliser ces fonctions avec la librairie libedit, un remplacement de la librairie readline, qui n'est pas GPL. La librairie libedit est sous license BSD, et elle est disponible depuis <http://sourceforge.net/projects/libedit/> .

7.88.3 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 librairie libedit readline, en remplacement, compilez ^{PHP} avec l'option `--with-libedit[=DIR]` .

7.88.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.88.5 Types de ressources

Cette extension ne définit aucune ressource.

7.88.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)

- [Constantes prédefinies](#)
- [readline_add_history](#)
- [readline_clear_history](#)
- [readline_completion_function](#)
- [readline_info](#)
- [readline_list_history](#)
- [readline_read_history](#)
- [readline_write_history](#)
- [readline](#)

7.88.8 `readline_clear_history()` : Efface l'historique

`bool readline_clear_history ( void )`

[readline_clear_history](#) efface tout l'historique.

7.88.9 `readline_completion_function()` : Enregistre une fonction de complétion

`bool readline_completion_function ( string line )`

[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.

7.88.10 `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`.

7.88.11 `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.

7.88.12 `readline_read_history()` : Lit l'historique

`bool readline_read_history ( string filename )`

[readline_read_history](#) lit une ligne de l'historique.

7.88.13 `readline_write_history()` : Ecrit dans l'historique

`bool readline_write_history ( string filename )`

`readline_write_history` écrit `filename` dans l'historique.

7.88.14 `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());

?>
```


7.89 Recode (GNU)

7.89.1 Introduction

Ce module contient l'interface à la librairie GNU Recode library, version 3.5. La librairie 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 librairie 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 plate-formes Windows.

7.89.2 Pré-requis

il faut que vous ayez la librairie GNU Recode 3.5 ou plus récent, installée sur votre système. Vous pouvez télécharger GNU Recode [ici](#) .

7.89.3 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 librairie c-client de IMAP et recode ont toutes les deux leur propre fonction `hash_lookup()` et les extensions mysql et recode ont toutes les deux leur fonction `hash_insert` .

Attention

L'extension [IMAP](#) ne peut être utilisée simultanément avec l'extension [recode](#) ou l'extension [YAZ](#) . Ceci est dû au fait que ces deux librairies utilisent un symbole interne commun.

7.89.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.89.5 Types de ressources

Cette extension ne définit aucune ressource.

7.89.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)

- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [recode_file](#)
- [recode_string](#)
- [recode](#)

7.89.8 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 `FALSE`, en cas d'échec, et `TRUE` 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
print recode_string("us..flat",
                    "Le caractère suivant est diacritique : &acute;");
?>
```

7.89.9 recode() : Recode une fonction grâce à une requête

string **recode** (string request , string string)

Note

[recode](#) est un alias de [recode_string](#). Elle a été ajoutée en PHP 4.

7.90 Expressions régulières compatibles Perl

7.90.1 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 antislash (\). Si un délimiteur doit être utilisé dans l'expression, il faudra l'échapper avec un antislash. Depuis PHP 4.0.4, vous pouvez utiliser les délimiteurs (), {}, [], et <>, comme en Perl.

Le délimiteur final peut être suivi d'options qui affecteront la recherche. Voir aussi [options de recherche](#).

7.90.2 Pré-requis

Le support des expressions régulières est assuré par la librairie 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/](http://ftp.csx.cam.ac.uk/pub/software/programming/pcre/).

7.90.3 Installation

A partir de ^{PHP} 4.2.0, ces fonctions activées par défaut. Pour les anciennes versions, vous devez configurer et compiler ^{PHP} avec l'option `--with-pcre-regex[=DIR]` afin de pouvoir les utiliser. Vous pouvez aussi désactiver le support des expressions régulières avec l'option `--without-pcre-regex`.

La version Windows de ^{PHP} dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

7.90.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.90.5 Types de ressources

Cette extension ne définit aucune ressource.

7.90.6 Constantes prédefinies

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> contienne les résultats qui correspondent à la première parenthèse capturante, <code>\$matches[2]</code> contienne les résultats qui correspondent à la deuxième parenthèse capturante, etc... Cette constante est utilisée avec preg_match_all .
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> contient la deuxième série de résultat, etc... Cette constante est utilisée avec preg_match_all .

PREG_OFFSET_CAPTURE	Voir la description de PREG_SPLIT_OFFSET_CAPTURE . 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.

7.90.7 Exemples

Exemples de masques valides
• <code>/<\w+>/</code> • <code> (\d{3})-\d+ Sm</code> • <code>/^(?i)php[34]/</code> • <code>{^\s+(\s+)?\$}</code>

Exemples de masques invalides
• <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 prédéfinies](#)
- [Exemples](#)
- [options de recherche](#)
- [syntaxe des masques](#)
- [preg_grep](#)
- [preg_match_all](#)
- [preg_match](#)
- [preg_quote](#)
- [preg_replace_callback](#)
- [preg_replace](#)
- [preg_split](#)

7.90.9 syntaxe des masques() : Fonctionnement des expressions régulières.

La bibliothèque PCRE est un ensemble de fonctions qui implémentent la recherche par expressions régulières, 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.

7.90.10 preg_grep() : Retourne un tableau avec les résultat de la recherche

array **preg_grep** (string pattern , array input)

`preg_grep` retourne un tableau qui contient les éléments de `input` qui satisfont le masque `pattern` .

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
preg_grep("/^(\d+)?\.\d+$/", $array);
?>
```

7.90.11 `preg_match_all()` : Expression régulière globale

`int preg_match_all ( string pattern , string subject , array matches , int flags )`

`preg_match_all` analyse `subject` pour trouver l'expression `pattern` et met les résultats dans `matches`, dans l'ordre spécifié par `order`.

Après avoir trouvé un premier résultat, la recherche continue jusqu'à la fin de la chaîne.

`order` 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",
    "<b>exemple : <\/b><div align=left>un test<\/div>",
    $out,
    PREG_PATTERN_ORDER);
print $out[0][0].", ".$out[0][1]."\n";
print $out[1][0].", ".$out[1][1]."\n";
?>
```

Cet exemple va afficher :

Résultat avec `PREG_PATTERN_ORDER`

```
<b>exemple : <\/b>, <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 `$matches[0]` contient la première série de résultat, `$matches[1]` contient la deuxième série de résultat, etc...

Exemple avec `preg_match_all` et `PREG_SET_ORDER`

```
<?php
preg_match_all("|<[^>]+>(.*?)<\/[^>]+>|U",
    "<b>exemple : <\/b><div align=left>un test<\/div>",
    $out,
    PREG_SET_ORDER);
print $out[0][0].", ".$out[0][1]."\n";
print $out[1][0].", ".$out[1][1]."\n";
?>
```

Cet exemple va afficher :

Résultat avec `PREG_SET_ORDER`

```
<b>exemple : <\/b>, 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.

`preg_match_all` retourne le nombre de résultat 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 quelquechose 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 = "<B>Texte en gras</B><a href=salut.html>clicque moi</a>";

preg_match_all ("/(<([\w]+)[^>]*>)(.*)(<\/\2>)/", $html, $matches);

for ($i=0; $i< count($matches[0]); $i++) {
    echo "Trouvé : ".$matches[0][$i]."\n";
    echo "partie 1 : ".$matches[1][$i]."\n";
    echo "partie 2 : ".$matches[3][$i]."\n";
    echo "partie 3 : ".$matches[4][$i]."\n\n";
}
?>
```

Cet exemple va produire :

```
Trouvé : <B>Texte en gras</B>
partie 1 : <B>
partie 2 : Texte en gras
partie 3 : </B>
```

```
Trouvé : <a href=salut.html>clicque moi</a>
partie 1 : <a href=salut.html>
partie 2 : clicque moi
partie 3 : </a>
```

Voir aussi `preg_match`, `preg_replace` et `preg_split`.

7.90.12 preg_match() : Expression régulière standard

```
int preg_match ( string pattern , string subject , array matches , int flags )
```

`preg_match` analyse ^{subject} pour trouver l'expression ^{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.

`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'il 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.

Extraction d'un numéro de page d'une chaîne.

```
<?php
if (preg_match("/page\s+(\d+)/i", "Aller à la page numéro 9.", $parts))
    print "La page suivante est $parts[1]";
else
    print "Page introuvable.";
?>
```

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.)) {
    print "Un mot a été trouvé.";
} else {
    print "Un mot n'a pas été trouvé.";
}
if (preg_match ("/\bweb\b/i", "PHP est le meilleur langage de script pour les webagency.)) {
    print "Un mot a été trouvé.";
} else {
    print "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:\V\)?([^\V]+)/i",
    "http://www.php.net/index.html", $matches);
$host = $matches[2];
// repérer les deux derniers segments du nom de l'hôte
preg_match("/[^\.\V]+\.[^\.\V]+$/", $host, $matches);
echo "Le nom de domaine est : ".$matches[0]."\n";
?>
```

Cet exemple va afficher :

Le nom de domaine est : php.net

Voir aussi `preg_match_all`, `preg_replace` et `preg_split`.

7.90.13 preg_quote() : Echappement des caractères spéciaux des expressions régulières.

```
string preg_quote ( string str, string delimiter )
```

`preg_quote` ajoute un antislash 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 : `. \ + * ? [ ^ ] $ ( ) { } = ! < > | :`

Protège des caractères spéciaux

```
<?php
$keywords = "$40 pour un g3/400";
$keywords = preg_quote ($keywords, "/");
echo $keywords;
// retourne \$40 pour un g3\400
?>
```

Mise en italique 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 régulière.
$textbody = "Ce livre est *très* difficile à trouver.";
$word = "**très*";
$textbody = preg_replace ("/".preg_quote($word)."/",
                        "<B>".$word."</B>",
                        $textbody);
?>
```

7.90.14 preg_replace_callback() : Rechercher/remplacer avec une expression régulière et fonction de callback

`mixed preg_replace_callback ( mixed pattern , callback callback , mixed subject , int limit )`

Le comportement de `preg_replace` est presque identique à celui de `preg_replace`, hormis le fait qu'à la place du paramètre `replacement`, 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 à jour pour 2003
$text = "Le premier avril est le 01/04/2002\n";
$text.= "Le dernier noel était le 12/24/2001\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 noel était le 12/24/2002
?>
```

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 pas 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](#) et [create_function](#).

7.90.15 preg_replace() : Rechercher et remplacer par expression régulière standard

`mixed preg_replace ( mixed pattern , mixed replacement , mixed subject , int limit )`

`preg_replace` analyse `subject` pour trouver l'expression `pattern` et remplace les résultats par `replacement`.

`replacement` 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.

Utilisation des références arrières avec des littéraux numériques

```
<?php
$string = "Avril 05, 2003";
$pattern = "/(\w+) (\d+), (\d+)/i";
$replacement = "\\${1}1,\\${3}";
print preg_replace($pattern, $replacement, $string);

/* Output
=====
Avril1,2003

*/
?>
```

Si la recherche n'aboutit à aucun résultat, `subject` sera inchangé.

Tous les paramètres de `preg_replace` peuvent être des tableaux.

Note

Lorsque vous utilisez des tableaux avec les paramètres `pattern` et `replacement`, les clés sont traitées dans l'ordre dans lequel elles apparaissent dans le tableau. Ce **n'est pas forcément** la même chose que l'ordre des index numérique. Si vous utilisez des index pour identifier quel `pattern` doit être remplacé par quel `replacement`, il est recommandé de faire un tri `ksort` sur chaque tableau avant de faire appel à `preg_replace`.

Utilisation de tableaux indexés avec `preg_replace`

```
<?php
$string = "The quick brown fox jumped over the lazy dog.";
```

```

$patterns[0] = "/quick/";
$patterns[1] = "/brown/";
$patterns[2] = "/fox/";

$replacements[2] = "bear";
$replacements[1] = "black";
$replacements[0] = "slow";

print preg_replace($patterns, $replacements, $string);

/* Output
=====

The bear black slow jumped over the lazy dog.

*/

/* By ksorting patterns and replacements,
   we should get what we wanted. */

ksort($patterns);
ksort($replacements);

print preg_replace($patterns, $replacements, $string);

/* Output
=====

The slow black bear jumped over the lazy dog.

*/
?>

```

Si `subject` est un tableau, alors l'opération sera appliquée à chacun des éléments du tableau, et le tableau sera retourné.

Si `pattern` et `replacement` sont des tableaux, alors `preg_replace` prend une valeur de chaque tableau, et l'utilise pour faire la recherche et le remplacement. Si `replacement` à moins d'éléments que `pattern`, alors la chaîne vide est utilisé pour le reste des valeurs. Si `pattern` est un tableau, et que `replacement` est une chaîne, alors cette chaîne sera utilisée pour chaque valeur de `pattern`. Le contraire n'aurait pas de sens.

/e force `preg_replace` à traiter `replacement` comme du code PHP une fois que les substitutions adéquates ont été faites. Conseil : assurez-vous que `replacement` est un code PHP valide, car sinon, PHP trouvera une erreur d'analyse (parse error) dans cette ligne.

/F indique que le paramètre `replacement` doit être considéré comme un nom de fonction. Cette fonction sera appelée, avec un tableau contenant les éléments trouvés comme arguments. La fonction doit retourner la chaîne de remplacement. Cette option a été ajoutée en PHP 4.0.4.

Remplacement de plusieurs valeurs simultanément

```

<?php
$patterns = array ("/(19|20)(\d{2})-(\d{1,2})-(\d{1,2})/",
                  "/^s*{(\w+)}\s*="/);
$replace = array ("\\3/\\4/\\1\\2", "$\\1 =");
print preg_replace ($patterns, $replace, "{startDate} = 1999-5-27");
?>

```

Cet exemple va afficher :

```
$startDate = 5/27/1999
```

Utilisation de l'option /e

```

<?php
preg_replace("/(</?)(\w+)([>]*>/e", "'\\1'.strtoupper('\\2').'\\3'", $html_body);
?>

```

Cela va mettre en majuscule toutes les balises HTML du texte.

Conversion HTML en texte

```
<?php
// $document contient un document HTML
// Ce script va effacer les balises HTML, les javascript
// et les espaces. Il remplace aussi quelques entités HTML
// courante en leur équivalent texte.
$search = array ('<script[?>]*?></script>'si", // Supprime le javascript
                '<[\r\n!]*?[^<>]*?>'si", // Supprime les balises HTML
                '([\r\n])[\s]+'", // Supprime les espaces
                '&(quot|#34);'i", // Supprime les entites HTML
                '&(amp|#38);'i",
                '&(lt|#60);'i",
                '&(gt|#62);'i",
                '&(nbsp|#160);'i",
                '&(iexcl|#161);'i",
                '&(cent|#162);'i",
                '&(pound|#163);'i",
                '&(copy|#169);'i",
                '&#(\d+);'e"); // Evaluation comme PHP
$replace = array ("",
                 "",
                 "\\1",
                 "\\n",
                 "&",
                 "<",
                 ">",
                 " ",
                 chr(161),
                 chr(162),
                 chr(163),
                 chr(169),
                 "chr(\\1)");
$text = preg_replace ($search, $replace, $document);
?>
```

Note

Le paramètre `limit` a été ajouté à partir de PHP 4.0.1pl2.

Voir aussi [preg_match](#), [preg_match_all](#) et [preg_split](#).

7.90.16 preg_split() : Eclate une chaîne par expression régulière

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.

Eclatement d'une chaîne de recherche

```
<?php
// scinde la phrase grâce aux virgules et espacements
// 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 offsets

```
<?php
$str = 'langage hypertexte, programmation';
$chars = preg_split('/ /', $str, -1, PREG_SPLIT_OFFSET_CAPTURE);
print_r($chars);
?>
```

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 Beta 3.

Voir aussi [spliti](#) , [split](#) , [implode](#) , [preg match](#) , [preg match all](#) et [preg replace](#) .

7.91 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 plate-formes Windows.

7.91.1 Pré-requis

Vous aurez besoin de la librairie Qt-library >=2.2.0.

7.91.2 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](#)

7.91.4 qdom_tree() : Crée une arbre à partir d'un code XML

object **qdom_tree** (*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.

7.92 Expressions régulières

7.92.1 Introduction

Note

PHP supporte aussi des expressions rationnelles compatibles Perl, avec l'extension PCRE functions . Ces fonctions supportent des recherches non-gourmandes, des assertions, des sous-masques conditionnels et toute une gamme de fonctionnalités absentes des expressions rationnelles POSIX.

Attention

Ces expressions rationnelles ne sont pas compatibles avec les données binaires. Les expressions PCRE le sont.

Les expressions régulières sont utilisées pour effectuer des manipulations complexes de chaînes de caractères. Les fonctions sont :

- ereg
- ereg_replace
- eregi
- eregi_replace
- split
- spliti

Ces fonctions requièrent toutes une expression régulière comme premier argument. PHP utilise les expressions régulières 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

7.92.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.92.3 Installation

Pour activer le support des expressions régulières 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".

Note

Ne changez pas la valeur de TYPE à moins que vous ne sachiez ce que vous faites.

La version Windows de ^{PHP} dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

7.92.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.92.5 Types de ressources

Cette extension ne définit aucune ressource.

7.92.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.92.7 Exemples

Expressions régulières
<pre> <?php ereg ("abc", \$string); /* Retourne <TT>TRUE</TT> si "abc" est trouvé quelque part dans la chaîne \$string. */ ereg ("^abc", \$string); /* Retourne <TT>TRUE</TT> si "abc" est trouvé au début de la chaîne \$string. */ ereg ("abc\$", \$string); /* Retourne <TT>TRUE</TT> si "abc" est trouvé à la fin de la chaîne \$string. */ ereg ("(ozilla.[23] MSIE.3)", \$HTTP_USER_AGENT); /* Retourne <TT>TRUE</TT> si le client est Netscape 2, 3 ou MSIE 3. */ ereg ("([[:alnum:]]+) ([[:alnum:]]+) ([[:alnum:]]+)", \$string,\$regs); /* Recherche trois mots séparés par des espaces dans les chaînes \$regs[1], \$regs[2] et \$regs[3]. */ \$string = ereg_replace ("^", "
", \$string); /* Insère une balise
 au début de la chaîne \$string. */ \$string = ereg_replace ("\$", "
", \$string); /* Insère une balise
 à la fin de la chaîne \$string. */ \$string = ereg_replace ("\n", "", \$string); /* Supprime toutes les nouvelles lignes de \$string. */ ?> </pre>

7.92.8 Voir aussi

Pour des expressions rationnelles compatibles Perl, voyez l'extension [PCRE](#) . Les commandes de recherches plus simples (avec support des caractères jokers) sont fournis par la fonction [fnmatch](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [Voir aussi](#)
- [ereg_replace](#)
- [ereg](#)

- [eregi_replace](#)
- [eregi](#)
- [split](#)
- [spliti](#)
- [sql_regcase](#)

7.92.10 ereg() : Expression régulière standard

`bool ereg ( string pattern , string string , array regs )`

Recherche dans la chaîne `string` les séquences de caractères qui correspondent au masque `pattern`.

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.

Si `ereg` trouve ses solutions pour les parenthèses capturantes, `$regs` contiendra exactement 10 éléments, même si il y avait plus ou moins de 10 parenthèses capturantes qui étaient valides. Cela n'a aucun effet sur les capacités de la fonction `ereg` à trouver d'autres sous chaînes. Si aucune valeur n'est trouvée, `$regs` ne sera pas modifié par `ereg`.

La recherche est sensible à la casse.

`ereg` retourne `TRUE` si une occurrence a été trouvée dans la chaîne 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 [ereg](#), [preg_match](#), [ereg_replace](#), [eregi_replace](#), [strpos](#) et [strstr](#).

7.92.11 eregi_replace() : Remplacement par expression régulière 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.

Voir aussi [ereg](#), [eregi](#) et [ereg_replace](#).

7.92.12 eregi() : Recherche par expression régulière insensible à la casse

bool **eregi** (string pattern , string string , array regs)

eregi 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
if (eregi("z", $string)) {
    echo "'$string' contient un 'z' ou un 'Z!';"
}
?>
```

Voir aussi ereg , ereg_replace , eregi_replace , stripos et striestr .

7.92.13 split() : Scinde une chaîne en un tableau, grâce à une expression régulière.

array **split** (string pattern , string string , int limit)

split retourne un tableau de chaînes : chacune d'entre elle est une sous-chaîne de ^{string} délimitée par les occurrences trouvées de l'expression régulière ^{pattern} . 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);
?>
```

Pour analyser une date qui est délimitée par des /, des points ou des tirets :

Exemple avec split

```
<?php
$date = "04/30/1973"; // Les délimiteurs peuvent être des tirets, points ou slash
list ($month, $day, $year) = split ('[/.-]', $date);
echo "Month: $month; Day: $day; Year: $year<br>\n";
?>
```

Notez que ^{pattern} est insensible à la casse

Notez bien que si vous n'avez pas besoin de la puissance des expressions régulières, il est plus rapide d'utiliser explode , qui n'utilise pas le moteur d'expressions régulières.

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 régulière. Si vous voulez utiliser n'importe quel caractère spécial des expressions régulières, vous devez les échapper. Si vous pensez que split (ou toute autre expression régulière) 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](#) .

7.92.14 spliti() : Scinde une chaîne en un tableau, grâce à une expression régulière

```
array spliti ( string pattern , string string , int limit )
```

[spliti](#) est identique à [split](#) , hormis le fait qu'elle ignore la casse.

Voir aussi [preg_spliti](#) , [split](#) , [explode](#) et [implode](#) .

7.92.15 sql_regcase() : Prépare une expression régulière pour effectuer une recherche insensible à la casse.

```
string sql_regcase ( string string )
```

[sql_regcase](#) retourne une expression régulière 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");
?>
```

affichera `[Ff][Oo][Oo] [Bb][Aa][Rr]` .

Cette expression sert à effectuer des recherches insensibles à la casse avec d'autres logiciels, qui n'acceptent les recherches insensibles à la casse.

7.93 Sémaphores et gestion de la mémoire partagée

7.93.1 Introduction

Ce module fournit un système de sémaphore. 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 mêmes 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.

Note
Cette extension n'est pas disponible sur les plate-formes Windows.

7.93.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.93.3 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`.

7.93.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<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 [ini_set](#).

7.93.5 Types de ressources

7.93.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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](#)

7.93.8 msg_get_queue() : Crée ou s'attache à une file de message

```
int 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.

Cette fonction a été introduite en PHP 4.3.0.

Voir aussi [msg_remove_queue](#), [msg_receive](#), [msg_send](#), [msg_stat_queue](#) et [msg_set_queue](#).

7.93.9 msg_receive() : Reçoit un message depuis une file de messages

```
bool msg_receive ( int 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 reception du message, auquel cas le

paramètre optionnel `errorcode` se verra assigné 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 retournée. 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 si 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>	Si 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_lrp_id` prend la valeur de l'identifiant de processus du processus appelant, `msg_qnum` est décrémenté 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](#).

Cette fonction a été introduite en PHP 4.3.0.

7.93.10 msg_remove_queue() : Détruit une file de messages

```
bool msg_remove_queue ( int 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.

Cette fonction a été introduite en PHP 4.3.0.

Voir aussi [msg_remove_queue](#), [msg_receive](#), [msg_stat_queue](#) et [msg_set_queue](#).

7.93.11 msg_send() : Envoie un message dans une file

```
bool msg_send ( int 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` vaut 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_lrpid` 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`.

Cette fonction a été introduite en PHP 4.3.0.

7.93.12 msg_set_queue() : Modifie des informations dans la file de messages

```
bool msg_set_queue ( int 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.

Cette fonction a été introduite en PHP 4.3.0.

Voir aussi `msg_remove_queue`, `msg_receive`, `msg_stat_queue` et `msg_set_queue`.

7.93.13 msg_stat_queue() : Retourne des informations sur la file de messages

```
array msg_stat_queue ( int 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 suivantes :

<code>msg_perm.uid</code>	L'uid du propriétaire de la file.
<code>msg_perm.gid</code>	Le gid du propriétaire de la file.
<code>msg_perm.mode</code>	Le mode d'accès à la file.

<code>msg_stime</code>	L'heure du dernier message envoyé à la file.
<code>msg_rtime</code>	L'heure du dernier message émis par la file.
<code>msg_ctime</code>	L'heure de modification de la file.
<code>msg_qnum</code>	Le nombre de messages en attente dans la file.
<code>msg_qbytes</code>	Le nombre d'octets actuellement disponible dans la file pour stocker des messages jusqu'à leur lecture.
<code>msg_lspid</code>	Le pid du processus qui a envoyé le dernier message à la file.
<code>msg_lrpid</code>	Le pid du processus qui a reçu le dernier message de la file.

Voir aussi [msg_remove_queue](#) , [msg_receive](#) , [msg_stat_queue](#) et [msg_set_queue](#) .

7.93.14 `sem_acquire()` : Réserve un sémaphore

```
bool sem_acquire ( int sem_identifiant )
```

[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.

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](#) .

7.93.15 `sem_get()` : Retourne un identifiant de sémaphore

```
int sem_get ( int key , int max_acquire , int perm )
```

[sem_get](#) retourne un identifiant qui pourra être utilisé pour accéder à un sémaphore System V. Le sémaphore est créé, si nécessaire, en utilisant les bits de permission (par défaut, 0666). Le nombre de processus qui peuvent réserver simultanément le sémaphore est précisé dans `max_acquire` (par défaut à 1). Actuellement, cette valeur n'est affectée que si le processus est le seul processus actuellement attaché au sémaphore.

[sem_get](#) retourne un identifiant positif 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](#) .

Note
sem_get n'est pas disponible sous Windows.

7.93.16 `sem_release()` : Libère un sémaphore

```
bool sem_release ( int sem_identifiant )
```

[sem_release](#) libère le sémaphore 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 .

Note

Cette fonction n'est pas disponible sous Windows.

7.93.17 sem_remove() : Détruit un sémaphore

```
bool sem_remove ( int sem_identifiant )
```

sem_remove supprime le sémaphore sem_identifiant si 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 .

Note

Cette fonction n'est pas disponible sous Windows. Elle a été ajoutée en PHP 4.1.0.

7.93.18 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 accèdera toujours à la même portion de mémoire partagée. Dans ce cas, memsize et perm seront ignorés.

Voir aussi ftok .

Note

Cette fonction n'est pas disponible sous Windows.

7.93.19 shm_detach() : Libère un segment de mémoire partagée

```
bool shm_detach ( int shm_identifiant )
```

shm_detach libère le segment de mémoire partagée identifié par shm_identifiant 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` .

7.93.20 shm_get_var() : Lit une variable dans la mémoire partagée.

```
mixed shm_get_var ( int id , int variable_key )
```

shm_get_var retourne la variable repérée par `variable_key`. La variable est toujours présente en mémoire partagée.

Note

<u>shm_get_var</u> n'est pas disponible sous Windows.

7.93.21 shm_put_var() : Insère ou modifie une variable de la mémoire partagée.

```
int shm_put_var ( int shm_identifiant , int variable_key , mixed variable )
```

shm_put_var insère ou modifie la variable `variable` avec la clé `variable_key`. Tous les types de variables sont supportés.

Note

<u>shm_put_var</u> n'est pas disponibles sous Windows.

7.93.22 shm_remove_var() : Efface une variable de la mémoire partagée.

```
int shm_remove_var ( int id , int variable_key )
```

shm_remove_var efface la variable `variable_key` de la mémoire partagée et libère la mémoire.

Note

<u>shm_remove_var</u> n'est pas disponible sous Windows.

7.93.23 shm_remove() : Supprime un segment de mémoire partagée sous Unix.

```
int shm_remove ( int shm_identifiant )
```

shm_remove supprime un segment de mémoire partagée sous Unix. Toutes les données seront supprimées.

Note

<u>shm_remove</u> n'est pas disponibles sous Windows.

7.94 SESAM

7.94.1 Introduction

SESAM/SQL-Server est une base de données mainframe, développée par Fujitsu Siemens Computers, Allemagne. Elle fonctionne sur les serveur 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.

7.94.2 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 librairie BS2000 PLAM contenant le module du pilote SESAM. Ceci est obligatoire pour utiliser les fonctions SESAM. La librairie 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 file 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.

7.94.3 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 librairie BS2000 PLAM qui contient les modules pilotes chargeable SESAM. Il 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
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 rapide, car il 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 librairie OML PLAM. Dans la configuration, vous devez indiquer à PHP le nom de la librairie PALM, et le fichier de lien à utiliser pour la configuration de SESAM (En SESAM V3.0, SQLSCI est disponible dans la librairie 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 interprété 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](#).

7.94.4 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 effectuée 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 librairie 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.

7.94.5 Types de curseurs

Les curseurs de résultat sont alloués pour les requêtes SQL de sélection, 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.

7.94.6 Portage

En PHP, il est naturel de commencer les index à zéro (plutôt que 1), et quelques adaptations ont été faite 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 colonne (exclus) du résultat (`$array["count"]`). Lors du portage d'applications depuis d'autres langage évolués vers le 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.

7.94.7 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.

7.94.8 Migration d'une autre base SQL

Deux langage SQL ne sont jamais 100% compatibles. Lorsque vous portez une application SQL depuis une autre interface vers SESAM, certaines adaptation 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 par 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 date disponible 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 si il y a des objets dépendant (par exemple, des vues), tandis qu'avec `CASCADE`, les objets dépendants seront supprimés en même temps que la table.

7.94.9 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 l'existence des champs (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 représenté par des chaînes vides).

7.94.10 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 multiple" peut être créé comme ceci :

Création d'une colonne de champs multiples

```
CREATE TABLE multi_field_test (
  pkey CHAR(20) PRIMARY KEY,
  multi(3) CHAR(12)
)
```

et peut être remplie avec :

Affectation d'une colonne de type "champs multiple"

```
INSERT INTO multi_field_test (pkey, multi(2..3) )
VALUES ('Second', <'first_val', 'second_val'>)
```

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 colonne. 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.

7.94.11 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](#)

7.94.13 [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](#) .

7.94.14 [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 librairie 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.

Voir aussi `sesam_disconnect`.

Connexion à une base SESAM
<pre><?php if (! sesam_connect ("moncatalogue", "monschema", "toto") die("Impossible de se connecter à SESAM"); ?></pre>

7.94.15 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 :

Elément	Contenu
<code>\$array["sqlstate"]</code>	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
<pre><?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) { \$err = Sesam_Diagnostic(); \$colspan=4; // 4 colonnes pour : sqlstate, errlin, errcol, rowcount if (\$err["errlin"] == -1) --\$colspan; if (\$err["errcol"] == -1) --\$colspan; if (\$err["rowcount"] == 0) --\$colspan; echo "<TABLE BORDER>\n"; echo "<TR><TH COLSPAN=".\$colspan.">ERROR: ". htmlspecialchars(\$err["errmsg"])."</TH></TR>\n"; if (\$err["errcol"] >= 0) { echo "<TR><TD COLSPAN=".\$colspan."><PRE>\n"; \$errstmt = \$exec_str."\n"; for (\$lin=0; \$errstmt != ""; ++\$lin) { if (\$lin != \$err["errlin"]) { // \$lin is less or greater than errlin</pre>


```

        if (! ($i = strchr($errstmt, "\n")))
            $i = "";
        $line = substr($errstmt, 0, strlen($errstmt)-strlen($i)+1);
        $errstmt = substr($i, 1);
        if ($line != "\n")
            print htmlspecialchars($line);
    }
    else {
        if (! ($i = strchr($errstmt, "\n")))
            $i = "";
        $line = substr($errstmt, 0, strlen($errstmt)-strlen($i)+1);
        $errstmt = substr($i, 1);
        for ($col=0; $col < $err["errcol"]; ++$col)
            echo (substr($line, $col, 1) == "\t") ? "\t" : ".";
        echo "<FONT COLOR=RED><BLINK>\\</BLINK></FONT>\n";
        print "<FONT COLOR=\#880000>\>".htmlspecialchars($line)."</FONT>";
        for ($col=0; $col < $err["errcol"]; ++$col)
            echo (substr($line, $col, 1) == "\t") ? "\t" : ".";
        echo "<FONT COLOR=RED><BLINK></BLINK></FONT>\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.

7.94.16 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.

Déconnexion d'une base SESAM

```

<?php
if (sesam_connect ("moncatalogue", "monschema", "toto")) {
    //... quelques requêtes et d'autres trucs ...
}
sesam_disconnect();
}
?>

```

Voir aussi [sesam_connect](#) .

7.94.17 **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))
    printf("%s<BR>\n", sesam_errormsg());
?>
```

Voir aussi [sesam_diagnostic](#) pour la liste complète des états de requêtes SQL.

7.94.18 **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 variables 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 lecture ultérieure avec [sesam_affected_rows](#) .

Notez que [sesam_query](#) peut gérer les requêtes immédiates et les requêtes de selection. Utilisez [sesam_execimm](#) uniquement si vous connaissez le type de requête auparavant. Une tentative de requête de selection 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 matable VALUES('un', 'deux')";
$result = sesam_execimm($stmt);
$error = sesam_diagnostic();
print("sqlstate = ".$error["sqlstate"]."\n".
      "Nombre de lignes affectées = ".$error["rowcount"]." == ".
      sesam_affected_rows($result)."\n");
?>
```

Voir aussi [sesam_query](#) et [sesam_affected_rows](#) .

7.94.19 **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` si 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 leur 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 noms, 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
<pre>SELECT TBL1.COL AS F00, TBL2.COL AS BAR FROM TBL1, TBL2</programlisting></pre>

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 indices sont numérotés à partir de 1.

Lire les champs multiples de SESAM
<pre>CREATE TABLE ... (... MULTI(3) INT)</pre>

Les index associatifs utilisé 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 <code>sesam_fetch_array</code>
<pre><?php \$result = sesam_query ("SELECT * FROM phone\n". " WHERE LASTNAME='".strtoupper(\$name)."' \n". " ORDER BY FIRSTNAME", 1); if (! \$result) { ... error ... } // Affiche la table print "<TABLE BORDER>\n"; while ((\$row = sesam_fetch_array (\$result)) && count(\$row) > 0) { print " <TR>\n"; print " <TD>".htmlspecialchars(\$row["firstname"])."</TD>\n"; print " <TD>".htmlspecialchars(\$row["lastname"])."</TD>\n"; print " <TD>".htmlspecialchars(\$row["phoneno"])."</TD>\n"; print " </TR>\n"; } print "</TABLE>\n"; sesam_free_result (\$result); ?></pre>

Voir aussi `sesam_fetch_row` qui retourne un tableau numérique.

7.94.20 `sesam_fetch_result()` : Retourne tout ou partie d'un résultat SESAM

mixed **sesam_fetch_result** (string result_id , int max_rows)

`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
int \$arr["count"]	Nombre de colonnes dans le résultat (ou zéro si c'était une requête immédiate).
int \$arr["rows"]	Nombre de ligne dans le résultat (entre zéro et max_rows)
bool \$arr["truncated"]	TRUE si le nombre de ligne était d'au moins max_rows , FALSE sinon. Notez que même si cette valeur est à TRUE , le prochain appel à <code>sesam_fetch_result</code> peut retourner aucune ligne parce qu'il n'y a plus d'entrées.
mixed \$arr[col][row]	les valeurs du résultat à la ligne row et colonne col . Le résultat est un tableau multidimensionnel. row va de 0 à \$arr["rows"]-1 , et col de 0 à \$arr["count"]-1). Les champs peuvent être vides : vous devez vérifier leur existence avec la fonction <code>isset</code> . Le type retourné dépend du type SQL déclaré pour cette colonne (voir Introduction SESAM 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` .

7.94.21 `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 déplacé à SESAM SEEK NEXT)
1	SESAM SEEK PRIOR	Lecture séquentielle à rebours (après la lecture, la position est déplacé 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 placé 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

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 valeur par défaut seront utilisées (initialisées à : `SESAM SEEK NEXT` , et modifiée 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
print "<TABLE BORDER>\n";
$row = sesam_fetch_row ($result, SESAM_SEEK_LAST);
while (is_array($row)) {
    print "<TR>\n";
    for($col = 0; $col < $row["count"]; ++$col) {
        print " <TD>".htmlspecialchars($row[$col])."</TD>\n";
    }
    print "</TR>\n";
    // utilise la valeur implicite de SESAM_SEEK_PRIOR
    $row = sesam_fetch_row ($result);
}
print "</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.

7.94.22 `sesam_field_array()` : Retourne des informations sur une colonne

array `sesam_field_array` (string `result_id`)

`result_id` is a valid result id returned by `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 <code>\$arr["count"]</code>	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 <code>\$arr[col]["name"]</code>	Le nom de la colonne <code>col</code> , avec <code>col</code> qui vaut entre 0 et <code>\$arr["count"]-1</code> . 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 <code>\$arr[col]["count"]</code>	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 <code>sesam_query</code> pour avoir un exemple d'utilisation.
string <code>\$arr[col]["type"]</code>	Type de variable PHP pour les données de la colonne <code>col</code> , où <code>col</code> vaut de 0 à <code>\$arr["count"]-1</code> . La valeur retournée peut être l'une de celles-ci : "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 celle-ci : • "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](#) .

7.94.23 `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.

7.94.24 `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.

7.94.25 `sesam_num_fields()` : Retourne le nombre de colonne 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 compte autant que leur taille respective, c'est à dire qu'un champ multiple de trois colonnes compte comme

trois colonne.

Voir aussi [sesam_query](#) et [sesam_field_array](#) pour savoir distinguer les champs multiples des colonnes standard.

7.94.26 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 selection. 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 selection, un descripteur de résultat et un curseur (scrollable ou séquentiel, suivant le paramètre optionnel `scrollable` passé) sear 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](#) .

Voir aussi : [sesam_fetch_row](#) et [sesam_fetch_result](#) .

Liste toutes les lignes de table "phone" sous forme de table HTML
<pre> <?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>\n"; // Ajoute l'en-tête de titre comme nom de colonne if (\$cols = sesam_field_array(\$result)) { echo " <TR><TH COLSPAN=". \$cols["count"]. ">Result:</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"; } } } </pre>

```

    echo " </TR>\n";
  }
} while ($ok["truncated"]); // tant qu'il a y encore des données
echo "</TABLE>\n";
// libère les ressources
sesam_free_result($result);
?>

```

7.94.27 `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 résultat 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`.

7.94.28 `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 déplacé à SESAM SEEK NEXT)
1	SESAM SEEK PRIOR	Lecture séquentielle à rebours (après la lecture, la position est déplacé 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 placé 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`.

7.94.29 `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") fixée par le fichier de configuration SESAM), afin d'optimiser les requêtes ultérieures et garantir la cohérence de la base. Ces valeurs ne sont 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.

Modifier les paramètres de configuration SESAM
<pre><?php sesam_settransaction(SESAM_TXISOL_REPEATABLE_READ, SESAM_TXREAD_READONLY); ?></pre>

7.95 Sessions

7.95.1 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'attractivité de votre site.

Si vous êtes familier avec la gestion de session de la librairie PHPLIB, vous remarquerez que certains concepts sont similaires.

Chaque visiteur accédant à votre page web se voit assigné 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 activé¹) ou sur demande (explicitement avec `session_start` ou implicitement avec `session_register`) si il existe une session du même nom. Si c'est le cas, l'environnement précédemment sauvé sera recréé.

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.

Note

La gestion des sessions a été ajoutée en PHP 4.0.

7.95.2 Sessions et sécurité

Lien externe : [Session fixation](#)

Utiliser les sessions ne signifie pas que les données de session ne pourront être vue 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.

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'authentification.

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 par exemple 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.

7.95.3 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.95.4 Installation

Le support des sessions est activé par défaut. Si vous souhaitez exclure le support des sessions 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 librairie supplémentaire pour disposer de ces fonctions.

7.95.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>session.save_path</code>	<code>"/tmp"</code>	PHP_INI_ALL
<code>session.name</code>	<code>"PHPSESSID"</code>	PHP_INI_ALL
<code>session.save_handler</code>	<code>"files"</code>	PHP_INI_ALL
<code>session.auto_start</code>	<code>"0"</code>	PHP_INI_ALL
<code>session.gc_probability</code>	<code>"1"</code>	PHP_INI_ALL
<code>session.gc_maxlifetime</code>	<code>"1440"</code>	PHP_INI_ALL
<code>session.serialize_handler</code>	<code>"php"</code>	PHP_INI_ALL
<code>session.cookie_lifetime</code>	<code>"0"</code>	PHP_INI_ALL
<code>session.cookie_path</code>	<code>"/"</code>	PHP_INI_ALL
<code>session.cookie_domain</code>	<code>""</code>	PHP_INI_ALL
<code>session.cookie_secure</code>	<code>""</code>	PHP_INI_ALL
<code>session.use_cookies</code>	<code>"1"</code>	PHP_INI_ALL
<code>session.use_only_cookies</code>	<code>"0"</code>	PHP_INI_ALL
<code>session.referer_check</code>	<code>""</code>	PHP_INI_ALL
<code>session.entropy_file</code>	<code>""</code>	PHP_INI_ALL
<code>session.entropy_length</code>	<code>"0"</code>	PHP_INI_ALL
<code>session.cache_limiter</code>	<code>"nocache"</code>	PHP_INI_ALL
<code>session.cache_expire</code>	<code>"180"</code>	PHP_INI_ALL
<code>session.use_trans_sid</code>	<code>"0"</code>	PHP_INI_SYSTEM PHP_INI_PERDIR
<code>session.bug_compat_42</code>	<code>"1"</code>	PHP_INI_SYSTEM
<code>session.bug_compat_warn</code>	<code>"1"</code>	PHP_INI_ALL
<code>url_rewriter.tags</code>	<code>"a=href,area=href,frame=src,input=src,form=fakeentry"</code>	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

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 de session. Par défaut, c'est le système intégré par fichier : `files`.

`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 fichier), cet argument sera utilisé comme dossier de sauvegarde des sessions. Par défaut, il vaut `/tmp`. Si le niveau d'imbrication de `session.save_path` est supérieur à 2, le nettoyage des sessions obsolètes ne s'opérera pas.

Attention

Si vous laissez cette option configurée avec un dossier accessible en lecture à tout le monde, comme par exemple `/tmp` (par défaut), les autres utilisateurs pourront exploiter ces sessions en obtenant la liste de fichiers dans ce dossier.

Note

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, comme 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 alphanumérique. Par défaut, c'est `PHPSESSID`.

`session.auto_start` integer

Spécifie si le module de session doit démarrer automatiquement, au début de chaque script PHP. Par défaut, c'est `0` (désactivé).

`session.cookie_lifetime` integer

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`.

`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` integer

Specifies the probability that the gc (garbage collection) routine is started on each request in percent. Defaults to `1`.

`session.gc_maxlifetime` integer

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 vous utilisez le gestionnaire de session par fichier, qui est fourni par défaut votre système doit garder la trace des date de dernier accès aux fichier (atime). La FAT de Windows ne le fait pas, alors vous faudra trouver un autre système pour gérer les sessions qui ont expirées.

`session.referer_check` integer

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 trouvé, l'identifiant de session sera considéré comme invalide. Par défaut, cette option est la 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` integer

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` integer

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` integer

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 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_path` string

Spécifie le chemin utilisé lors de la création du cookie. Par défaut, il vaut `'`.

`session.cookie_domain` string

Spécifie le domaine utilisé lors de la création du cookie. Par défaut, il ne vaut rien.

`session.cookie_secure` bool

Spécifie que les cookies ne doivent être émis que sur des connexion 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`.

`session.cache_expire` integer

Spécifie la durée de vie des données de sessions, en minute. Cette option n'a aucune conséquence sur le contrôle de cache. Par défaut, il vaut ¹⁸⁰ (3 heures).

`session.use_trans_sid` integer

Note

En PHP 4.1.2 ou plus ancien, cette option est activée en utilisant l'option de compilation `--enable-trans-sid`. Depuis PHP 4.2.0, `trans-sid` est toujours compilée.

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 4.2.0 disposait d'une fonctionnalité/bug 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`.

`session.bug_compat_warn` boolean

Les versions de ^{PHP} antérieures à la version 4.2.0 disposait d'une fonctionnalité/bug 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.url_rewriter.tags` string

Spécifie quels 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`.

Les options `track_vars` et `register_globals` influencent le comportement des sessions, leur stockage et leur restauration.

Note

Depuis PHP 4.0.3, `track_vars` est toujours activé.

7.95.6 Types de ressources

Cette extension ne définit aucune ressource.

7.95.7 Constantes prédefinies

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".

7.95.8 Exemples

Note
Depuis PHP 4.1.0, <code>\$_SESSION</code> est disponible comme variable globale, au même titre que <code>\$_POST</code>, <code>\$_GET</code>, <code>\$_REQUEST</code>, etc.. Contrairement à <code>\$HTTP_SESSION_VARS</code>, <code>\$_SESSION</code> est toujours globale. Par conséquent, vous n'avez pas besoin d'utiliser le mot réservé <code>global</code> avec <code>\$_SESSION</code>. Notez que cette documentation a été modifiée pour utiliser <code>\$_SESSION</code>. Vous pouvez toujours le remplacer par <code>\$HTTP_SESSION_VARS</code> si vous préférez l'ancienne version. Les clés du tableau <code>\$_SESSION</code> 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 <code>'_'</code>. 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 maintenable. 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 <code>\$_SESSION</code> .
<pre><?php session_start(); // Utilisez \$HTTP_SESSION_VARS avec PHP 4.0.6 ou plus ancien if (!isset(\$_SESSION['count'])) { \$_SESSION['compteur'] = 0; } else { \$_SESSION['compteur']++; } ?></pre>

Retirer une variable de session avec <code>\$_SESSION</code> et <code>register_globals</code> inactif.
<pre><?php session_start(); // Utilisez \$HTTP_SESSION_VARS avec PHP 4.0.6 ou plus ancien unset(\$_SESSION['compteur']); ?></pre>

Retirer une variable de session avec <code>\$_SESSION</code> et <code>register_globals</code> activé, après l'avoir enregistré avec <code>\$_SESSION</code> .
<pre><?php session_start(); // Avec PHP 4.3 et plus récent, vous pouvez simplement utiliser l'exemple précédent session_unregister('compteur'); ?></pre>

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

Si vous utilisez `$HTTP_SESSION_VARS` et `$_SESSION` et désactivez `register_globals`, n'utilisez pas `session_register`, `session_is_registered` et `session_unregister`.

Si vous activez `register_globals`, `session_unregister` doit être utilisé, car les variables de session sont enregistrés comme variables globales lorsque les données de sessions sont relues. Désactiver `register_globals` est recommandé pour des raisons de sécurité et de performances.

Enregistrer une variable avec `register_globals` activé

```
<?php
if (!session_is_registered('compteur')) {
    session_register("compteur");
    $compteur = 0;
}
else {
    $compteur++;
}
?>
```

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).

De plus, si vous enregistrez une nouvelle variable avec la fonction `session_register`, l'entrée dans l'environnement globale et `$_SESSION` ne fera pas de référence vers la même valeur jusqu'à la prochaine utilisation de `session_start` (ceci s'applique à PHP 4.2 et avant seulement). C'est à dire qu'une modification dans les variables globales ne seront pas répercutés dans les entrées de `$_SESSION`. Il est peu probable que cela ait un impact en pratique, et de plus, cela a été corrigé en PHP 4.3.

7.95.9 Passer l'identifiant de session (session ID)

Il y a deux méthodes de propagation de l'identifiant de session :

- Cookies
- Paramètre URL

The session module supports both methods. Cookies are optimal, but since they are not reliable (clients are not bound to accept them), we cannot rely on them. The second method embeds the session id directly into URLs.

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.

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 hits d'un utilisateur sur une page

```
<?php
if (!session_is_registered('compteur')) {
    session_register('compteur');
    $compteur = 1;
}
else {
    $compteur++;
}
?>
```

Bonjour visiteur, vous avez vu cette page <?php echo \$compteur; ?> fois.<p>;

Pour continuer, <A HREF="nextpage.php?<?php echo SID?>">cliquez ici.

<?php echo SID?> (ou bien <?=SID?> peut être utilisé si les short open tag sont actifs) est nécessaire pour préserver l'identifiant de session dans le cas où l'utilisateur a désactivé les cookies. <? echo SID?> n'est pas nécessaire, si `--enable-trans-sid` a été activé à la compilation de PHP.

Note

Les URLs non relatives sont considérées comme externes au site, et ne recevront pas le SID, car c'est une fuite d'information vers un autre site (envoi d'informations importantes).

7.95.10 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 prédefinies](#)
- [Exemples](#)
- [Passer l'identifiant de session \(session ID\)](#)
- [Gestion personnalisée des sessions](#)
- [session_cache_expire](#)
- [session_cache_limiter](#)
- [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](#)

7.95.12 [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
<code>private_no_expire</code> a été ajouté en <code>PHP 4.2.0dev</code> .

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
<pre><?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<p>"; ?></pre>

Voir aussi la directive de configuration [session.cache_limiter](#).

7.95.13 [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](#)

7.95.14 session_destroy() : Détruit une session

`bool session_destroy ( void )`

session_destroy détruit la session courante et toutes les données associées avec cette session. Aucune des variables contenant les données de la session ne seront détruite, mais celles écrites sur le serveur le seront. L'identifiant de session est libéré, et les prochains accès à cette session ne fourniront pas les valeurs actuelles. Le cookie de session n'est pas supprimé immédiatement, mais dès la prochaine page.

session_destroy retourne `TRUE` en cas de succès, et `FALSE` sinon.

Destruction d'une session avec session_destroy

```
<?php
// Initialisation de la session.
// Si vous utilisez un autre nom
// session_name("something")
session_start();
// Détruit toutes les variables de session
session_unset();
// Finalement, détruit la session
session_destroy();
?>
```

Destruction d'une session avec `$_SESSION`

```
<?php
// Initialisation de la session.
// Si vous utilisez un autre nom
// session_name("something")
session_start();
// Détruit toutes les variables de session
$_SESSION = array();
// Finalement, détruit la session
session_destroy();
?>
```

7.95.15 session_encode() : Encode les données de session

`string session_encode ( void )`

session_encode retourne une chaîne contenant les variables de la session courante encodé, prêt au stockage.

Voir aussi session_decode

7.95.16 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](#) .

7.95.17 session_id() : Let et/ou modifie l'identifiant courant de session

string **session_id** (string id)

[session_id](#) retourne l'identifiant courant de session.

Si ^{id} est fourni, il remplacera l'identifiant courant de session. [session_id](#) doit alors être appelé avant [session_start](#) . Suivant le gestionnaire de session que vous utilisez, tous les caractères ne seront pas acceptés dans cette valeur. Par exemple, le gestionnaire de session par défaut, basé sur les fichiers, n'accepte que les lettres et chiffres!

La constante ^{SID} peut alors être utilisée pour lire le nom de la session courante, et l'identifiant de session à fournir dans les URL. Notez que ^{SID} est uniquement défini si le client n'a pas envoyé le bon cookie. Voyez aussi [Gestion de session](#) .

Voir aussi [session_start](#) , [session_set_save_handler](#) et [session.save_handler](#) .

7.95.18 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 `$_SESSION` (ou `$HTTP_SESSION_VARS` en PHP 4.0.6 et plus ancien) est utilisé, utilisez la fonction [isset](#) pour vérifier la présence de la variable dans la session : elle aura été enregistrée dans `$_SESSION` .

Attention

Si vous utilisez `$_SESSION` (ou `$HTTP_SESSION_VARS`), n'utilisez pas [session_register](#) , [session_is_registered](#) et [session_unregister](#) .

7.95.19 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.

7.95.20 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 .

Exemple avec session_name

```
<?php
/* choisi le nom de session : MaSession */
$previous_name = session_name("MaSession");
echo "L'ancien nom de la session était $previous_name<p>";
?>
```

Voir aussi la directive de configuration session.name .

7.95.21 session_regenerate_id() : Remplace l'identifiant de session courant par un nouveau

bool **session_regenerate_id** ()

session_regenerate_id va remplacer l'identifiant de session courant par un nouveau, généré automatiquement, tout en conservant les valeurs de session.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec session_regenerate_id

```
<?php
session_start();
print session_id();
session_regenerate_id();
// This is now different
print session_id();
print_r($_SESSION);
?>
```

Note

Depuis ^{PHP} 4.3.3, si les cookies de session sont activés, l'utilisation de session_regenerate_id va aussi envoyer un cookie de session avec le nouveau numéro.

Voir aussi session_id , session_start et session_name .

7.95.22 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 enregistré est libre. Les noms peuvent être passé 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.

Attention

Si vous voulez que votre script fonctionne indépendamment de la configuration de la directive register_globals, vous devez utiliser la variable `$_SESSION`. Tous les éléments de `$_SESSION` sont automatiquement enregistrés. Si votre script utilise session_register, il ne fonctionnera pas dans les environnements où register_globals est désactivée.

Note

register_globals : note importante

Depuis PHP 4.2.0, la valeur par défaut de la directive de configuration PHP register_globals 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.

Attention

Cette fonction enregistre une variable **global**. Si vous enregistrer une variable globale dans une fonction, vous devez vous assurer de la rendre *globale*, avec `global`, ou bien le tableau `$GLOBALS[]`, ou bien utiliser un des tableaux de session ci-dessous.

Attention

Si vous utilisez `$_SESSION` (ou `$HTTP_SESSION_VARS`), n'utilisez pas session_register, session_is_registered et session_unregister.

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 ne sera fait.

Vous pouvez aussi créer une variable de session, simplement en ajoutant l'index approprié dans la variable `$_SESSION` ou `$HTTP_SESSION_VARS`.

Exemple avec session_register

```
<?php
$barney = "Un gros dinosaure violet.";
session_register("barney");

$_SESSION["zim"] = "Un envahisseur d'un autre planète.";

// L'ancienne méthode avec $HTTP_SESSION_VARS
$HTTP_SESSION_VARS["spongebob"] = "Il a un caleçon carré!";
?>
```

Note

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 *resource* dans leur définitions. Une liste de fonction qui retournent des ressources sont disponible dans l'annexe types de ressources.

Si `$_SESSION` (ou `$HTTP_SESSION_VARS` pour les versions antérieures à PHP 4.0.6) est utilisé, assignez les variables à `$_SESSION` : i.e. `$_SESSION['var'] = 'ABC'` ;

Voir aussi [session_is_registered](#) et [session_unregister](#) .

7.95.23 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, `reiserfs` peut se rendre plus efficace que `ext2fs`.

Voir aussi la directive de configuration [session.save_path](#) .

7.95.24 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 le script courant.

Note

Le paramètre `secure` 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](#) .

7.95.25 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 session, 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 que le ^{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.

Note

Le gestionnaire d'écriture n'est pas appelé s'il n'y a pas de données dans la session. Cela s'applique aussi si des variables de sessions vides sont enregistrées. Cela diffère du comportement du système de gestion des sessions par fichier, qui crée un fichier vide dans ce cas.

L'exemple suivant fournit un exemple de stockage des sessions semblable au système par défaut : par fichier. Cet exemple peut facilement être adapté pour réaliser des sauvegardes en bases 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.

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 devrez implémenter un
 * collecteur de données obsoletés 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` .

7.95.26 session_start() : Initialise une session

bool **session_start** (void)

`session_start` crée une session (ou restaure la session trouvée sur le serveur, via l'identifiant de session passé).

Si vous voulez utiliser une session nommée, vous devez appeler `session_name` avant `session_start` .

`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['favcouleur'] = '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 sessions. Lisez la référence sur les [sessions](#) pour des informations sur la [propagation des identifiants de sessions](#) , 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 utilisée 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`.

Voir aussi `session_id`.

7.95.27 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 récent) est utilisé, utilisez `unset` pour supprimer une variable de 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`.

7.95.28 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`), utilisez plutôt ceci : `$_SESSION = array();`

7.95.29 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 stocker les données.

Les données de session sont généralement stockés après 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, et ainsi, libérant les données pour les autres scripts.

7.96 Mémoire partagée

7.96.1 Introduction

Shmop est un ensemble de fonctions simples pour gérer la mémoire partagée avec ^{PHP} (lecture, écriture, création et suppressions de segments de mémoire partagée UNIX). Ces fonctions ne fonctionnent pas sous Windows, car ce système d'exploitation ne supporte pas la mémoire partagée. Toutefois, depuis Windows 2000, en activant la librairie `php_shmop.dll` dans votre fichier `php.ini` vous activerez aussi cette fonctionnalité.

Note

En PHP 4.0.3, ces fonctions étaient préfixées avec `shm` au lieu de `shmop`.

7.96.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.96.3 Installation

Pour utiliser les fonctions shmop, compilez ^{PHP} avec l'option `--enable-shmop`.

7.96.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.96.5 Types de ressources

7.96.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.96.7 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, "my shared memory block", 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 mis 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 prédefinies](#)
- [Exemples](#)
- [shmop_close](#)
- [shmop_delete](#)
- [shmop_open](#)
- [shmop_read](#)
- [shmop_size](#)
- [shmop_write](#)

7.96.9 shmop_delete() : Détruit un bloc de mémoire partagée

`int 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`.

7.96.10 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 `IPC_EXCL`) utilisez cette option pour ouvrir un bloc déjà existant.

- "c" pour création (utilise `IPC_CREATE`) utilisez cette option pour créer un nouveau bloc.
- "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 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_id = shmop_open(0x0fff, "c", 0644, 100);
?>
```

Cet exemple ouvre un nouveau bloc de mémoire partagée, dont l'identifiant est 0x0fff.

7.96.11 shmop_read() : Lit un bloc

string **shmop_read** (int shm_id , int start , int count)

`shmop_read` lit une chaîne dans un bloc de mémoire partagée.

`shmop_read` prend 3 paramètres: `shm_id`, 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 `$shm_data`.

7.96.12 shmop_size() : Lire la taille du bloc de mémoire partagée

int **shmop_size** (int shm_id)

`shmop_size` sert à connaître la taille, en octets, d'un bloc de mémoire partagée.

`shmop_size` prend comme argument `shm_id`, un identifiant de bloc de mémoire partagée créé par `shmop_open`, et retourne un entier, qui représente la 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 `$shm_id`, et le place dans `$shm_size`.

7.96.13 shmop_write() : Ecrire dans un bloc de mémoire partagée

```
int shmop_write ( int shm_id , string data , int offset )
```

`shmop_write` écrit une chaîne dans un bloc de mémoire partagée.

`shmop_write` prend 3 paramètres: `shm_id`, 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.

7.97 SQLite

7.97.1 Introduction

Cette extension fait l'interface avec le moteur de base `SQLite Embeddable SQL Database Engine`. `SQLite` est une librairie C qui implémente un moteur de base 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 librairie `SQLite` lit et écrit directement dans les fichiers de bases de données sur le disque.

Note
Pour plus d'informations sur <code>SQLite</code> , voyez le site web http://sqlite.org/ .

7.97.2 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 Windows peuvent télécharger la version DLL de l'extension `SQLite` ici : [php_sqlite.dll](#).

7.97.3 Contacts

Toutes les questions concernant cette extension doivent être envoyées aux listes de diffusion `PHP`, en anglais.

7.97.4 Pré-requis

Afin de faire disposer de ces fonctions, vous devez compiler `PHP` avec le support `SQLite`, ou charger dynamiquement l'extension `SQLite` depuis votre fichier `php.ini`.

7.97.5 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.

7.97.6 Constantes prédéfinies

Les fonctions `sqlite_fetch_array` et `sqlite_current` utilisent des constantes pour spécifier les différents types de résultats. Les constantes sont les suivantes :

Constante	Signification
<code>SQLITE_ASSOC</code>	Les colonnes sont retournées dans le tableau, en utilisant le nom de la colonne comme nom d'index.
<code>SQLITE_BOTH</code>	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	Les colonnes sont retournées dans le tableau, en utilisant une indexation numérique. L'index commence à 0, pour identifier le premier champ.

7.97.7 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
<code>sqlite.assoc_case</code>	0	PHP_INI_ALL

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

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 librairie `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 en majuscules ou minuscules, respectivement.

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](#)
- [Contacts](#)
- [Pré-requis](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [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_fetch_array](#)
- [sqlite_fetch_single](#)
- [sqlite_fetch_string](#)
- [sqlite_field_name](#)
- [sqlite_has_more](#)
- [sqlite_last_error](#)
- [sqlite_last_insert_rowid](#)

- [sqlite_libencoding](#)
- [sqlite_libversion](#)
- [sqlite_next](#)
- [sqlite_num_fields](#)
- [sqlite_num_rows](#)
- [sqlite_open](#)
- [sqlite_popen](#)
- [sqlite_query](#)
- [sqlite_rewind](#)
- [sqlite_seek](#)
- [sqlite_udf_decode_binary](#)
- [sqlite_udf_encode_binary](#)
- [sqlite_unbuffered_query](#)

7.97.9 [sqlite_busy_timeout\(\)](#) : Configure le délai d'attente d'une base SQLite occupée

```
void sqlite_busy_timeout ( resource dbhandle , int milliseconds )
```

[sqlite_busy_timeout](#) spécifie la durée maximale d'attente de SQLite, pour qu'une base de données ^{database} soit utilisable. La durée est exprimée en millisecondes. Si ^{milliseconds} vaut ⁰, les contraintes d'attentes seront supprimées, et SQLite va retourner immédiatement le code de statut ^{SQLITE_BUSY}, si un autre processus ou thread a verrouillé la base.

^{PHP} utilise une durée maximale d'attente de 60 secondes lorsque la base est ouverte.

Note
Il y a mille (1000) millisecondes dans une seconde.

Voir aussi [sqlite_open](#) .

7.97.10 [sqlite_changes\(\)](#) : Retourne le nombre de lignes qui ont été modifiées par la dernière requête SQLite

```
int sqlite_changes ( resource dbhandle )
```

[sqlite_changes](#) retourne le nombre de lignes qui ont été modifiées par la dernière requête SQLite, exécutée dans la base ^{db} .

Voir aussi [sqlite_num_rows](#) .

7.97.11 [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.

Voir aussi [sqlite_open](#) et [sqlite_popen](#) .

7.97.12 `sqlite_column()` : Lit la valeur d'une colonne dans un résultat SQLite

`mixed` **`sqlite_column`** (`resource` `result` , `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` . Le paramètre `decode_binary` fonctionne de la même façon que pour `sqlite_fetch_array` .

Utilisez cette fonction lorsque vous manipulez un grand résultat avec de nombreuses colonnes, ou des colonnes qui contiennent de grandes quantités de données.

Voir aussi `sqlite_fetch_string` .

7.97.13 `sqlite_create_aggregate()` : Enregistre une UDF agregeante pour les requêtes SQLite

`bool` **`sqlite_create_aggregate`** (`resource` `dbhandle` , `string` `function_name` , `mixed` `step_func` , `mixed` `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.

Exemple d'UDF SQLite : taille maximale de chaîne

```
<?php
$data = array(
    'one',
    'two',
    'three',
    'four',
    'five',
    'six',
    'seven',
    'eight',
    'nine',
    'ten'
);
$db = sqlite_open(':memory:');
sqlite_query($db, "CREATE TABLE strings(a)");
foreach ($data as $str) {
    sqlite_query($db, "INSERT INTO strings VALUES('" .
        sqlite_escape_string($str) . "')");
}

function max_len_step(&$context, $string) {
    if (strlen($string) > $context) {
        $context = strlen($string);
    }
}

function max_len_finalize(&$context) {
    return $context;
}

sqlite_create_aggregate($db, 'max_len', 'max_len_step', 'max_len_finalize');
var_dump(sqlite_array_query($db, 'SELECT max_len(a) from strings'));
?>
```

Dans cet exemple, nous créons une fonction d'agrégation qui va calculer la taille de la plus longue chaîne dans l'une des colonnes de la table. Pour chaque ligne, la fonction `max_len_step` est appelée, et reçoit le paramètre `context`. Le contexte est une autre variable ^{PHP}, et il peut contenir un tableau ou même un objet. Dans notre exemple, nous l'utilisons pour stocker la taille maximale de la colonne courante. Si `string` est plus grand que la valeur maximale que nous avons enregistré, nous modifions notre contexte en conséquence.

Une fois que toutes les lignes ont été traitées, SQLite appelle la fonction `max_len_finalize` pour déterminer le résultat de l'agrégation. Nous pourrions alors effectuer plusieurs calculs en fonction des données que nous avons récolté dans `context`. Dans notre exemple simple, nous avons calculé le résultat au fur et à mesure de la progression de la requête, ce qui nous permet de n'avoir qu'à retourner la valeur de notre contexte.

Note

L'exemple ci-dessus ne fonctionnera pas correctement si une colonne contient une donnée binaire. Reportez-vous à la fonction [sqlite_udf_decode_binary](#) pour plus de détails, et d'un exemple pour travailler avec des chaînes binaires.

Array Array

Voir aussi [sqlite_create_function](#), [sqlite_udf_encode_binary](#) et [sqlite_udf_decode_binary](#).

7.97.14 `sqlite_create_function()` : Enregistre une fonction utilisateur "classique" UDF pour SQLite

`bool sqlite_create_function ( resource dbhandle , string function_name , mixed 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.

`db` spécifie une ressource de base de données SQLite, que vous souhaitez améliorer. `function_name` spécifie le nom de la fonction que vous utiliserez dans les requêtes SQL. `callback` peut être n'importe quelle fonction de callback ^{PHP}. Le paramètre optionnel `num_args` est utilisé par l'analyseur d'expressions SQLite. Il est recommandé que vous fournissiez une valeur si votre fonction n'accepte qu'un nombre fixe d'arguments.

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.

Exemple avec [sqlite_create_function](#)

```
<?php
function md5_and_reverse($string) {
    return strrev(md5($string));
}

sqlite_create_function($db, 'md5rev', 'md5_and_reverse', 1);

$rows = sqlite_array_query($db, 'SELECT md5rev(filename) from files');
?>
```

Dans cet exemple, nous avons une fonction qui calcule la somme de contrôle MD5 d'une chaîne, et l'inverse. Lorsque la requête SQL s'exécute, elle retourne le nom du fichier, transformé par cette opération. Les données retournées dans `$rows` contient le résultat.

La beauté de cette technique est que vous n'avez pas à traiter vous même les données avec une

boucle, une fois que vous avez lu les données.

`PHP` enregistre une fonction spéciale appelée `php` lorsque la base est ouverte. Cette fonction est utilisée pour appeler toute fonction `PHP` sans avoir à l'enregistrer au préalable.

Exemple d'utilisation d'une fonction PHP dans SQLite

```
<?php
$rows = sqlite_array_query($db, "SELECT php('md5', filename) from files");
?>
```

Cet exemple va appeler la fonction `md5` sur chaque colonne `filename` dans la base, et retourner le contenu dans le tableau `$rows`.

Note

Pour des raisons de performances, `PHP` ne va pas automatiquement coder ou décoder les données binaires passées à votre UDF. Vous devez le faire manuellement. Voyez les fonctions `sqlite_udf_encode_binary` et `sqlite_udf_decode_binary` pour plus de détails.

Array Array

Voir aussi [sqlite_create_aggregate](#).

7.97.15 sqlite_current() : Lit une ligne de résultat SQLite dans un tableau

array **sqlite_current** (resource result , 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.

Si la position courante est au-delà de la dernière ligne, cette fonction retourne `FALSE`

Note

Cette fonction ne fonctionnera pas sur des résultats non bufferisés.

Voir aussi [sqlite_fetch_array](#), [sqlite_seek](#) et [sqlite_next](#).

7.97.16 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`.

Voir aussi [sqlite_last_error](#).

7.97.17 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

`addslashes` **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 `sqlite_create_function` ou `sqlite_create_aggregate` : utilisez plutôt `sqlite_udf_encode_binary`.

Voir aussi `sqlite_udf_encode_binary`.

7.97.18 `sqlite_fetch_array()` : Lit la prochaine ligne de résultat SQLite dans un tableau

```
array sqlite_fetch_array ( resource result , int result_type , bool decode_binary )
```

`sqlite_fetch_array` lit la prochaine ligne dans le résultat `result`. Si il n'y a pas d'autres lignes, `sqlite_fetch_array` retourne `FALSE`, et sinon, elle retourne un tableau représentant les données lues.

`result_type` peut être utilisé pour spécifier le type de résultat. La valeur par défaut est `SQLITE_BOTH`, dans lequel les colonnes sont indexées avec leur nom de colonne et leur index dans le résultat. `SQLITE_ASSOC` fait que le tableau sera indexé uniquement avec les noms de colonnes, et `SQLITE_NUM` sera indexé uniquement avec les index numériques.

Les noms de colonnes retournés avec les options `SQLITE_ASSOC` et `SQLITE_BOTH` seront adaptés en fonction de l'option de configuration `sqlite.assoc_case`.

Lorsque `decode_binary` vaut `TRUE` (par défaut), `PHP` va décoder les données binaires, si elles ont été codées avec la fonction `sqlite_escape_string`. Vous allez généralement laisser cette valeur à sa valeur par défaut, à moins que vous ne travaillez avec des bases opérées par d'autres applications.

Voir aussi `sqlite_array_query` et `sqlite_fetch_string`.

7.97.19 `sqlite_fetch_single()` : Lit la première ligne d'un résultat SQLite sous forme de chaîne

```
string sqlite_fetch_single ( resource result , int result_type , 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.

Exemple avec [sqlite_fetch_single](#)

```
<?php
if ($db = sqlite_open('mysqldb', 0666, $sqliteerror)) {

    $res = sqlite_query($db, "SELECT id FROM sometable WHERE id = 42");

    if (sqlite_num_rows($res) > 0) {
        echo sqlite_fetch_single($res); // 42
    }

    sqlite_close($db);
}
?>
```

Voir aussi [sqlite_fetch_array](#) .

7.97.20 [sqlite_fetch_string\(\)](#) : Alias de [sqlite_fetch_single](#)

Cette fonction est un alias de [sqlite_fetch_single](#) .

7.97.21 [sqlite_field_name\(\)](#) : Retourne le nom du champ SQLite

string [sqlite_field_name](#) (resource result , int field_index)

[sqlite_field_name](#) retourne le nom du champ identifié par son index numérique `field_index` , dans le résultat `result` .

7.97.22 [sqlite_has_more\(\)](#) : Indique si il reste des lignes SQLite à lire

bool [sqlite_has_more](#) (resource result)

[sqlite_has_more](#) retourne `TRUE` si il y a d'autres lignes de résultat à lire dans la ressource `result` et `FALSE` sinon.

Voir aussi [sqlite_num_rows](#) et [sqlite_changes](#) .

7.97.23 [sqlite_last_error\(\)](#) : Retourne le dernier code d'erreur SQLite

int [sqlite_last_error](#) (resource dbhandle)

[sqlite_last_error](#) retourne le code d'erreur de la dernière opération SQL effectuée sur la base `dbhandle` .

Voir aussi [sqlite_error_string](#) .

7.97.24 `sqlite_last_insert_rowid()` : Retourne le numéro de ligne de la dernière ligne insérée

`int sqlite_last_insert_rowid ( resource dbhandle )`

`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

7.97.25 `sqlite_libencoding()` : Retourne l'encodage utilisé par la librairie 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`.

Voir aussi `sqlite_libversion`.

7.97.26 `sqlite_libversion()` : Retourne la version de la librairie SQLite

`string sqlite_libversion ( void )`

`sqlite_libversion` retourne la version de la librairie SQLite, sous la forme d'une chaîne.

Voir aussi `sqlite_libencoding`.

7.97.27 `sqlite_next()` : Déplace le pointeur SQLite vers la prochaine ligne

`bool sqlite_next ( resource result )`

`sqlite_next` avance le pointeur de résultat à la prochaine ligne du résultat `result`. `sqlite_next` retourne `FALSE` si il n'y a plus de lignes, et `TRUE` sinon.

Note

Cette fonction ne peut utiliser de résultats non bufferisés.

Voir aussi [sqlite_seek](#) , [sqlite_current](#) et [sqlite_rewind](#) .

7.97.28 `sqlite_num_fields()` : Retourne le nombre de champs dans un résultat SQLite

```
int sqlite_num_fields ( resource result )
```

[sqlite_num_fields](#) retourne le nombre de champs dans le résultat SQLite `result` .

Voir aussi [sqlite_column](#) et [sqlite_num_rows](#) .

7.97.29 `sqlite_num_rows()` : Retourne le nombre de lignes d'un résultat SQLite

```
int sqlite_num_rows ( resource result )
```

[sqlite_num_rows](#) retourne le nombre de lignes dans le résultat SQLite `result` .

Note

Cette fonction ne fonctionne pas avec les résultats non bufferisés.

Voir aussi [sqlite_changes](#) et [sqlite_query](#) .

7.97.30 `sqlite_open()` : Ouvre une base SQLite

```
resource sqlite_open ( string filename , int mode , string &error_message )
```

[sqlite_open](#) retourne une ressource de connexion SQLite, ou `FALSE` en cas d'erreur.

Le paramètre `filename` est le nom de la base. Cela peut être un nom relatif ou absolu, que SQLite va utiliser pour stocker vos données. Si le fichier n'existe pas, SQLite va tenter de le créer. Vous **DEVEZ** avoir les droits d'écriture si vous voulez faire des insertions de données, ou modifier votre schéma de base.

Le paramètre `mode` spécifie le mode d'ouverture du fichier, et il permet notamment d'ouvrir la base en mode lecture seule. Actuellement, ce paramètre est ignoré par la librairie SQLite. La valeur par défaut pour ce paramètre est la valeur octale `0666` . C'est la valeur recommandée si vous avez besoin d'utiliser le paramètre `errmessage` .

`errmessage` est passé par référence, et il contient un message d'erreur descriptif, expliquant pourquoi la base n'a pu être ouverte.

Exemple avec [sqlite_open](#)

```
<?php
if ($db = sqlite_open('mysqlitedb', 0666, $sqliteerror)) {
    sqlite_query($db,'CREATE TABLE foo (bar varchar(10))');
    sqlite_query($db,"INSERT INTO foo VALUES ('fnord')");
    $result = sqlite_query($db,'select bar from foo');
    var_dump(sqlite_fetch_array($result));
} else {
    die ($sqliteerror);
}
```

```
}
?>
```

Array Array

Note

Depuis SQLite version 2.8.2, vous pouvez spécifier `:memory:` comme valeur de `filename` pour créer une base qui sera installée en mémoire vive, sur le serveur. C'est très pratique pour des traitements temporaires, car la base en mémoire sera détruite dès que le script sera terminé. Cela peut aussi être utile, lorsqu'utilisé conjointement avec la commande SQL `ATTACH DATABASE`, pour charger d'autres bases, et faire des requêtes liées.

Note

SQLite prend en compte le safe mode et l'option `open_basedir`.

Voir aussi [sqlite_popen](#), [sqlite_close](#) et [sqlite_query](#).

7.97.31 sqlite_popen() : Ouvre une connexion SQLite persistante

```
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 que vos processus parallèles ouvrent les bases et fassent leurs modifications dans une transaction.

Voir aussi [sqlite_open](#), [sqlite_close](#) et [sqlite_query](#).

7.97.32 sqlite_query() : Exécute une requête SQLite et lit le résultat

```
resource sqlite_query ( resource dbhandle , string query ) resource sqlite_query ( string query , resource dbhandle )
```

[sqlite_query](#) exécute la requête SQL `query` dans la base SQLite `db`.

Pour les requêtes qui retournent des lignes, cette fonction va retourner une ressource que vous pourrez utiliser avec des fonctions telles que [sqlite_fetch_array](#) et [sqlite_seek](#).

Pour les autres types de requêtes, cette fonction va retourner un booléen, en guise de résultat. ^{TRUE} pour le succès, et ^{FALSE} pour l'échec.

Indépendamment du type, cette fonction retournera ^{FALSE} si la requête échoue.

[sqlite_query](#) retourne un résultat bufferisé. C'est très pratique pour des résultats de petites tailles, où vous aurez besoin d'accéder aléatoirement aux lignes. Les résultats bufferisés allouent la mémoire nécessaire pour stocker tout le résultat, et ne se termineront qu'une fois toutes ces données lues. Si vous n'avez besoin que d'un accès séquentiel aux données, il est recommandé d'utiliser [sqlite_unbuffered_query](#).

Note

Deux syntaxes alternatives sont supportées pour assurer la compatibilité avec les autres bases de données (telles que MySQL) : La forme recommandée est la première, où le paramètre ^{db} est le premier dans la fonction.

Attention

SQLite **va** exécuter les requêtes multiples, séparées par des points-virgules, ce qui vous permet d'exécuter plusieurs requêtes SQL en une seule fois.

Lorsque vous exécutez des requêtes multiples, la valeur retournée par cette fonction sera ^{FALSE} si une erreur s'est produite, mais indéfini dans les autres cas (cela peut être ^{TRUE} en cas de succès, ou bien elle peut retourner un résultat).

Voir aussi [sqlite_unbuffered_query](#) et [sqlite_array_query](#).

7.97.33 sqlite_rewind() : Place le pointeur de résultat SQLite au début

```
bool sqlite_rewind ( resource result )
```

[sqlite_rewind](#) remplace le pointeur à la première ligne du résultat ^{result}. [sqlite_rewind](#) retourne ^{FALSE} si il n'y a plus de lignes dans le résultat, et ^{TRUE} sinon.

Note

Cette fonction ne fonctionne pas avec les résultats non bufferisés.

Voir aussi [sqlite_next](#), [sqlite_current](#) et [sqlite_seek](#).

7.97.34 sqlite_seek() : Déplace le pointeur de résultat SQLite vers une ligne

```
bool sqlite_seek ( resource result , int rownum )
```

[sqlite_seek](#) déplace le pointeur de résultat à la ^{rownum}ème ligne. Le pointeur de lignes commence à 0. [sqlite_seek](#) retourne ^{FALSE} si la ligne n'existe pas, et ^{TRUE} sinon.

Note

Cette fonction ne peut pas être utilisée avec les résultats non-bufferisés.

Voir aussi [sqlite_next](#), [sqlite_current](#) et [sqlite_rewind](#).

7.97.35 `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.

Exemple de fonction d'agrégation SQLite, compatible avec les données binaires

```
<?php
$data = array(
    'one',
    'two',
    'three',
    'four',
    'five',
    'six',
    'seven',
    'eight',
    'nine',
    'ten'
);
$db = sqlite_open(':memory:');
sqlite_query($db, "CREATE TABLE strings(a)");
foreach ($data as $str) {
    sqlite_query($db, "INSERT INTO strings VALUES('" .
        sqlite_escape_string($str) . "')");
}

function max_len_step(&$context, $string) {
    $string = sqlite_udf_decode_binary($string);
    if (strlen($string) > $context) {
        $context = strlen($string);
    }
}

function max_len_finalize(&$context) {
    return $context;
}

sqlite_create_aggregate($db, 'max_len', 'max_len_step', 'max_len_finalize');
var_dump(sqlite_array_query($db, 'SELECT max_len(a) from strings'));
?>
```

Voir aussi [`sqlite\_udf\_encode\_binary`](#), [`sqlite\_create\_function`](#) et [`sqlite\_create\_aggregate`](#).

7.97.36 `sqlite_udf_encode_binary()` : Encode les données binaires avant de les retourner d'une UDF SQLite

`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 librairie ^{libsqlite} n'est pas compatible avec les données binaires).

Si il y a des chances que vos données ne soient pas compatibles, (comme par exemple, si elle contient 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 [sqlite_escape_string](#) pour protéger les données dans vos UDF, car cela conduira à doubler la protection. Utilisez plutôt cette fonction à la place.

Voir aussi: [sqlite_udf_decode_binary](#), [sqlite_escape_string](#), [sqlite_create_function](#) et [sqlite_create_aggregate](#).

7.97.37 `sqlite_unbuffered_query()` : Exécute une requête SQLite non bufferisée

```
resource sqlite_unbuffered_query ( resource dbhandle , string query ) resource sqlite_unbuffered_query (
string query , resource dbhandle )
```

[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 [sqlite_seek](#) et [sqlite_rewind](#) ne fonctionnent pas avec les résultats non bufferisés de cette fonction.

Voir aussi: [sqlite_query](#).

7.98 Shockwave Flash

7.98.1 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 librairie libswf ne propose pas de support pour Windows. Le développement de cette librairie 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 .

7.98.2 Pré-requis

Vous avez besoin de la librairie libswf pour compiler PHP avec le support de cette extension. Vous pouvez télécharger cette librairie [ftp://ftp.sgi.com/cgi/graphics/grafica/flash/](http://ftp.sgi.com/cgi/graphics/grafica/flash/) .

7.98.3 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.

7.98.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.98.5 Types de ressources

Cette extension ne définit aucune ressource.

7.98.6 Constantes prédefinies

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 (*nombre à virgule flottante*)
 BSOver (*nombre à virgule flottante*)
 BSUp (*nombre à virgule flottante*)
 OverDowntoIdle (*entier*)
 IdletoOverDown (*entier*)
 OutDowntoIdle (*entier*)
 OutDowntoOverDown (*entier*)
 OverDowntoOutDown (*entier*)
 OverUptoOverDown (*entier*)
 OverUptoIdle (*entier*)
 IdletoOverUp (*entier*)
 ButtonEnter (*entier*)
 ButtonExit (*entier*)
 MenuEnter (*entier*)
 MenuExit (*entier*)

7.98.7 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 prédéfinies](#)
- [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](#)

7.98.9 swf_actiongotoframe() : Joue un frame puis stoppe

```
void swf_actiongotoframe ( int framenumbers )
```

[swf_actiongotoframe](#) se déplace jusqu'au frame `framenumbers` , le joue, puis s'arrête.

7.98.10 swf_actiongotolabel() : Affiche le frame nommé.

```
void swf_actiongotolabel ( string label )
```

[swf_actiongotolabel](#) affiche le frame de nom `label` , puis stoppe.

7.98.11 swf_actionnextframe() : Avance d'un frame

```
void swf_actionnextframe ( void )
```

[swf_actionnextframe](#) avance d'un frame le frame courant.

7.98.12 swf_actionplay() : Joue l'animation flash à partir du frame courant.

```
void swf_actionplay ( void )
```

[swf_actionplay](#) joue l'animation Flash à partir du frame courant.

7.98.13 swf_actionprevframe() : Recule d'un frame

```
void swf_actionprevframe ( void )
```

swf_actionnextframe recule d'un frame le frame courant.

7.98.14 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.

7.98.15 swf_actionstop() : Arrête l'animation flash.

```
void swf_actionstop ( void )
```

swf_actionstop arrête l'animation Flash au frame courant.

7.98.16 swf_actiontogglequality() : Choisit le niveau de qualité haut ou bas.

```
void swf_actiontogglequality ( void )
```

swf_actiontogglequality modifie le niveau de qualité haut ou bas.

7.98.17 swf_actionwaitforframe() : Ignore les actions si le frame n'est pas chargé.

```
void swf_actionwaitforframe ( int framenumbr , int skipcount )
```

swf_actionwaitforframe vérifie que le frame `framenumbr` 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...".

7.98.18 swf_addbuttonrecord() : Contrôle la situation, l'apparance et la zone active du bouton courant

```
void swf_addbuttonrecord ( int states , int shapeid , int depth )
```

swf_addbuttonrecord permet de modifier les caractéristiques d'un bouton. `states`, définit les états du bouton autorisés : ce peut être : BSHitTest, BSDown, BSOVer ou BSUp. `shapeid` est l'apparance du bouton, c'est-à-dire l'objet qui représente le bouton. `depth` est la profondeur de placement du bouton, dans le frame 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 ();
?>
```


7.98.19 swf_addcolor() : Fixe la couleur globale d'addition (? : 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.

7.98.20 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);
}
?>
```

7.98.21 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.

7.98.22 swf_definefont() : Définit une police.

```
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.

7.98.23 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`.

7.98.24 swf_definepoly() : Définit un polygone.

```
void swf_definepoly ( int objid , array coords , int npoints , float width )
```

swf_definepoly définit un polygone, dont les coordonnées des sommets sont placés dans le tableau `coords`). `npoints` est le nombre de points contenu dans le tableau `coords`. `width` est la largeur des bords du polygone. Si `width` vaut 0.0, le polygone sera rempli.

7.98.25 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.

7.98.26 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.

7.98.27 **swf_endbutton()** : Termine la définition du bouton courant.

`void swf_endbutton ( void )`

swf_endbutton termine la définition du bouton courant.

7.98.28 **swf_enddoaction()** : Termine l'action courante

`void swf_enddoaction ( void )`

swf_startdoaction termine l'action courante, démarrée par swf_startdoaction .

7.98.29 **swf_endshape()** : Complète la définition de la forme courante.

`void swf_endshape ( void )`

swf_endshape complète la définition de la forme courante.

7.98.30 **swf_endsymbol()** : Termine la définition de symbole

`void swf_endsymbol ( void )`

swf_endsymbol termine la définition de symble, qui a été commencée avec swf_startsymbol .

7.98.31 **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} .

7.98.32 **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 values créeront une inclinaison vers la droite, et les valeurs négatives, vers la gauche.

7.98.33 **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, et les valeurs négatives le réduisent.

7.98.34 `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 a 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.

7.98.35 `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.

7.98.36 `swf_getframe()` : Retourne le numéro de frame courant

int `swf_getframe` (void)

`swf_getframe` retourne le numéro de frame courant.

7.98.37 `swf_labelframe()` : Nomme le frame courant

void `swf_labelframe` (string `name`)

`swf_labelframe` donne le nom `name` au frame courant.

7.98.38 `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 des z de l'utilisateur.

7.98.39 `swf_modifyobject()` : Modifie un objet

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.

7.98.40 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 <code>rgba</code> peuvent être positives ou négatives.

7.98.41 swf_nextid() : Retourne le prochain identifiant d'objet libre

```
int swf_nextid ( void )
```

[swf_nextid](#) retourne le prochain identifiant d'objet libre.

7.98.42 swf_oncondition() : Décrit une 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 transition possibles, les suivantes sont destinées aux boutons de type TYPE_MENUBUTTON:

- IdletoOverUp
- OverUptoidle
- OverUptoOverDown
- OverDowntoOverUp
- IdletoOverDown
- OutDowntoidle
- MenuEnter (IdletoOverUp|IdletoOverDown)
- MenuExit (OverUptoidle|OverDowntoidle)

Pour les types TYPE_PUSHBUTTON voici les options :

- IdletoOverUp
- OverUptoidle
- OverUptoOverDown
- OverDowntoOverUp
- OverDowntoOutDown
- OutDowntoOverDown
- OutDowntoidle

- ButtonEnter (IdletoOverUp|OutDowntoOverDown)
- ButtonExit (OverUptoldle|OverDowntoOutDown)

7.98.43 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).

7.98.44 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.

7.98.45 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.

7.98.46 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 projection, car Flash ne dispose que d'une matrice à 2 dimensions. Certaines distorsions font vraiment tâche d'encre.

7.98.47 swf_placeobject() : Place un objet sur la scène

```
void swf_placeobject ( int objid , int depth )
```

swf_placeobject place l'objet `objid` dans le frame 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.

7.98.48 swf_polarview() : Définit le point de vue de l'utilisateur en coordonnées polaire.

```
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.

7.98.49 swf_popmatrix() : Dépile la matrice de transformation.

```
void swf_popmatrix ( void )
```

swf_popmatrix dépile la matrice de transformation.

7.98.50 swf_posround() : Active l'approximation des translation d'objets.

```
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).

7.98.51 swf_pushmatrix() : Empile la matrice de transformation courante dans la pile.

```
void swf_pushmatrix ( void )
```

swf_pushmatrix empile la matrice de transformation courante dans la pile.

7.98.52 swf_removeobject() : Enlève un objet

```
void swf_removeobject ( int depth )
```

swf_removeobject enlève l'objet situé à la profondeur `depth` de la scène.

7.98.53 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' (axe x), 'y' (axe y) ou 'z' (axe z).

7.98.54 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.

7.98.55 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}.

7.98.56 swf_setframe() : Fixe le frame courant

```
void swf_setframe ( int framenumbrer )
```

swf_setframe sélectionne le frame ^{framenumbrer} comme frame actif.

7.98.57 swf_shapearc() : Dessine une 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.

7.98.58 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}).

7.98.59 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 (x^3 , y^3) comme point final. La position finale devient alors la position courante.

7.98.60 `swf_shapefillbitmapclip()` : Choisit le mode de remplissage par texture.

```
void swf_shapefillbitmapclip ( int bitmapid )
```

Choisit le mode de remplissage par texture : les espaces vides seront remplis avec la bitmap ^{bitmapid} .

7.98.61 `swf_shapefillbitmaptile()` : Choisit le mode de remplissage par texture répétée.

```
void swf_shapefillbitmaptile ( int bitmapid )
```

Choisit 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).

7.98.62 `swf_shapefilloff()` : Désactive le remplissage.

```
void swf_shapefilloff ( void )
```

`swf_shapefilloff` désactive le remplissage pour la forme courante.

7.98.63 `swf_shapefillsolid()` : Fixe la couleur pour le style courant de remplissage.

```
void swf_shapefillsolid ( float r , float g , float b , float a )
```

`swf_shapefillsolid` fixe la couleur pour le style courant de remplissage à ^{rgba} .

7.98.64 `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.

7.98.65 `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).

7.98.66 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).

7.98.67 swf_showframe() : Affiche le frame courant

void **swf_showframe** (void)

swf_showframe affiche le frame courant.

7.98.68 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.

7.98.69 swf_startdoaction() : Commence la description d'une liste d'action pour la frame courante.

void **swf_startdoaction** (void)

swf_startdoaction commence la description d'une liste d'actions pour la frame courante. Cette fonction doit être appelée avant que les actions ne soient définies pour le cadre courant.

7.98.70 swf_startshape() : Commence une forme complexe

void **swf_startshape** (int objid)

swf_startshape commence une forme complexe, qui sera repéré par l'identifiant d'objet ^{objid} .

7.98.71 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.

7.98.72 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.

7.98.73 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.

7.98.74 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.

7.99 SNMP

7.99.1 Introduction

7.99.2 Pré-requis

Afin de pouvoir utiliser les fonctions ^{SNMP} sous Unix, vous aurez besoin d'installer le package UCD SNMP ou NET-SNMP. Sous Windows ces fonctions ne sont disponibles que sous NT, et pas sous Win95/98.

7.99.3 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 étrange 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 compiler 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}.

7.99.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.99.5 Types de ressources

7.99.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)

- [snmp_get_quick_print](#)
- [snmp_set_quick_print](#)
- [snmpget](#)
- [snmprealwalk](#)
- [snmpset](#)
- [snmpwalk](#)
- [snmpwalkoid](#)

7.99.8 `snmp_set_quick_print()` : Ecrit la valeur courante de l'option `quick_print` de la librairie UCD

`void snmp_set_quick_print ( bool quick_print )`

`snmp_set_quick_print` fixe la valeur de l'option `quick_print` de la librairie UCD ^{SNMP}. Lorsqu'elle a la valeur de (1), la librairie ^{SNMP} retournera des valeurs 'rapides'. Cela signifie que seule, la valeur sera retournée. Lorsqu'elle a la valeur de (0), la librairie va afficher d'autres informations (telles que l'adresse IP (IpAddress) ou OID). De plus, si `quick_print` n'est pas activée, la librairie 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 `snmp_set_quick_print`

```
<?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 librairie UCD ^{SNMP}. `snmp_set_quick_print` n'est pas disponible avec la librairie Windows ^{SNMP}.

7.99.9 `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 `snmpget`

```
<?php
$syscontact = snmpget("127.0.0.1", "public", "system.SysContact.0");
?>
```

7.99.8 `snmp_set_quick_print()` : Ecrit la valeur courante de l'option `quick_print` de la librairie UCD ^{SNMP}

7.99.10 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.

7.99.11 snmpset() : Envoie 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}.

7.99.12 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
for ($i=0; $i<count($a); $i++) {
    echo $a[$i];
}
?>
```

7.99.13 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";
}
?>
```

7.100 Socket

7.100.1 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 [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 alter ego 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 réutilisés, avec des adaptations mineures. [UNIX Socket FAQ](#) est un bon début.

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.

7.100.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.100.3 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 .

7.100.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.100.5 Types de ressources

Cette extension ne définit aucune ressource.

7.100.6 Constantes prédefinies

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*)
 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*)

7.100.7 Socket Errors

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 Windows32. 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 interrompu. Il est commun de supprimer les erreurs avec l'opérateur `@` et de traiter les erreurs avec la fonction `socket_last_error`, au niveau 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 le message d'erreur s'affichera en fonction de la configuration locale (`LC_MESSAGES`):

Warning – socket_bind() unable to bind address [98]: Die Adresse wird bereits verwendet

7.100.8 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 (($sock = socket_create (AF_INET, SOCK_STREAM, 0)) < 0) {
    echo "socket_create() a échoué : raison : " . socket_strerror ($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

Ce exemple illustre 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.exemple.com');

/* Cree une socket TCP/IP. */
$socket = socket_create (AF_INET, SOCK_STREAM, 0);
if ($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";
$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 prédefinies](#)
- [Socket Errors](#)
- [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_iovec_add](#)
- [socket_iovec_alloc](#)
- [socket_iovec_delete](#)
- [socket_iovec_fetch](#)
- [socket_iovec_free](#)
- [socket_iovec_set](#)
- [socket_last_error](#)
- [socket_listen](#)

- [socket_read](#)
- [socket_readv](#)
- [socket_recv](#)
- [socket_recvfrom](#)
- [socket_recvmsg](#)
- [socket_select](#)
- [socket_send](#)
- [socket_sendmsg](#)
- [socket_sendto](#)
- [socket_set_block](#)
- [socket_set_nonblock](#)
- [socket_set_option](#)
- [socket_shutdown](#)
- [socket_strerror](#)
- [socket_write](#)
- [socket_writev](#)

7.100.10 [socket_bind\(\)](#) : Lie un nom à une socket

`bool socket\_bind ( resource socket , string address , int port )`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_bind](#) lie le nom donné par `address` à la socket décrite par `socket` , qui doit être une ressource de socket valide, créée par [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_connect](#) , [socket_listen](#) , [socket_create](#) , [socket_last_error](#) et [socket_strerror](#) .

7.100.11 [socket_clear_error\(\)](#) : Efface toutes les erreurs précédemment générées par une socket

`void socket\_clear\_error ( resource socket )`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_clear_error](#) efface tous les codes d'erreurs 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'erreurs 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](#).

7.100.12 socket_close() : Ferme une socket

```
void socket_close ( resource socket )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_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](#).

7.100.13 socket_connect() : Crée une connexion sur une socket

```
bool socket_connect ( resource socket , string address , int port )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_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](#).

7.100.14 `socket_create_listen()` : Ouvre une socket sur un port pour accepter les connexions

resource **socket_create_listen** (int port , int backlog)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_create_listen` est destinée à simplifier la création de nouvelles sockets destinées 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` .

7.100.15 `socket_create_pair()` : Crée une paire de socket identiques et les stocke dans fds.

bool **socket_create_pair** (int domain , int type , int protocol , array &fd)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.16 `socket_create()` : Crée une socket (point de communication)

resource **socket_create** (int domain , int type , int protocol)

`socket_create` crée un point de communication (une socket) et retourne une ressource de socket.

Le paramètre `domain` sélectionne le domaine (famille de protocole) à utiliser pour les communications. Actuellement, `AF_INET` et `AF_UNIX` sont les valeurs possibles de ce paramètre. `AF_INET` est typiquement

utilisé pour les communications Internet. `AF_UNIX` utilise des chemins systèmes pour identifier les sockets, et par conséquent, peut être uniquement utilisé pour les communications locales (mais elles seront aussi plus rapides).

Le paramètre `type` sélectionne le type de socket. Le type est représenté par les constantes suivantes : `SOCK_STREAM`, `SOCK_DGRAM`, `SOCK_SEQPACKET`, `SOCK_RAW`, `SOCK_RDM` ou `SOCK_PACKET`. Les deux types les plus courants sont `SOCK_DGRAM` pour le protocole UDP (connectionless) et `SOCK_STREAM` pour le protocole TCP.

Le paramètre `protocol` sélectionne le protocole qui peut être soit `SOL_UDP`, soit `SOL_TCP`.

La fonction `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.

Pour plus d'informations sur l'utilisation de `socket_create`, et sur les significations des différents paramètres, reportez-vous au manuel Unix sur les sockets (2).

Note

Si le paramètre `domain` ou `type` est donné, `socket_create` utilisera par défaut les valeurs `AF_INET` et `SOCK_STREAM`, respectivement. De plus, elle générera un message d'alerte (`E_WARNING`).

Voir aussi `socket_accept`, `socket_bind`, `socket_connect`, `socket_listen`, `socket_last_error`, et `socket_strerror`.

7.100.17 socket_get_option() : Lit les options de socket

`mixed socket_get_option ( resource socket , int level , int optname )`

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

Note

Cette fonction était appelée `socket_getopt()` avant PHP 4.3.0.

7.100.18 socket_getpeername() : Interroge l'autre extrémité de la communication

`bool socket_getpeername ( resource socket , string &addr , int &port )`

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 la socket `socket` est de type `AF_INET`, `socket_getpeername` retournera l'adresse IP de l'hôte, en notation numérique (e.g. `127.0.0.1`) 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`.

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` 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`.

7.100.19 socket_getsockname() : Interroge la socket locale

```
bool socket_getsockname ( resource socket, string &addr, int &port )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 la socket `socket` est de type `AF_INET`, `socket_getsockname` retournera l'adresse IP locale, en notation numérique (e.g. `127.0.0.1`) 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`.

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_UNIX`, auquel cas le dernier code d'erreur socket n'est **pas** modifié.

Voir aussi `socket_getpeername`, `socket_last_error` et `socket_strerror`.

7.100.20 socket_iovec_add() : Ajoute un nouveau vecteur dans le tableau

```
bool socket_iovec_add ( resource iovec, int iov_len )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.21 socket_iovec_alloc() : Construit un 'struct iovec'

```
resource socket_iovec_alloc ( int num_vectors, int )
```

Attention

Cette fonction est **EXPERIMENTALE**. Cela signifie que le comportement de cette fonction, son

nom et concrètement, TOUT ce 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.

7.100.22 socket_iovec_delete() : Efface un vecteur

```
bool socket_iovec_delete ( resource iovec , int iov_pos )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.23 socket_iovec_fetch() : Lit les données du vecteur spécifié par iovec_id[iovec_position]

```
string socket_iovec_fetch ( resource iovec , int iovec_position )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.24 socket_iovec_free() : Détruit le vecteur iovec spécifié par iovec_id

```
bool socket_iovec_free ( resource iovec )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.25 socket_iovec_set() : Modifie les données du vecteur iovec_id[iovec_position]

```
bool socket_iovec_set ( resource iovec , int iovec_position , string new_val )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.26 socket_last_error() : Lit la dernière erreur générée par une socket

```
int socket_last_error ( resource socket )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_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'echec, 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.

7.100.27 socket_listen() : Attend une connexion (listen) sur une socket

```
bool socket_listen ( resource socket , int backlog )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

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 entrantes.

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, c'est tronqué automatiquement à `SOMAXCONN`. Sous Windows32, 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`.

7.100.28 socket_read() : Lit des données d'une socket

string **socket_read** (resource socket , int length , 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.

`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` or `\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. 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_read` peut retourner une chaîne de longueur zéro, qui indique la fin de la communication (l'autre socket a fermé la communication).

Le paramètre optionnel `type` peut prendre l'une des valeurs constantes suivantes :

- `PHP_BINARY_READ` – utilise la fonction système `read()`. 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`.

7.100.29 socket_readv() : Lit des données dans un fd, en utilisant le tableau scatter-gather défini par iovec_id

bool **socket_readv** (resource socket , resource iovec_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.

7.100.30 socket_recv() : Reçoit des données d'une socket connectée

```
int socket_recv ( resource socket , string &buf , int len , 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.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.100.31 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 est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.32 socket_recvmsg() : Reçoit des messages sur un socket, qu'elle soit orientée connexion ou pas

```
bool socket_recvmsg ( resource socket , resource iovec , array &control , int &controllen , int &flags , string &addr , int &port )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.33 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 )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_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 sets de file descriptor. 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.

Attention

En sortie de fonction, les tableaux sont modifiés pour indiquer quelles sockets ont changé d'états.

Vous n'avez pas besoin de passer tous les tableaux à socket_select . 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 socket_select .

Exemple avec socket_select

```
<?php
/* Prépare le tableau read (socket surveillées en lecture) */
$read = array($socket1, $socket2);

if (false === ($num_changed_sockets = socket_select($read, $write = NULL, $except = NULL, 0))) {
    /* Gestion des erreurs */
} else if ($num_changed_sockets > 0) {
    /* Au moins une des sockets a été modifié */
}
?>
```

Note

A cause d'une limitation du Zend Engine 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 :

Passer NULL à socket_select

```
<?php
socket_select($r, $w, $e = NULL, 0);
?>
```

Les paramètres ^{tv_sec} and ^{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 contenus 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 <code>FALSE</code> via <code>==</code> donnerait <code>TRUE</code> :
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 :
• 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 dépends 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'elles ne pourront pas écrire ou lire toutes les données que vous demandez. Soyez prêts 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`.

7.100.34 socket_send() : Envoie des données à une socket connectée

```
int socket_send ( resource socket , string buf , int len , 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.

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`.

7.100.35 socket_sendmsg() : Envoie un message à une socket, qu'elle soit orientée connexion ou pas

```
bool socket_sendmsg ( resource socket , resource iovector , int flags , string addr , int port )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.36 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 )`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_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
$ssh = socket_create(AF_INET,SOCK_STREAM,SOL_TCP);
if (socket_bind($ssh, '127.0.0.1', 4242)) {
    echo "Socket connectée correctement";
}
$buf = 'Test Message';
$len = strlen($buf);
if (socket_sendto($ssh, $buf, $len, 0x100, '192.168.0.2', 4242) !== FALSE) {
    echo "Message envoyé correctement";
}
socket_close($ssh);
?>
```

Voir aussi `socket_send` et `socket_sendmsg` .

7.100.37 socket_set_block() : Met la socket en mode bloquant

`bool socket_set_block ( resource socket )`

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi `socket_set_nonblock` et `socket_set_option`

7.100.38 socket_set_nonblock() : Sélectionne le mode non bloquant d'un descripteur de fichier

```
bool socket_set_nonblock ( resource socket )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.100.39 socket_set_option() : Modifie les options de socket

```
bool socket_set_option ( resource socket , int level , int optname , mixed optval )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

Note

Cette fonction était appelée `socket_setopt()` avant PHP 4.3.0.

7.100.40 socket_shutdown() : Eteind une socket en lecture et/ou écriture

```
bool socket_shutdown ( resource socket , int how )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_shutdown` vous permet d'empêcher les données entrantes ou sortantes ou les deux (par défaut), d'être émise 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

7.100.41 socket_strerror() : Retourne une chaîne décrivant un message d'erreur

```
string socket_strerror ( 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.

`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 c'est passé.

Exemple avec `socket_strerror`

```
<?php
if (false == ($socket = @socket_create(AF_INET, SOCK_STREAM, 0))) {
    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` .

7.100.42 `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 nécessaire tous les octets du buffer fourni. Il est valide que, suivant certaines configuration de buffer réseau, que seulement une certaine quantité de données, même un octets, soit écrit, même 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](#).

7.100.43 `socket_writev()` : Ecrit des données dans un descripteur de fichier, fd, en utilisant le tableau scatter-gather définie par `iovec_id`

`bool socket_writev ( resource socket , resource iovec_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.

7.101 Flôts

7.101.1 Introduction

Les flôts ("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 flôt 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 flôt se comporte spécifiquement. C'est le cas du gestionnaire ^{http}, qui sait comment traduire une URL en une requête sur un serveur distant. Il existe de nombreux gestionnaires intégré à ^{PHP} par défaut (voir [Liste des protocoles supportés](#)), et de plus, des gestionnaires spécifiques peuvent être ajouté 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 flôts 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](#).

7.101.2 Filtres de flôts

Un ^{filtre} est une fonction finale qui effectue des opérations sur les données qui sont lues ou écrites dans un flôt. Un nombre arbitraire de filtre peuvent être ajoutés sur un flôt. 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 flôts les auteurs d'extensions](#). Pour connaître la liste des gestionnaires actuellement enregistrés, utilisez la fonction [stream_get_filters](#).

Un flôt est identifié dans la forme ^{protocole} :// ^{serveur}

- ^{protocole} ([chaîne de caractères](#)) – Le nom du protocole utilisé. Par exemple : file, http, https, ftp, ftps, compress.zlib, compress.bz2, ssl, tls et php. Voyez [Liste des protocoles supportés](#) pour une liste de gestionnaires intégrés à ^{PHP}. Si aucun gestionnaire n'est spécifié, la fonction utilise par défaut ^{file} ://).
- ^{serveur} – Dépend du gestionnaire utilisé. Pour les flôts liés au système de fichier, c'est généralement un chemin et un nom de fichier. Pour les objets réseau, c'est généralement un nom de serveur, et un chemin. Encore une fois, reportez vous à [Liste des protocoles supportés](#) pour une description des différents serveurs de gestionnaire.

7.101.3 Contextes de flôts

Un ^{contexte} est un jeu de paramètres et d'options spécifiques à un gestionnaire qui modifie ou améliore le comportement d'un flôt. Les contextes sont créés en utilisant la fonction [stream_context_create](#) et peuvent être donné aux fonctions de créations de flôts sur le système de fichier (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 flôt. La fonction d'alerte est appelé durant la reception 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 flôt, 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 lorsque possible.

7.101.4 Installation

Les flôts font parti de ^{PHP} depuis la version 4.3.0. Aucune étape supplémentaire n'est requise pour les activer.

7.101.5 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 et 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 implémentation de filtres utilisateurs.

7.101.6 Constantes prédefinies

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
<code>STREAM_FILTER_READ</code>	Utilisée avec stream_filter_append et stream_filter_prepend pour indiquer que le filtre spécifié ne doit être appliqué qu'après la lecture .
<code>STREAM_FILTER_WRITE</code>	Utilisée avec stream_filter_append et stream_filter_prepend pour indiquer que le filtre spécifié ne doit être appliqué qu'après l' écriture .
<code>STREAM_FILTER_ALL</code>	Cette constante est équivalente à <code>STREAM_FILTER_READ STREAM_FILTER_WRITE</code>
<code>STREAM_USE_PATH</code>	Option indiquant si stream a utilisé l'include path.
<code>STREAM_REPORT_ERRORS</code>	Option indiquant si le gestionnaire est responsable pour la levée des erreurs avec trigger_error durant l'ouverture du flôt. Si cette constante n'existe pas, vous ne devez pas émettre d'erreurs.

7.101.7 Erreurs de flôts

Comme avec n'importe quel fichier ou socket, les opérations sur un flôt peuvent échouer pour une grande variété de raisons (par exemple : impossible de se connecter au serveur distant, fichier introuvable, etc...). Un flôt peut aussi échouer parceque 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.

7.101.8 Exemples

Exemples avec `file_get_contents`

```
<?php
// Lit un fichier local dans le dossier /home/bar
$localfile = file_get_contents("/home/bar/foo.txt");

// Identique au precedent, mais utilise explicitement le gestionnaire FILE
$localfile = file_get_contents("file:///home/bar/foo.txt");

// Lit un fichier distant sur le serveur www.exemple.com avec le protocole HTTP
$httpfile = file_get_contents("http://www.exemple.com/foo.txt");

// Lit le meme fichier sur le serveur www.exemple.com avec le protocole HTTPS
$httpsfile = file_get_contents("https://www.exemple.com/foo.txt");

// Lit un fichier distant sur le serveur ftp.exemple.com en utilisant le protocole FTP
$ftpfile = file_get_contents("ftp://user:pass@ftp.exemple.com/foo.txt");

// Lit un fichier distant sur le serveur ftp.exemple.com en utilisant le protocole FTPS
$ftpsfile = file_get_contents("ftps://user:pass@ftp.exemple.com/foo.txt");
?>
```

Envoi d'une requête de type POST sur une 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";

print "\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 flûts](#)
- [Contextes de flûts](#)
- [Installation](#)
- [Classes Stream](#)
- [Constantes prédefinies](#)

- [Erreurs de flôts](#)
- [Exemples](#)
- [stream_context_create](#)
- [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_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_get_name](#)
- [stream_socket_server](#)
- [stream_wrapper_register](#)

7.101.10 stream_context_get_options() : Lit la valeur des options pour un flôt/wrapper/contexte

array **stream_context_get_options** (resource stream|context)

[stream_context_get_options](#) retourne un tableau contenant la liste des options pour le flôt ou le contexte stream|context .

7.101.11 stream_context_set_option() : Configure une option pour un flôt/wrapper/contexte

bool **stream_context_set_option** (resource context|stream , string wrapper , string option , mixed value)

[stream_context_set_option](#) configure l'option *option* avec la nouvelle valeur *value* le flôt ou le wrapper ou le contexte *wrapper* .

7.101.12 stream_context_set_params() : Configure les paramètres pour un flôt/wrapper/contexte

bool **stream_context_set_params** (resource stream|context , array params)

stream_context_set_params configure les paramètres pour un flôt, un wrapper 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 flôt génère une notification.

7.101.13 stream_copy_to_stream() : Copie des données depuis un flôt vers un autre

```
int stream_copy_to_stream ( resource source , resource dest , int maxlength )
```

stream_copy_to_stream copie jusqu'à `maxlength` octets de données depuis la position courante du pointeur dans le flôt `source` vers le paramètre `dest`. Si `maxlength` n'est pas spécifié, tout le reste du flôt `source` sera copié. stream_copy_to_stream retourne le total d'octets copié.

Exemple avec stream_copy_to_stream

```
<?php
$src = fopen('http://www.exemple.com','r');
$dest1 = fopen('first1k.txt','w');
$dest2 = fopen('remainder.txt','w');

print stream_copy_to_stream($src, $dest1, 1024) . " octets copiés vers first1k.txt\n";
print stream_copy_to_stream($src, $dest2) . " octets copiés vers remainder.txt\n";

?>
```

Voir aussi copy.

7.101.14 stream_filter_append() : Attache un filtre à un flôt en fin de liste

```
bool stream_filter_append ( resource stream , string filtername , int read_write , string params )
```

stream_filter_append ajoute le filtre `filtername` à la liste de filtres attachés au flôt `stream`. Ce filtre sera ajoutés avec les paramètres spécifiés dans `params` à la fin de cette liste et sera donc appelé en dernier durant les opérations de flôts. Pour ajouter un filtre au début de la liste, utilisez la fonction stream_filter_prepend.

Par défaut, stream_filter_prepend va ajouter le filtre à la liste de filtre 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.

Contrôler l'application des filtres

```
<?php
/* Open a test file for reading and writting */
$fzp = fopen("test.txt","rw");

/* Apply the ROT13 filter to the
 * write filter chain, but not the
 * read filter chain */
stream_filter_append($fp, "string.rot13", STREAM_FILTER_WRITE);

/* Write a simple string to the file
 * it will be ROT13 transformed on the
 * way out */
fwrite($fp, "This is a test\n");

/* Back up to the beginning of the file */
rewind($fp);

/* Read the contents of the file back out.
 * Had the filter been applied to the
 * read filter chain as well, we would see
 * the text ROT13ed back to its original state */
fpassthru($fp);

fclose($fp);
```

```
/* Expected Output
-----

Guvf vf n grfg

*/
?>
```

Note

Quand utiliser des filtres personnalisés

[stream_register_filter](#) doit être appelé avant [stream_filter_prepend](#) pour enregistrer le filtre sous le nom de `filtername`.

Voir aussi [stream_register_filter](#) et [stream_filter_prepend](#)

7.101.15 stream_filter_prepend() : Attache un filtre à un flôt en début de liste

```
bool stream_filter_prepend ( resource stream , string filtername , int read_write , string params )
```

[stream_filter_prepend](#) ajoute le filtre `filtername` à la liste de filtres attachés au flôt `stream`. Ce filtre sera ajoutés avec les paramètres spécifiés dans `params` à la fin de cette liste et sera donc appelé en dernier durant les opérations de flôts. 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 filtre 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.

Note

Quand utiliser des filtres personnalisés

[stream_register_filter](#) doit être appelé avant [stream_filter_prepend](#) pour enregistrer le filtre sous le nom de `filtername`.

Voir aussi [stream_register_filter](#) et [stream_filter_append](#)

7.101.16 stream_filter_register() : Enregistre un filtre de flôt

```
bool stream_filter_register ( string filtername , string classname )
```

[stream_filter_register](#) vous permet d'implémenter votre propre filtre de flôt, 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 flôt 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 comportement indéfini.

[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 appelé à chaque fois que des données sont lues ou écrites dans le flût 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 donnée 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 flût 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 : `PSFS_PASS_ON` indique la réussite, et les données sont dans le paramètre `out`, une bucket brigade. `PSFS_FEED_ME` indique que le filtre n'a aucune données à retourner, et requiert d'autres données du flût. `PSFS_ERR_FATAL` indique que le flût a rencontré une erreur fatale, et ne peut continuer. Si aucune valeur n'est retournée par cette méthode, `PSFS_ERR_FATAL` est utilisé.

```
void oncreate ( void )
```

Cette méthode est appelée durant l'instantiation du filtre. Si votre filtre alloue ou initialise d'autres ressources (comme des buffers), c'est le moment de le faire.

```
void onclose ( void )
```

Cette méthode est appelée durant l'extinction du filtre (généralement, lorsque le flût est fermé), et est exécuté **après** l'appel de la fonction `flush`. Si aucune ressource 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 flût `foo-bar.txt`, qui réalise un chiffrement de type ROT-13 sur toutes les lettres lue ou écrite dans le flût.

```
Filtre ROT13 pour les flûts

<?php

/* Definition de notre classe de filtre */
class rot13_filter extends php_user_filter {
    function read($length) {
        $tempstr = parent::read($length);
        for($i = 0; $i < strlen($tempstr); $i++)
            if (($tempstr[$i] >= 'A' AND $tempstr[$i] <= 'M') OR
                ($tempstr[$i] >= 'a' AND $tempstr[$i] <= 'm')) $tempstr[$i] = chr(ord($tempstr[$i]) + 13);
            else if (($tempstr[$i] >= 'N' AND $tempstr[$i] <= 'Z') OR
                ($tempstr[$i] >= 'n' AND $tempstr[$i] <= 'z')) $tempstr[$i] = chr(ord($tempstr[$i]) - 13);
        return $tempstr;
    }

    function write($data) {
        for($i = 0; $i < strlen($data); $i++)
            if (($data[$i] >= 'A' AND $data[$i] <= 'M') OR
                ($data[$i] >= 'a' AND $data[$i] <= 'm')) $data[$i] = chr(ord($data[$i]) + 13);
            else if (($data[$i] >= 'N' AND $data[$i] <= 'Z') OR
                ($data[$i] >= 'n' AND $data[$i] <= 'z')) $data[$i] = chr(ord($data[$i]) - 13);
        return parent::write($data);
    }
}

/* Enregistrement du filtre avec PHP */
stream_filter_register("rot13", "rot13_filter")
or die("Impossible d'enregistrer le filtre");

$fp = fopen("foo-bar.txt", "w");

/* Liaison du filtre avec le flût */
stream_filter_append($fp, "rot13");

fwrite($fp, "Line\n");
fwrite($fp, "Word - 2\n");
fwrite($fp, "Easy As 123\n");

fclose($fp);

/* Le filtre ne s'applique qu'au flût $fp
 * ce qui fait que cette lecture se fait sans déchiffrement
```

```

*/
readfile("foo-bar.txt");

/* affichage
 * -----

Yvar1
Jbeq - 2
Rnfl Nf 123

*/
?>

```

Voir aussi [stream_register_wrapper](#) , [stream_filter_prepend](#) et [stream_filter_append](#)

7.101.17 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 flôts disponibles sur le système.

Exemple avec la fonction [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](#)

7.101.18 stream_get_line() : Lit une ligne dans un flôt

string **stream_get_line** (resource handle , int length , string ending)

[stream_get_line](#) lit une ligne de taille maximale `length` dans le flôt `handle` . La lecture se termine lorsque `length` ont été lus, ou que la chaîne `ending` est trouvée dans le flôt (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](#) ,

7.101.19 stream_get_meta_data() : Lit les en-têtes et données méta des flôts

array **stream_get_meta_data** (resource stream)

stream_get_meta_data retourne les informations disponibles sur le flôt `stream`. Ce flôt peut être n'importe quel flôt créé par `fopen`, `fsockopen` et `pfsockopen`. Le tableau résultant peut contenir les éléments suivants :

- `timed_out` (booléen) – `TRUE` si le flôt a atteint de délai d'expiration en attendant des données durant le dernier appel aux fonctions `fread` et `fgets`.
- `blocked` (booléen) – `TRUE` si le flôt est en mode bloquant. Voir aussi `socket_set_blocking`.
- `eof` (booléen) – `TRUE` si le flôt 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 si il reste des données à lire, utilisez plutôt la fonction `feof`.
- `unread_bytes` (entier) – le nombre d'octets actuellement placés dans le buffer de lecture.

Les éléments suivants ont été ajoutés en PHP 4.3 :

- `stream_type` (chaîne de caractères) – un nom, qui décrit l'implémentation sous-jacente de flôt.
- `wrapper_type` (chaîne de caractères) – un nom qui décrit le gestionnaire de protocole pour ce flôt. Voyez Liste des protocoles supportés pour plus d'informations sur les gestionnaires.
- `wrapper_data` (mixed) – des données spécifiques au gestionnaire liés à ce flôt. 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 flôt. Les filtres sont actuellement non documentés.

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 cette fonction.

Note

Cette fonction n'est pas utilisable avec les sockets créées par les fonctions de l'extension `socket`.

7.101.20 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
$xporthlist = stream_get_transports();
print_r($xporthlist);

/* L'affichage ne sera jamais 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](#)

7.101.21 stream_get_wrappers() : Liste les gestionnaires de flôts

array **stream_get_wrappers** (void)

stream_get_wrappers retourne un tableau indexé contenant le nom de tous les gestionnaires de flôts disponibles sur le système.

Voir aussi [stream_wrapper_register](#) .

7.101.22 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.

7.101.23 stream_select() : Runs the equivalent of the select() system call on the given arrays of streams with a timeout specified by tv_sec and tv_usec

int **stream_select** (resource &read , resource &write , resource &except , int tv_sec , int tv_usec)

[stream_select](#) accepte un tableau de flôts et attend que l'un d'entre eux change de statut. Cette opération est équivalente à ce que fait la fonction [socket_select](#) , ormis le fait qu'elle travaille sur flôts.

Les flôts 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 flôts qui sont en fin de fichier, auquel cas un appel à la fonction [fread](#) retournera une chaîne de taille 0).

Les flôts qui sont listés dans le paramètre ^{write} seront surveillé en écriture (pour être précis, si une écriture ne bloquera pas).

Les flôts qui sont listés dans le paramètre ^{except} seront surveillé pour voir si une exception est levée.

Attention

Lors de la fin de la fonction, les tableaux sont modifiés pour rapporter leurs évolutions respectives.

Vous n'êtes pas obligés 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.

Exemple

```
<?php
/* Preparation du tableau de flôts lecture */
$read = array($stream1, $stream2);

if (false === ($num_changed_streams = stream_select($read, $write = NULL, $except = NULL, 0))) {
    /* Gestion d'erreur */
} else if ($num_changed_streams > 0) {
```

```

    /* Au moins un des flôts a évolué */
}
?>

```

Note

A cause d'une limitation du Zend Engine 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);
?>

```

Les paramètres `tv_sec` et `tv_usec` ensemble forment le délai d'expiration. Ce délai est la durée maximale d'exécution de `stream_select`. `tv_sec` peut valoir zéro, ce qui fait que `stream_select` s'achève immédiatement. C'est pratique pour réaliser du polling. Si `tv_sec` est `NULL` (pas de délai d'expiration), `stream_select` peut bloquer le script indéfiniment.

En cas de succès, `stream_select` retourne le nombre de flôts qui ont évolué, ce qui peut être zéro si le délai d'expiration a été atteint avant que les flôts n'évoluent. En cas d'erreur, la fonction retourne `FALSE`.

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 `==` la confondrait avec une erreur.

```

<?php
if (false === stream_select($r, $w, $e = NULL, 0)) {
    echo "stream_select() failed\n";
}
?>

```

Note

Gardez à l'esprit que certaines implémentations de flôts doivent être gérées très prudemment. Voici quelques règles à suivre :

- Il faut toujours essayer d'utiliser `stream_select` sans délai d'expiration. Votre programme ne devrait rien faire si il ne reçoit pas de données extérieures. Les codes qui dépendent du délai d'expiration ne sont généralement pas portable, et difficiles à déboguer.
- Si vous avez écrit ou lu dans un flôt qui retournés dans les tableaux de flôts, soyez bien conscient que ces flôts n'ont peut être pas écrit ou lu la totalité des données demandées. Soyez préparés à être en mesure de lire un seul octet.

Voir aussi [stream_set_blocking](#).

7.101.24 stream_set_blocking() : Configure le mode bloquant d'un flôt

```
bool stream_set_blocking ( resource stream, int mode )
```

`stream_set_blocking` configure le mode bloquant du flôt `stream`. Si `mode` vaut `FALSE`, `stream` sera configuré en mode non-bloquant, et si il vaut `TRUE`, `stream` sera configuré en mode bloquant. Cet appel affecte les fonctions telles que `fgets` et `fread` qui lisent dans des flôts. En mode non-bloquant, la fonction `fgets` s'exécute juste après son appel, alors qu'en mode bloquant, elle attendra des données.

`stream_set_blocking` était appelée `set_socket_blocking` et plus tard `socket_set_blocking` mais ces appellations sont obsolètes.

Note

Depuis PHP 4.3, cette fonction peut (potentiellement) fonctionner avec n'importe quel flôt. Avant PHP 4.3, les flôts utilisant des sockets sont les seuls qui soient supportés dans le cœur de PHP, même si les autres extensions pourraient supporter cette fonction.

Voir aussi [stream_select](#) .

7.101.25 stream_set_timeout() : Configure la durée d'expiration d'un flôt

`bool stream_set_timeout ( resource stream , int seconds , int microseconds )`

[stream_set_timeout](#) configure la durée d'expiration du flôt `stream`, exprimé comme la durée de `seconds` secondes et `microseconds` micro-secondes.

Exemple avec [stream_set_timeout](#)

```
<?php
$fp = fsockopen("www.example.com", 80);
if(!$fp) {
    echo "Unable to open\n";
} else {
    fputs($fp, "GET / HTTP/1.0\n\n");
    $start = time();
    stream_set_timeout($fp, 2);
    $res = fread($fp, 2000);
    var_dump(stream_get_meta_data($fp));
    fclose($fp);
    print $res;
}
?>
```

Note

Depuis PHP 4.3, cette fonction peut (potentiellement) fonctionner avec n'importe quel flôt. Avant PHP 4.3, les flôts utilisant des sockets sont les seuls qui soient supportés dans le cœur de PHP, même si les autres extensions pourraient supporter cette fonction.

[stream_set_timeout](#) était appelée auparavant [set_socket_timeout](#), et aussi [socket_set_timeout](#) mais ces appellations sont obsolètes.

Voir aussi [fsockopen](#) et [fopen](#) .

7.101.26 stream_set_write_buffer() : Configure la bufferisation de fichier pour un flôt

`int stream_set_write_buffer ( resource stream , int buffer )`

[fwrite](#) est habituellement configuré avec un buffer de 8ko. Cela signifie que si deux processus veulent écrire dans le même flôt de sortie (par exemple, un fichier), ils font une pause tous les 8ko pour laisser l'autre écrire aussi. [stream_set_write_buffer](#) configure la taille du buffer des opérations d'écritures pour le flôt `stream` avec la nouvelle valeur de `buffer` octets. Si `buffer` vaut 0, les opérations d'écritures 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 flôt 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](#) .

7.101.27 stream_socket_accept() : Accepte une connexion sur une socket créée par [stream_socket_server](#)

resource **stream_socket_accept** (resource server_socket , int 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` .

Voir aussi [stream_socket_server](#) , [stream_socket_get_name](#) , [stream_set_blocking](#) , [stream_set_timeout](#) , [fgets](#) , [fgetss](#) , [fputs](#) , [fclose](#) , [feof](#) et l' [extension CURL](#) .

7.101.28 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 flôt ou une connexion datagramme avec la destination `remote_socket` . Le type de socket créé 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 quelle option de connexion. Actuellement, les valeurs possibles pour ces options sont `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 flôt, 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 retournée `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 qui sont disponibles sur le système est accessible via `stream_get_transports`.

Le flût 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.exemple.com:80", $errno, $errstr, 30);
if (!$fp) {
    echo "Errstr ($errno)<br>\n";
} else {
    fputs ($fp, "GET / HTTP/1.0\r\nHost: www.exemple.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](#).

7.101.29 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. Si il vaut `FALSE`, c'est le nom de la socket locale.

Voir aussi [stream_socket_accept](#).

7.101.30 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 flût ou un datagramme sur la socket spécifiée [local_socket](#). Le type de socket créé 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.

[flags](#) est un champ de bits qui peut être la combinaison de n'importe quelle option de connexion. Actuellement, les valeurs possibles pour ces options sont [STREAM_CLIENT_ASYNC_CONNECT](#) et [STREAM_CLIENT_PERSISTENT](#). La valeur par défaut est [STREAM_SERVER_BIND](#) | [STREAM_SERVER_LISTEN](#).

[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 de l'appel à [connect\(\)](#). Si la valeur retournée dans [errno](#) est 0 et que la fonction a retournée [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 passé par référence.

Suivant votre environnement, les sockets Unix ou le délai d'expiration peuvent ne pas être disponibles. Une liste des transports disponibles qui sont disponibles sur le système est accessible via [stream_get_transports](#).

Exemple avec [stream_socket_server](#)

```
<?php
$socket = stream_socket_server("tcp://0.0.0.0:8000", $errno, $errstr);
if (!$socket) {
    echo "Erreur: $errstr<br>\n";
} else {
    while ($conn = stream_socket_accept($socket)) {
        fputs($conn, "Merci de vous connecter à mon serveur. 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, $errstr);
if (!$socket) {
    echo "ERROR: $errstr<br>\n";
} else {
    while ($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](#), [fputs](#), [fclose](#), [feof](#) et l'[extension CURL](#).

7.101.31 stream_wrapper_register() : Enregistre une enveloppe URL, implémentée comme classe PHP

```
bool stream_wrapper_register ( string protocol , string classname )
```

[stream_wrapper_register](#) vous permet d'implémenter vos propres gestionnaires de protocoles et de flôts, à utiliser avec toutes les autres fonctions de fichiers, comme [fopen](#), [fread](#) etc...

Pour implémenter une enveloppe, vous devez définir une classe avec la liste des membres définie ci-dessous. Lorsque quelqu'un ouvre votre flôt, PHP va créer une instance de la classe `classname` et appeler les méthodes de cette instance. Vous devez implémenter ces méthodes exactement comme décrit ci-dessous : sinon, vous risquez de rencontrer des comportements indéfinis.

[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 flôt. `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.

`options` contient des options supplémentaires, utilisées par les API de flôts. Il peut contenir une ou plusieurs des options suivantes, combinées par des OR.

Option	Description
STREAM_USE_PATH	Si <code>path</code> est relatif, recherche la ressources en utilisant la configuration de <code>l'include_path</code> .
STREAM_REPORT_ERRORS	Si cette option est activée, vous êtes responsable pour lever les erreurs avec trigger_error durant l'ouverture du flôt. Si cette option n'est pas activée, vous ne devez lever aucune erreur.

Si le paramètre `path` est ouvert avec succès, et que `STREAM_USE_PATH` est activé dans le paramètre `options`, vous devez affecter à `opened_path` le chemin complet de la ressource ou du fichier que vous avez réellement ouvert.

Si la ressource demandée a été ouverte, vous devez retourner `TRUE`, ou sinon, vous devez retourner `FALSE`.

```
void stream_close ( void )
```

Cette méthode est appelée lors que flôt est fermée, grâce à la fonction [fclose](#). Vous devez libérez toutes les ressources réservées par le flôt.

```
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. Si il y a moins que ^{count} octets disponibles, vous devez en retourner autant que possible. Si aucune autre donnée n'est disponible, retournez soit ^{FALSE} soit une chaîne vide. Vous devez aussi tenir à jour la position du pointeur d'écriture/lecture dans le flût, 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 flût. Si 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 flût, en ajoutant ou retranchant le nombre d'octets lus.

```
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 si il n'y a plus de donnée disponible 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.

```
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 `fflush` est utilisée. Si vous avez mis des données dans un système de cache pour votre flût, mais qu'ils ne sont pas encore stockés 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 donnée 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 flût, et retourne un tableau, avec les valeurs appropriées pour le flût.

```
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

Userspace wrapper unlink method is not supported prior to ^{PHP 5.0.0} .

```
bool dir_opendir ( string path , int options )
```

Cette méthode est appelée immédiatement lorsque votre flôt 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.

```
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 flôt 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 flôt pour accéder aux variables globales

```
<?php

class VariableStream {
    var $position;
    var $varname;

    function stream_open($path, $mode, $options, &$opened_path)
    {
        $url = parse_url($path);
        $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);
?>

```

7.102 Chaînes de caractères (Strings)

7.102.1 Introduction

Ces fonctions permettent de manipuler des chaînes de caractères de différentes façons. Certaines fonctionnalités plus spécialisées sont disponibles dans les sections d'expression régulières 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 .

7.102.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.102.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.102.4 Constantes prédefinies

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 integer
CRYPT_STD_DES integer
CRYPT_EXT_DES integer
CRYPT_MD5 integer
CRYPT_BLOWFISH integer
HTML_SPECIALCHARS ( integer )
HTML_ENTITIES ( integer )
ENT_COMPAT ( integer )
ENT_QUOTES ( integer )
ENT_NOQUOTES ( integer )
CHAR_MAX ( integer )
LC_CTYPE ( integer )
LC_NUMERIC ( integer )
LC_TIME ( integer )
LC_COLLATE ( integer )
LC_MONETARY ( integer )
LC_ALL ( integer )
LC_MESSAGES ( integer )
STR_PAD_LEFT ( integer )
STR_PAD_RIGHT ( integer )
STR_PAD_BOTH ( integer )
```

7.102.5 Voir aussi

Pour des fonctions encore plus puissantes de gestion et manipulation des chaînes, reportez-vous aux expressions régulières POSIX et expressions régulières Perl .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes prédefinies](#)
- [Voir aussi](#)
- [addslashes](#)
- [addslashes](#)
- [bin2hex](#)
- [chop](#)
- [chr](#)
- [chunk_split](#)
- [convert_cyr_string](#)
- [count_chars](#)
- [crc32](#)
- [crypt](#)
- [echo](#)
- [explode](#)
- [fprintf](#)
- [get_html_translation_table](#)
- [hebrew](#)
- [hebrevc](#)
- [html_entity_decode](#)
- [htmlentities](#)
- [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](#)
- [strpos](#)
- [strrchr](#)
- [strev](#)
- [stripos](#)
- [strrpos](#)
- [strspn](#)
- [strstr](#)
- [strtok](#)
- [strtolower](#)
- [strtoupper](#)
- [strtr](#)
- [substr_count](#)
- [substr_replace](#)
- [substr](#)
- [trim](#)
- [ucfirst](#)
- [ucwords](#)
- [vprintf](#)
- [vsprintf](#)
- [wordwrap](#)

7.102.7 addslashes() : Ajoute des anti-slashes dans une chaîne

`string addslashes ( string str )`

[addslashes](#) retourne la chaîne `str`, après avoir échappé tous les caractères qui doivent l'être, pour être utilisé dans une requête de base de données. Ces caractères sont les guillemets simples (`'`), guillemets doubles (`"`), anti-slash (```) et NUL (le caractère `NULL`).

Note

<code><u>magic_quotes_gpc</u></code> est activé par défaut.

Voir aussi `stripslashes` , `htmlspecialchars` , `quotemeta` et `get_magic_quotes_gpc` .

7.102.8 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.

Voir aussi `pack` et `unpack` .

7.102.9 chop() : Alias de `rtrim`

`chop` est un alias de `rtrim` .

Note

<code><u>chop</u></code> est différente de son équivalent Perl <code>chop()</code> , qui supprime le dernier caractère dans la chaîne.

7.102.10 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 <code><u>chr</u></code>

<pre><?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("The string ends in escape: %c", 27); ?></pre>

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` .

7.102.11 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 des 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
// Formatter des données pour suivre la norme RFC 2045

$new_string = chunk_split(base64_encode($data));
?>
```

Voir aussi [explode](#) , [split](#) , [wordwrap](#) et [RFC 2045](#) .

7.102.12 `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

7.102.13 `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.

7.102.14 `crc32()` : Calcule la somme de contrôle CRC32

`int crc32 ( string str )`

`crc32` génère la somme de contrôle cyclique CRC 32, calculée sur 32 bits, et appliqué à la chaîne `str` . Cette fonction est généralement utilisée pour valider l'intégrité de données durant une transmission.

Note

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` .

7.102.15 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`.

Si l'argument `salt` n'est pas fourni, `PHP` en générera un lui-même.

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 récursivement, 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 1 ou 0, 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\$

Note

Il n'existe pas de fonction de déchiffrement, car la fonction `crypt` utilise un algorithme à un seul sens (injection).

Exemple avec `crypt`

```
<?php
$mot_de_passe = crypt("MonM0TdEPa&&e");
// laissons le salt initialisé par
<TT>PHP</TT>
```

```

/*
  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!";
}
?>

```

Voir aussi [md5](#) et [l'extension Mcrypt](#) .

7.102.16 echo() : Affiche une chaîne de caractères

```
void echo ( string arg1 , string argn... )
```

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 obligés d'utiliser des parenthèses. En fait, si vous voulez passer plusieurs paramètres à afficher, vous ne devez pas utiliser de parenthèses. Il n'est pas possible d'utiliser [echo](#) dans un contexte de [fonction variable](#) . Dans ce cas, il faut utiliser [print](#) à la place.

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("clef" => "foo");

echo "c'est {$bar['clef']} !" ; // 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

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;

// comme echo() n'est pas 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 une fonction
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 (=).

Afficher sans l'aide de `echo`

J'ai <?=\$foo?> fous.

Note

Cette syntaxe courte n'est possible que si la configuration de `short_open_tag` a été activée.

Pour une brève discussion à propos des différences entre `print` et `echo`, consultez cet article des FAQs 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`.

7.102.17 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.

Note

Le paramètre `limit` a été ajouté en PHP 4.0.1.

Exemple avec `explode`

```
<?php
$pizza = "garniture1 garniture2 garniture3 garniture4";
$pieces = explode(" ", $pizza);

$data = "foo*:1023:1000:/home/foo:/bin/sh";
list($user,$pass,$uid,$gid,$gecos,$home,$shell) = explode(":",$data);
?>
```

Note

bien que pour des raisons historiques, **implode** accepte les paramètres dans un ordre ou l'autre, **explode** ne le peut pas. Vous devez vous assurer que l'argument séparateur `separator` arrive avant l'argument `string`.

Voir aussi `preg_split`, `spliti`, `split` et `implode`.

7.102.18 fprintf() : Ecrit une chaîne formatée dans un flût

int **fprintf** (resource handle , string format , mixed args)

fprintf écrit la chaîne produite avec le format `format` dans le flût représenté par `handle`.

Le paramètre `format` est composé de zéro, une ou plusieurs directives : des caractères ordinaires (excluant %) qui sont copiés directement dans le résultat sans modification, et des **spécifications**

de conversion , qui sont exploitées avec un des paramètres suivants. Cette technique s'applique à `fprintf` , `sprintf` et `printf` .

Chaque spécification de conversion est constituée d'un signe pourcentage, (%), suivi par un ou plusieurs éléments, dans cet ordre :

1. 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.
2. 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 est aligné à droite.
3. Un nombre optionnel, **spécificateur de taille** indique le nombre minimum de caractères que cette conversion doit fournir en résultat.
4. Un **spécificateur de précision** qui indique le nombre de décimales qui doivent être affichées pour les nombres à virgule flottante. Cette option n'a pas d'effet pour les autres types que ^{float} . Une autre fonction pratique pour formater les nombres à virgule flottante est `number_format` .
5. 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.
 - ♦ ^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.
 - ♦ ^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).

Voir aussi `printf` , `sprintf` , `sscanf` , `fscanf` , `vsprintf` et `number_format` .

7.102.19 `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. 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` .

Exemple avec la table de traduction des caractères en entités HTML

```
<?php
$trans = get_html_translation_table(HTML_ENTITIES);
$str = "Hallo & <Frau> & Krämer";
$encoded = strtr($str, $trans);
?>
```

La variable `$encoded` contient maintenant "Hallo & <Frau> & Krämer".

Un truc pratique est d'utiliser [array_flip](#) pour changer la direction de la traduction.

Exemple avec inversion de la table de traduction des caractères en entités HTML

```
<?php
$trans = array_flip($trans);
$original = strtr($encoded, $trans);
?>
```

Le contenu de `$original` sera alors : "Hallo & <Frau> & Krämer" .

Voir aussi [htmlspecialchars](#) , [htmlentities](#) , [strtr](#) et [array_flip](#) .

7.102.20 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 lignes dans le résultat. [hebrew](#) essaie d'éviter de scinder les mots.

Voir aussi [hebrevc](#) .

7.102.21 hebrevc() : Convertit un texte logique hébreu 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 (`\n`) en "`<br>\n`". Le paramètre optionnel `max_chars_per_line` indique le nombre maximum de caractères par lignes dans le résultat. [hebrevc](#) essaie d'éviter de scinder les mots.

Voir aussi [hebrew](#) .

7.102.22 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
ENT_COMPAT	Convertit les guillemets doubles et ignore les guillemets simples.
ENT_QUOTES	Convertit les guillemets doubles et les guillemets simples.
ENT_NOQUOTES	Convertit les guillemets simples et ignore les guillemets doubles.

Le jeu de caractères ISO-8859-1 est utilisé par défaut, comme paramètre `charset`. Ce paramètre permet de choisir le jeu de caractères utilisés dans la conversion.

Les jeux de caractères suivantes sont disponibles et supportés par `PHP 4.3.0` et plus récent.

Jeux de caractères	Description
ISO-8859-1	Europe occidentale, Latin-1
ISO-8859-15	Europe occidentale, Latin-9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manque au Latin-1(ISO-8859-1).
UTF-8	Unicode 8 bits multi octets, compatible avec l'ASCII
cp1252	Jeux de caractères spécifique de Windows pour l'Europe occidentale.
BIG5	Chinois traditionnel, principalement utilisé à Taiwan.
GB2312	Chinois simplifié, officiel.
BIG5-HKSCS	Big5 avec les extensions de Hong Kong, chinois traditionnel.
Shift_JIS	Japanais
EUCJP	Japanais

Note
Les autres jeux de caractères ne sont pas reconnus, et le ISO-8859-1 sera utilisé à la place.

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) { \$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
Vous pourriez vous demander pourquoi <code>trim(html_entity_decode('&nbsp;'))</code> ; ne résuit pas la chaîne à la chaîne vide. C'est parceque l'entité ' ' 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.

Voir aussi [htmlentities](#), [htmlspecialchars](#), [get_html_translation_table](#), [htmlspecialchars](#) et [urldecode](#).

7.102.23 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 un deuxième argument optionnel, qui indique comment doivent être traités les guillemets doubles et simples. Vous pouvez utiliser l'une des constantes suivantes :

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 les guillemets doubles et les guillemets simples

Le support du paramètre optionnel `quote` 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 suivantes sont disponibles et supportés par ^{PHP} 4.3.0 et plus récent.

Jeux de caractères	Description
ISO-8859-1	Europe occidentale, Latin-1
ISO-8859-15	Europe occidentale, Latin-9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manque au Latin-1(ISO-8859-1).
UTF-8	Unicode 8 bits multi octets, compatible avec l'ASCII
cp1252	Jeux de caractères spécifique de Windows pour l'Europe occidentale.
BIG5	Chinois traditionnel, principalement utilisé à Taiwan.
GB2312	Chinois simplifié, officiel.
BIG5-HKSCS	Big5 avec les extensions de Hong Kong, chinois traditionnel.
Shift_JIS	Japanais
EUCJP	Japanais

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 htmlentities
<pre><?php \$str = "Un 'apostrophe' en gras"; // Affiche : Un 'apostrophe' en gras echo htmlentities(\$str); // Affiche : Un &#039;apostrophe&#039; en gras echo htmlentities(\$str, ENT_QUOTES); ?></pre>

Voir aussi [html_entity_decode](#), [get_html_translation_table](#), [htmlspecialchars](#), [nl2br](#) et [urlencode](#).

7.102.24 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. Cette fonction prend un deuxième argument optionnel, 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 :

- '&' (ampersand) devient '&#039;'
- '"' (guillemets doubles) devient '"'
- "'" (single quote) devient '''
- '<' (supérieur à) devient '<'
- '>' (inférieur 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 jeux de caractères utilisé durant la conversion. Le support de cet argument a été ajouté en ^{PHP} 4.1.0.

Les jeux de caractères suivantes sont disponibles et supportés par ^{PHP} 4.3.0 et plus récent.

Jeux de caractères	Description
ISO-8859-1	Europe occidentale, Latin-1
ISO-8859-15	Europe occidentale, Latin-9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manque au Latin-1(ISO-8859-1).
UTF-8	Unicode 8 bits multi octets, compatible avec l'ASCII
cp1252	Jeux de caractères spécifique de Windows pour l'Europe occidentale.
BIG5	Chinois traditionnel, principalement utilisé à Taiwan.
GB2312	Chinois simplifié, officiel.
BIG5-HKSCS	Big5 avec les extensions de Hong Kong, chinois traditionnel.
Shift_JIS	Japanais
EUCJP	Japanais

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`, `htmlentities` et `nl2br`.

7.102.25 implode() : Rassemble les éléments d'un tableau en une chaîne

string **implode** (string glue , array pieces) 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 `implode`

```
<?php
$array = array('nom', 'email', 'telephone');
$comma_separated = implode(",", $array);

print $comma_separated;
// nom,email,telephone
?>
```

Note

[implode](#) peut, pour des raisons historiques, accepter les paramètres dans un sens ou dans l'autre. Pour des raisons de cohérence avec [explode](#), toutefois, il est préférable d'utiliser l'ordre des arguments tels que documenté.

Note

Depuis ^{PHP} 4.3.0, le paramètre the *glue* de [implode](#) est optionnel, et par défaut, il vaut la chaîne vide ''.

Voir aussi [explode](#) et [split](#).

7.102.26 join() : Rassemble les éléments d'un tableau en une chaîne

```
string join ( string glue , array pieces )
```

[join](#) est un alias de [implode](#).

Voir aussi [explode](#), [implode](#) et [split](#).

7.102.27 levenshtein() : Calcule la distance Levenshtein entre deux chaînes

```
int levenshtein ( string str1 , string str2 ) int levenshtein ( string str1 , string str2 , int cost_ins , int cost_rep , int cost_del ) int levenshtein ( string str1 , string str2 , function cost )
```

[levenshtein](#) calcule la distance Levenshtein entre deux chaînes de caractères. Elle retournera -1 si l'un des deux arguments contient plus de 255 caractères (cela devrait être plus que suffisant pour faire des comparaisons dans un dictionnaire ou annuaire, et personne de sérieux ne fera de comparaison génétique en ^{PHP}).

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.

La troisième variante (qui n'est pas implémentée actuellement), est la version la plus générale, mais la plus lente. Elle appellera une fonction utilisateur qui déterminera le coût de chaque opération.

La fonction utilisateur qui sera appelée reçoit les arguments suivants :

- Opération à réaliser : 'I', 'R' ou 'D'
- Caractère dans ^{str1}
- Caractère dans ^{str2}
- Position dans ^{str1}
- Position dans ^{str2}

- Caractères restants dans `str1`
- Caractères restants dans `str2`

Cette fonction doit retourner un entier positif, qui représente le coût de cette opération particulière. Il peut ne prendre en compte que certains des paramètres fournis.

Grâce à cette fonction utilisateur, il est possible de prendre en compte la pertinence ou la valeur des caractères eux-mêmes, ou encore le contexte, pour définir le coûts d'une insertion, d'un effacement ou d'un remplacement. Cela se fait en perdant toutes les optimisations faites en terme d'exploitation du CPU et des buffers.

Voir aussi [soundex](#), [similar text](#) et [metaphone](#).

7.102.28 localeconv() : Lit la configuration locale

array **localeconv** (void)

[localeconv](#) retourne un tableau associatif contenant les informations de formats localisée 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
<code>decimal_point</code>	Séparateur décimal
<code>thousands_sep</code>	Séparateur de milliers
<code>grouping</code>	Tableau contenant les regroupements numériques
<code>int_curr_symbol</code>	Symbole monétaire international (i.e. EUR)
<code>currency_symbol</code>	Symbole monétaire local (i.e. €)
<code>mon_decimal_point</code>	Séparateur décimal monétaire
<code>mon_thousands_sep</code>	Séparateur de milliers monétaires
<code>mon_grouping</code>	Tableau contenant les regroupements numériques monétaires
<code>positive_sign</code>	Signe des valeurs positives
<code>negative_sign</code>	Signe des valeurs négatives
<code>int_frac_digits</code>	Nombre international de décimales
<code>frac_digits</code>	Nombre local de décimales
<code>p_cs_precedes</code>	TRUE si le <code>currency_symbol</code> précède une valeur positive et FALSE s'il lui succède.
<code>p_sep_by_space</code>	TRUE si un espace sépare <code>currency_symbol</code> d'une valeur positive, et FALSE sinon.
<code>n_cs_precedes</code>	TRUE si <code>currency_symbol</code> précède une valeur négative, et FALSE s'il lui succède.
<code>n_sep_by_space</code>	TRUE si un espace sépare <code>currency_symbol</code> d'une valeur négative, et FALSE sinon.
<code>p_sign_posn</code>	• 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
<code>n_sign_posn</code>	• 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 de regroupements contiennent des tableaux qui définissent la manière dont les nombres doivent être regroupés. Par exemple, le champ de regroupement pour `en_US` 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
setlocale(LC_ALL, "en_US");

$locale_info = localeconv();

echo "<PRE>\n";
echo "-----\n";
echo "  Informations monétaires pour la configuration locale : \n";
echo "-----\n";

echo "int_curr_symbol:    {$locale_info["int_curr_symbol"]}\n";
echo "currency_symbol:    {$locale_info["currency_symbol"]}\n";
echo "mon_decimal_point:  {$locale_info["mon_decimal_point"]}\n";
echo "mon_thousands_sep:  {$locale_info["mon_thousands_sep"]}\n";
echo "positive_sign:      {$locale_info["positive_sign"]}\n";
echo "negative_sign:      {$locale_info["negative_sign"]}\n";
echo "int_frac_digits:    {$locale_info["int_frac_digits"]}\n";
echo "frac_digits:        {$locale_info["frac_digits"]}\n";
echo "p_cs_precedes:      {$locale_info["p_cs_precedes"]}\n";
echo "p_sep_by_space:      {$locale_info["p_sep_by_space"]}\n";
echo "n_cs_precedes:      {$locale_info["n_cs_precedes"]}\n";
echo "n_sep_by_space:      {$locale_info["n_sep_by_space"]}\n";
echo "p_sign_posn:        {$locale_info["p_sign_posn"]}\n";
echo "n_sign_posn:        {$locale_info["n_sign_posn"]}\n";
echo "</PRE>\n";
?>
```

La constante `CHAR_MAX` est aussi définie pour être utilisée comme indiqué ci-dessus.

Voir aussi [setlocale](#) .

7.102.29 ltrim() : Supprime les caractères invisibles 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.
- " " (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 = "      Voici quelques mots :) ... ";
$trimmed = ltrim($text);
// $trimmed = "Voici quelques mots :) ... "
$trimmed = ltrim($text, " ");
// $trimmed = "Voici quelques mots :) ... "
$clean = ltrim($binary, "\0x00..\0x1F");
// 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](#).

7.102.30 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. Si le paramètre optionnel [raw_output](#) vaut `TRUE`, le digest MD5 est retourné au format binaire brut, avec une taille de 16.

Note
Le paramètre optionnel raw_output a été ajouté en ^{PHP} 5.0.0 et vaut par défaut <code>FALSE</code> .

[md5_file](#) a la même utilité que la commande en ligne md5sum.

Voir aussi [md5](#), [crc32](#) et [sha1_file](#).

7.102.31 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 <code>FALSE</code> .

Exemple avec md5
<pre><?php \$str = 'pomme'; if (md5(\$str) === 'ede0f9c3a1d2093e3f48fcafd3c70915') { echo "Voulez vous une golden ou une spartan?"; exit; } ?></pre>

Voir aussi [crc32](#), [md5_file](#) et [sha1](#).

7.102.32 metaphone() : Calcule la clé métaphone

```
string metaphone ( string str )
```

[metaphone](#) calcule la clé métaphone 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 métaphone 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 développé par Lawrence Philips <lphilips@verity.com>. Cette méthode est décrite dans le livre ["Practical Algorithms for Programmers", Binstock & Rex, Addison Wesley, 1995].

7.102.33 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 `strmon()` de la librairie C, à la différence près que cette implémentation ne convertit qu'un nombre à la fois.

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 définit dans la configuration locale).

`+ ou (`

Specifie 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 champs

`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

specification 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 supprimées. 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 longs, 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 [sscanf](#) .

7.102.34 nl_langinfo() : Rassemble des informations sur la langue et la configuration locale.

```
string nl_langinfo ( int item )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.102.35 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 '
'.

Voir aussi [htmlspecialchars](#) , [htmlentities](#) , [wordwrap](#) et [str_replace](#) .

7.102.36 number_format() : Formate un nombre pour l'affichage

```
string number_format ( float number , int decimals ) 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.

Note

Seul le premier caractère du paramètre `thousands_sep` est utilisé. Par exemple, si vous utilisez `foo` comme séparateur de milliers, sur le nombre `1000`, `number_format` retournera `1f000`.

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 :

Exemple avec `number_format`

```
<?php
$nombre = 1234.56;
// Notation anglaise (par défaut)
$english_format_number = number_format($nombre);
// 1,234.56
// Notation française
$nombre_format_francais = number_format($nombre, 2, ',', ' ');
// 1 234,56
?>
```

Voir aussi `sprintf`, `printf` et `sscanf`.

7.102.37 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`.

7.102.38 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, en méthode GET ou POST. 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.

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` , `set_magic_quotes_runtime` et `urldecode` .

7.102.39 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 du langage). Cela fait que vous n'êtes pas obligés 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("clef" => "foo");

print "c'est {$bar['clef']} !" ; // c'est foo !

// Les guillemets simples annulent le remplacement des variables
print 'foo vaut $foo'; // foo vaut $foo

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 FAQs 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` .

7.102.40 printf() : Affiche une chaîne de caractères formatée

```
void printf ( string format , mixed args )
```

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.

Voir aussi print, sprintf, sscanf, fscanf et flush.

7.102.41 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 en format binaire 8 bits. Cette fonction est similaire à imap_qprint, hormis le fait qu'elle n'exige pas le module IMAP, pour fonctionner.

7.102.42 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 : `. \ + * ? [ ^ ] ( $ )`

Voir aussi addslashes, htmlentities, htmlspecialchars, nl2br et stripslashes.

7.102.43 rtrim() : Supprime les espaces 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

```
<?php
$text = "      Voici quelques mots :) ... ";
$trimmed = rtrim($text);
// $trimmed = "      Voici quelques mots :) ..."
```

```
$trimmed = rtrim($text, "    .");
// $trimmed = "    Voici quelques mots :)"
$clean = rtrim($binary, "\0x00..\0x1F");
// 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 [ltrim](#).

7.102.44 setlocale() : Modifie les informations de localisation

string **setlocale** (mixed category , string locale , string ...) string **setlocale** (mixed 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.

category est une constante (ou une chaîne) qui spécifie la catégorie de fonction 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](#)

Si ^{locale} est la chaîne vide "", les noms de locales seront pris dans l'environnement, à partir des variables de même noms 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.

Note

La possibilité de passer plusieurs locales est disponible depuis ^{PHP} 4.3.0.

[setlocale](#) retourne la valeur de la locale courante, ou bien ^{FALSE} si la fonctionnalité de localisation n'est pas implémentée sur votre plate-forme, si la locale spécifiée n'existe pas, ou si la catégorie n'existe pas. Une catégorie invalide causera aussi l'affichage d'un message d'alerte.

Si ^{locale} vaut zéro ou "0", la configuration de localisation ne sera pas modifiée, et seule la valeur courante sera retournée.

Note

Passer plusieurs locales n'était pas disponible avant ^{PHP} 4.3.0.

[setlocale](#) retourne la nouvelle valeur locale, ou bien ^{FALSE} si la fonctionnalité n'est pas implantée sur votre plate forme, que la configuration spécifiée n'existe pas ou que le nom de catégorie est invalide. Un nom de catégorie invalide causera aussi l'affichage d'une alerte. Les noms de catégories et de configurations locales sont disponibles dans la [RFC 1766](#) et [ISO 639](#).

Note

La valeur retournée par `setlocale` dépend du système d'exploitation qui fait tourner ^{PHP}. La valeur retournée est exactement celle que la fonction `setlocale()` du système retourne.

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
<TT>PHP</TT>
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
<TT>PHP</TT>
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'";
?>
```

7.102.45 sha1_file() : Calcule le sha1 d'un fichier

string **sha1_file** (string filename , bool raw_output)

sha1_file 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. En cas d'erreur, `FALSE` est retourné.

Voir aussi [sha1](#), [crc32](#) et [md5_file](#)

7.102.46 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.

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](#)

7.102.47 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 grand 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.

7.102.48 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](#) .

7.102.49 sprintf() : Retourne une chaîne formatée

```
string sprintf ( string format , mixed args )
```

sprintf retourne une chaîne formatée, avec le format `format` et 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 passé 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 **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.
2. 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 est aligné à droite.
3. Un nombre optionnel, **spécificateur de taille** indique le nombre minimum de caractères que cette conversion doit fournir en résultat.
4. Un **spécificateur de précision** qui indique le nombre de décimales qui doivent être affichées pour les nombres à virgule flottante. Cette option n'a pas d'effet pour les autres types que ^{float}. Une autre fonction pratique pour formater les nombres à virgule flottante est `number_format`.
5. 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.
 - ◆ 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.
 - ◆ 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);
?>
```

Ce exemple peut afficher par exemple : "Il y a 5 signes dans le bananier". Mais imaginez que la chaîne de format soit créée dans un script séparé (une librairie), ca nous devons internationaliser cet affichage. Nous aurons peut être à écrire :

Echange d'arguments (2)

```
<?php
$format = "Le %s a %d singes";
printf($format,$num,$location);
?>
```


Maintenant, nous avons un problème. L'ordre des arguments a été changé, et ne correspond pas à l'ordre des arguments dans le script ^{PHP}. Nous souhaitons laisser le code ^{PHP} tel quel, et simplement indiquer dans la chaîne de formatage l'ordre dans lequel les arguments doivent être repris. La chaîne de format peut être réécrite ainsi :

Echange d'arguments (3)

```
<?php
$format = "Le %2\$s a %1\$d signes";
printf($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
$format = "Le %2\$s a %1\$d singes.
          C'est un beau %2\$s, avec %1\$d signes.";
printf($format, $num, $location);
?>
```

Voir aussi printf , sscanf , fscanf et number format .

7.102.50 sscanf() : Analyse une chaîne à l'aide d'un format

mixed **sscanf** (string str , string format , string var1)

sscanf est l'inverse de la fonction printf . sscanf lit des données dans la chaîne ^{str} , et l'interprète en fonction du format ^{format} . Si seuls deux paramètres sont fournis, les valeurs trouvées seront retournées sous forme de tableau. Sinon, les valeurs seront placées dans les arguments ^{var1} .

Tous les caractères blancs dans la chaîne ^{format} correspondent à un caractère blanc dans la chaîne ^{str} . Cela signifie que même une tabulation 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
$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. Les paramètres optionnels doivent être passés par référence.

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 %s %s", &$id, &$first, &$last);
echo "<author id=\"$id\">
      <firstname>$first</firstname>
      <surname>$last</surname>
</author>\n";
?>
```

Voir aussi fscanf , printf et sprintf .

7.102.51 `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 complexe, il est recommandé d'utiliser cette fonction qui est plus rapide que `ereg_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 seront retournées dans le paramètre `count` passé par référence.

Voir aussi `str_replace`, `ereg_replace`, `preg_replace` et `strtr`.

7.102.52 `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
$input = "Alien";
print str_pad($input, 10); // affiche "Alien   "
print str_pad($input, 10, "-", STR_PAD_LEFT); // affiche "---Alien"
print str_pad($input, 10, "_", STR_PAD_BOTH); // affiche "__Alien__"
?>
```

7.102.53 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);
?>
```

Ceci affichera "-- -- -- -- --".

Voir aussi `str_pad` et `substr_count`.

7.102.54 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`.

En PHP 4.0.5 et plus récent, tous les paramètres de `str_replace` peuvent être des tableaux.

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`.

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 bière 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", "bières", "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
?>
```

`str_replace` est compatible avec les chaînes binaires.

Note

`str_replace` a été ajoutée en ^{PHP} 3.0.6, mais était buggée jusqu'en ^{PHP} 3.0.8.

Note

Cette fonction gère les chaînes binaires

Note

Depuis ^{PHP} 5.0, le nombre de valeurs de `needle` trouvées et remplacées seront retournées dans le paramètre `count` passé par référence.

Voir aussi `str_ireplace`, `substr_replace`, `ereg_replace`, `preg_replace` et `strtr`.

7.102.55 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.

7.102.56 str_shuffle() : Mélange les caractères d'une chaîne de caractères

string **str_shuffle** (string str)

`str_shuffle` mélange les caractères d'une chaîne de caractères. Une permutation de tous les caractères est créée.

Exemple `str_shuffle`

```
<?php
$str = 'abcdef';
$shuffled = str_shuffle($str);

// Cela va afficher une ligne comme par exemple : bfdaec
print $shuffled;
?>
```

Voir aussi `shuffle` et `rand`.

7.102.57 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 une taille d'un caractère.

`FALSE` est retourné 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'affichage donnera :

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 [preg_split](#) , [split](#) , [count_chars](#) et [str_word_count](#) .

7.102.58 str_word_count() : Compte le nombre de mots utilisés dans une chaîne

mixed **str_word_count** (string string , int format)

[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` et le tableau retourné sont :

- 1 – retourne un tableau contenant tous les mots trouvés dans la chaîne `string` .
- 2 – retourne un tableau associatif, où les clés sont représentent la position du mot en regard, dans la chaîne `string` .

Dans cette fonction, la notion de 'mot' est dépendant de la configuration de localisation. C'est une chaîne qui contient tous les caractères alphabétiques, et qui peut contenir, mais pas commencer par "" and "-".

Exemples d'utilisation de str_word_count

```

<?php
$str = "Salut l'ami, vous
      avez      une belle mine!";

$a  = str_word_count($str, 1);
$b  = str_word_count($str, 2);
$c  = str_word_count($str);

print_r($a);
print_r($b);
print $c;

/* Le résultat est :
Array
(
    [0] => Salut
    [1] => l'ami
    [2] => vous
    [3] => avez
    [4] => une
    [5] => belle
    [6] => mine
)
Array
(
    [0] => Salut
    [6] => l'ami
    [13] => vous
    [27] => avez
    [41] => une
    [45] => belle
    [51] => mine
)
7
*/
?>

```

Voir aussi [explode](#) , [preg_split](#) , [split](#) , [count_chars](#) et [substr_count](#) .

7.102.59 strcasecmp() : Comparaison 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.

Notez que cette comparaison est insensible à la casse.

Exemple avec [strcasecmp](#)

```

<?php
$var1 = "Hello";
$var2 = "hello";
if (strcasecmp($var1, $var2) == 0) {
    echo "$var1 est égale à $var2 (comparaison insensible à la casse)";
}
?>

```

Vior aussi [ereg](#) , [strcmp](#) , [substr](#) , [stristr](#) , [strncasecmp](#) et [strstr](#) .

7.102.60 strchr() : Trouve la première occurrence d'un caractère dans une chaîne

```
string strchr ( string haystack , string needle )
```

[strchr](#) est un alias de [strstr](#) .

7.102.61 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 .

7.102.62 strcoll() : Comparaison de chaînes localisée

```
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.

Voir aussi ereg , strcmp , strcasecmp , substr , stristr , strncasecmp , strncmp , strstr et setlocale .

7.102.63 strcspn() : Trouve un segment de chaîne ne contenant pas certains caractères

```
int strcspn ( string str1 , string str2 )
```

strcspn retourne la taille du premier segment de ^{str1} qui ne contient **aucun** des caractères de ^{str2} .

Voir aussi strspn .

7.102.64 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.

Exemple avec <u>strip_tags</u>
<pre><?php \$string = strip_tags(\$string, '<a><i><u>'); ?></pre>

Attention

`strip_tags` ne modifie pas les attributs des balises que vous autorisez via le paramètre `allowable_tags`, y compris le style et les `onmouseover`, que des utilisateurs mal intentionnés peuvent utiliser.

7.102.65 stripslashes() : Supprime les anti-slash d'une chaîne C

```
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`.

7.102.66 strpos() : Recherche la position d'une occurrence dans une chaîne, sans tenir compte de la casse

```
int strpos ( string haystack , string needle , int offset )
```

`strpos` retourne la position numérique de la première occurrence de `needle` dans la chaîne `haystack`. Contrairement à `strpos`, `strpos` 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, `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 exacte de cette fonction.

Exemples avec `strpos`

```
<?php
$findme = 'a';
$mystring1 = 'xyz';
$mystring2 = 'ABC';

$pos1 = strpos($mystring1, $findme);
$pos2 = strpos($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 '$mystring1'";
}

// 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 '$mystring1'";
    echo " et à la position $pos2";
}
?>
```

Si `needle` n'est pas une chaîne, elle est convertie 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 [strpos](#) , [strrpos](#) , [strchr](#) , [substr](#) , [stristr](#) , [strstr](#) et [stri_replace](#) .

7.102.67 stripslashes() : Supprimer 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](#) .

Exemple avec [stripslashes](#)

```
<?php
$str = "Avez vous \\\'oreille dure?";

// Affiche : Avez vous l'oreille dure?
echo stripslashes($str);
?>
```

Voir aussi [addslashes](#) .

7.102.68 stristr() : Trouve la première occurrence dans une chaîne (insensible à la casse)

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';
$domaine = stristr($email, 'e');
print $domaine;

// Affiche ER@EXAMPLE.com
?>
```

Voir aussi [strchr](#) , [strrchr](#) , [substr](#) et [ereg](#) .

7.102.69 strlen() : Calcule la taille d'une chaîne

int **strlen** (string str)

[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](#) .

7.102.70 `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](#) .

7.102.71 `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);
?>
```

Le code ci-dessus générera ce qui suit :

```
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 .

7.102.72 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 .

7.102.73 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} . Si l'une des deux chaînes est plus petite que ^{len} , alors la longueur de cette chaîne sera utilisée.

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 .

7.102.74 strpos() : Trouve la position d'un caractère dans une chaîne

```
int strpos ( string haystack , string needle , int offset )
```

strpos retourne la position numérique de la première occurrence de ^{needle} dans la chaîne de caractères ^{haystack} . Contrairement à la fonction strrpos , 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 exacte de cette fonction.

Exemple avec `strpos`

```
<?php
$mystring = 'abc';
$findme = 'a';
$pos = strpos($mystring, $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 '$mystring'";
} else {
    echo "La chaîne '$findme' a été trouvée dans la chaîne '$mystring'";
    echo " et à la position $pos";
}
?>
```

Si `needle` n'est pas une chaîne, elle est convertie 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 `strstr`, `stripos`, `strripos`, `strchr`, `substr`, `stristr` et `strstr`.

7.102.75 strrchr() : Retourne la fin de la 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.

Si `needle` n'est pas une chaîne, elle est convertie 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, "\n"), 1);
?>
```

Voir aussi `strchr`, `substr`, `stristr` et `strstr`.

7.102.76 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 e'l ruojnoB"
?>

```

7.102.77 stripslashes() : Trouve la position de la dernière occurrence d'une chaîne dans une autre de façon insensible à la casse.

```

int stripslashes ( string haystack , string needle )

```

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.

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 exacte de cette fonction.

Un exemple simple avec `stripos`

```

<?php
$haystack = 'ababcd';
$needle   = 'aB';

$pos      = stripslashes($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)";
}

/* Outputs:
    Félicitations !
    Nous avons trouvé le dernier (aB) dans (ababcd) à la position (2)
*/
?>

```

Voir aussi `strrpos`, `strchr`, `substr` et `stristr`.

7.102.78 strrpos() : Trouve la position de la dernière occurrence d'un caractère dans une chaîne

```

int strrpos ( string haystack , char needle )

```

`strrpos` retourne la position numérique de la dernière occurrence du caractère `needle` dans la chaîne `haystack`. Notez que `needle` doit être un caractère unique. Si une chaîne est passée dans ce paramètre, seul le premier caractère sera utilisé.

Si `needle` n'est pas trouvé, `strrpos` retournera `FALSE`.

Note

Il est facile de confondre le résultat "caractère trouvé à la position 0" et "caractère non trouvé". Voici comment faire la différence :

Exemple avec strpos

```
<?php
// en
<TT>PHP</TT>
4.0b3 et plus récent :
    $pos = strpos($machaine, "b");
    if ($pos === false) { // note : trois égaux
        // pas trouvé
    }
// en versions plus anciennes que 4.0b3 :
    $pos = strpos($machaine, "b");
    if (!is_integer($pos)) {
        // pas trouvé
    }
?>
```

Si `needle` n'est pas une chaîne, elle est convertie en entier, et utilisée comme caractère de code ASCII correspondant.

Voir aussi strpos, stripos, strrchr, substr, strstr et strstr.

7.102.79 strpos() : Trouve le premier segment de chaîne

```
int strpos ( string str1, string str2 )
```

strpos retourne la taille du segment initial de la chaîne `str1` qui est entièrement constitué de caractères contenus dans `str2`.

Exemple avec strpos

```
<?php
echo strpos("421 est la réponse, mais quelle est la question ...", "1234567890");
?>
```

va afficher 3, car la chaîne "421" est le plus long segment initial de "421 est la réponse, mais quelle est la question ...", dont tous les caractères sont contenus dans "1234567890".

Voir aussi strcspn.

7.102.80 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

```
<?php
```

```
$email = 'user@example.com';
$domaine = strstr($email, '@');
print $domaine;
// Affiche @example.com
?>
```

Voir aussi [ereg](#) , [preg_match](#) , [strpos](#) , [strchr](#) , [stristr](#) , [strrpos](#) , [strrchr](#) et [substr](#) .

7.102.81 strtok() : Coupe une chaîne en segments

string **strtok** (string arg1 , string arg2)

[strtok](#) coupe la chaîne ^{arg1} en segments, chaque segment étant délimité par ^{arg2} . Par exemple, si vous avez une chaîne telle que "Voici un exemple", vous pouvez en extraire les différents mots en utilisant cette fonction :

Exemple avec [strtok](#)

```
<?php
$string = "Ceci est un exemple\ninteressant";
/* Utilisez aussi les nouvelles lignes et les tabulations comme séparateur de mots */
$tok = strtok($string, " \n ");
while ($tok) {
    echo "Mot = $tok<br>";
    $tok = strtok(" \n ");
}
?>
```

Notez que seul le premier appel à [strtok](#) nécessite les deux arguments. Tous les appels ultérieurs à [strtok](#) ne nécessitent que la chaîne à découper. Pour initialiser à nouveau [strtok](#) , ou pour recommencer, fournissez à nouveau le paramètre ^{arg2} . La chaîne ^{arg1} sera découpé dès que l'un des caractères de ^{arg2} 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](#)

```
$first_token = strtok('/chose', '/');
$second_token = strtok('/');
var_dump ($first_token, $second_token);

/* Affiche :
string(0) ""
string(9) "chose"
*/
```

Nouveau comportement de [strtok](#)

```
$first_token = strtok('/chose', '/');
$second_token = strtok('/');
var_dump ($first_token, $second_token);

/* Affiche :
string(9) "chose"
bool(false)
*/
```

De plus, soyez prudents avec les séparateurs qui sont égaux à "0". Ces valeurs s'évaluent comme ^{FALSE} dans les expression conditionnelles.

Voir aussi [split](#) et [explode](#) .

7.102.82 strtolower() : Renvoie une chaîne en minuscules

string **strtolower** (string str)

strtolower retourne string , après avoir converti tous les caractères alphabétique 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);
print $str;
// Affiche : marie a un petit agneau, et l'aime très fort.
?>
```

Voir aussi strtoupper , ucfirst et ucwords .

7.102.83 strtoupper() : Renvoie une chaîne en majuscules

string **strtoupper** (string string)

strtoupper retourne string , après avoir converti tous les caractères alphabétique 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 TRÈS fORT.";
$str = strtoupper($str);
print $str;
// Affiche : MARIE A UN PETIT AGNEAU, ET L'AIME TRES FORT.
?>
```

Voir aussi strtolower , ucfirst et ucwords .

7.102.84 strtr() : Remplace des chaînes dans une chaîne

string **strtr** (string str , string from , string to) string **strtr** (string str , array replace_pairs)

strtr 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 strtr

```
<?php
$addr = strtr($addr, "ääö", "aao");
?>
```


`strtr` peut être appelé 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`. `strtr` commence toujours par rechercher la chaîne la plus longue, et ne travaille pas sur des segments qu'elle a déjà modifié.

Exemple avec `strtr` avec deux arguments

```
<?php
$trans = array("Bonjour" => "salut", "salut" => "bonjour");
echo strtr("Bonjour à tous, j'ai dit bonjour", $trans);
?>
```

Cet exemple va afficher :

Salut à tous, j'ai dit bonjour

Note

Les paramètres optionnels `to` et `from` ont été ajoutés en PHP 4.0.0.

Voir aussi [ereg_replace](#).

7.102.85 substr_count() : Compte de le nombre d'occurrences de segments dans une chaîne

`int substr_count ( string haystack , string needle )`

`substr_count` retourne le nombre d'occurrences de `needle` dans la chaîne `haystack`.

Exemple avec `substr_count`

```
<?php
print substr_count("Ceci est un test", "ci");
// affiche 1
?>
```

7.102.86 substr_replace() : Remplace un segment dans une chaîne

`string substr_replace ( string 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`. Le résultat 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<br>\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";
?>

```

Voir aussi [str_replace](#) et [substr](#) .

7.102.87 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
$rest = substr("abcdef", 1); // retourne "bcdef"
$rest = substr("abcdef", 1, 3); // retourne "bcd"
$rest = substr("abcdef", 0, 4); // retourne "abcd"
$rest = substr("abcdef", 0, 8); // retourne "abcdef"

// Accéder aux caractères via les accolades
$string = 'abcdef';
echo $string{0}; // retourne a
echo $string{3}; // retourne d
?>

```

Si start est négatif, la chaîne retournée commencera au caractère numéro start dans 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 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 de 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 [strchr](#), [substr_replace](#), [ereg](#) et [trim](#).

7.102.88 trim() : Supprime les espaces 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` a été 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.

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 [trim](#)

```
<?php
$text = "      Voici quelques mots :) ... ";
$trimmed = trim($text);
// $trimmed = "Voici quelques mots :) ..."
$trimmed = trim($text," .");
// $trimmed = "Voici quelques mots :)"
$clean = trim($binary,"\x00..\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 [ltrim](#) et [rtrim](#).

7.102.89 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 convertis.

Exemple avec [ucfirst](#)

```
$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](#) .

7.102.90 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.

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

La définition d'un mot est : toute séquence de caractères qui suit immédiatement un caractère blanc (espace, tabulation, nouvelle ligne, retour chariot, tabulation verticale).

Voir aussi [strtoupper](#) , [strtolower](#) et [ucfirst](#) .

7.102.91 vprintf() : Affiche une chaîne formatée

void **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.

Voir aussi [printf](#) , [sprintf](#) et [vsprintf](#)

7.102.92 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](#) , [vsprintf](#) et [vprintf](#)

7.102.93 wordwrap() : Effectue la césure d'une chaîne

string **wordwrap** (string *str* , int *width* , string *break* , boolean *cut*)

wordwrap retourne la chaîne *str* , après avoir inséré *break* tous les *width* caractères.

Par défaut, wordwrap va automatiquement insérer une nouvelle ligne tous les 75 caractères, si *width* et *break* ne sont pas fournis.

Si le paramètre *cut* est mis à 1, la chaîne sera toujours césurée à la taille *width* . Si vous avez un mot qui est plus long que la taille de césure, il sera coupé en morceaux.

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";
?>
```

Cet exemple va afficher :

```
Portez ce vieux
whisky au juge
blond qui fume.
```

Exemple avec wordwrap avec césure

```
<?php
$text = "Un mot tres tres loooooooooooooooooooooong.";
$newtext = wordwrap( $text, 8, "\n", 1);

echo "$newtext\n";
?>
```

Cet exemple va afficher :

```
Un mot
tres
tres
loooooooo
ooooooooo
ooooong.
```

Voir aussi [nl2br](#) .

7.103 Sybase

7.103.1 Introduction

7.103.2 Pré-requis

7.103.3 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`.

7.103.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Sybase

Nom	Par défaut	Modifiable
<code>sybase.allow_persistent</code>	"On"	PHP_INI_SYSTEM
<code>sybase.max_persistent</code>	"-1"	PHP_INI_SYSTEM
<code>sybase.max_links</code>	"-1"	PHP_INI_SYSTEM
<code>sybase.interface_file</code>	"/usr/sybase/interfaces"	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>	"Off"	PHP_INI_SYSTEM
<code>magic_quotes_sybase</code>	"Off"	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.

Sybase-CT

Nom	Par défaut	Modifiable
<code>sybct.allow_persistent</code>	"On"	PHP_INI_SYSTEM
<code>sybct.max_persistent</code>	"-1"	PHP_INI_SYSTEM
<code>sybct.max_links</code>	"-1"	PHP_INI_SYSTEM
<code>sybct.min_server_severity</code>	"10"	PHP_INI_ALL
<code>sybct.min_client_severity</code>	"10"	PHP_INI_ALL
<code>sybct.hostname</code>	NULL	PHP_INI_ALL

Voici un éclaircissement sur l'utilisation des directives de configuration.

`sybct.allow_persistent` booléen

Whether to allow persistent Sybase-CT connections. The default is 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 librairie 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.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 [ini_set](#).

7.103.5 Types de ressources

7.103.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [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](#)

7.103.8 sybase_close() : Ferme une connexion Sybase

bool **sybase_close** (resource link_identifiant)

[sybase_close](#) termine la connexion avec le serveur Sybase associé à l'identifiant de connexion `link_identifiant`.

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](#).

7.103.9 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](#) .

7.103.10 sybase_data_seek() : Déplace le pointeur interne de lignes Sybase

`bool sybase_data_seek ( resource result_identif , int row_number )`

[sybase_data_seek](#) déplace le pointeur interne de ligne du résultat Sybase associé à `result_identif` 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](#) .

7.103.11 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.

-1	Essai jusqu'à réussir (par défaut)
0	Ne pas reessayer
n	Reessaye n fois

Note

Cette fonction n'est disponible qu'avec la librairie CT Sybase, et non pas avec la librairie DB.

7.103.12 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 nom des champs comme clés.

Il est très important de noter que [sybase_fetch_array](#) N'est PAS nettement plus lent que [sybase_fetch_row](#), tandis qu'elle fournit un confort d'utilisation notable.

Cas des colonnes de noms identiques avec Sybase

```
<?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);
?>
```

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](#).

7.103.13 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` si il n'y a plus de lignes.

[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 sur 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](#).

7.103.14 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](#).

7.103.15 `sybase_fetch_object()` : Retourne une ligne Sybase sous la forme d'un objet

`object 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`.

Note

Depuis ^{PHP} 4.3.0, cette fonction ne retourne plus les membres numériques.

Ancienne méthode :

```
object(stdclass)(3) {
  [0]=>
  string(3) "foo"
  ["foo"]=>
  string(3) "foo"
  [1]=>
  string(3) "bar"
  ["bar"]=>
  string(3) "bar"
}
```

Nouvelle méthode :

```
object(stdclass)(3) {
  ["foo"]=>
  string(3) "foo"
  ["bar"]=>
  string(3) "bar"
}
```

[sybase_fetch_object](#) retourne l'objet Foo

```
<?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());
// {...}
?>
```

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](#).

7.103.16 `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 champs 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](#).

7.103.17 `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](#).

7.103.18 `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. Mais, si vous êtes sûr de ne pas avoir besoin du résultat ultérieurement.

7.103.19 `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](#).

7.103.20 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 librairie CT Sybase, et non pas avec la librairie DB.

Voir aussi sybase_min_server_severity .

7.103.21 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.

Voir aussi sybase_min_message_severity .

7.103.22 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.

Voir aussi sybase_min_error_severity .

7.103.23 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 librairie CT Sybase, et non pas avec la librairie DB.

Voir aussi sybase_min_client_severity .

7.103.24 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](#) .

7.103.25 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](#) .

7.103.26 sybase_pconnect() : Ouvre une connexion persistante à un serveur Sybase

```
int sybase_pconnect ( string servername , string username , string password , string charset )
```

[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](#) .

7.103.27 sybase_query() : Envoie une requête à une base Sybase

```
resource sybase_query ( string query , resource link_identifiant )
```

[sybase_query](#) retourne un identifiant de résultat positif en cas de succès, et `FALSE` sinon.

[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](#) .

7.103.28 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](#).

7.103.29 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](#).

7.103.30 sybase_set_message_handler() : Configure le gestionnaire de message Sybase

```
bool sybase_set_message_handler ( callback handler )
```

[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.

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.

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 si 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');
?>

```

7.103.31 `sybase_unbuffered_query()` : Envoie une requête à Sybase et ne bloque pas

resource **sybase_unbuffered_query** (string query , resource link_identifier)

`sybase_unbuffered_query` retourne une ressource Sybase result en cas de réussite, et `FALSE` sinon.

`sybase_unbuffered_query` envoie la requête `query` au serveur Sybase actif, représenté par `link_identifier`. Si `link_identifier` 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 ramenés 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);
?>

```


7.104 Analyseur de code PHP

7.104.1 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 .

7.104.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.104.3 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 librairie supplémentaire pour disposer de ces fonctions.

Note

Le support automatique du tokenizer est disponible depuis PHP 4.3.0.

7.104.4 Constantes prédefinies

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.

```

T_INCLUDE (entier)
T_INCLUDE_ONCE (entier)
T_EVAL (entier)
T_REQUIRE (entier)
T_REQUIRE_ONCE (entier)
T_LOGICAL_OR (entier)
T_LOGICAL_XOR (entier)
T_LOGICAL_AND (entier)
T_PRINT (entier)
T_PLUS_EQUAL (entier)
T_MINUS_EQUAL (entier)
T_MUL_EQUAL (entier)
T_DIV_EQUAL (entier)
T_CONCAT_EQUAL (entier)
T_MOD_EQUAL (entier)
T_AND_EQUAL (entier)
T_OR_EQUAL (entier)

```

T_XOR_EQUAL (*entier*)
 T_SL_EQUAL (*entier*)
 T_SR_EQUAL (*entier*)
 T_BOOLEAN_OR (*entier*)
 T_BOOLEAN_AND (*entier*)
 T_IS_EQUAL (*entier*)
 T_IS_NOT_EQUAL (*entier*)
 T_IS_IDENTICAL (*entier*)
 T_IS_NOT_IDENTICAL (*entier*)
 T_IS_SMALLER_OR_EQUAL (*entier*)
 T_IS_GREATER_OR_EQUAL (*entier*)
 T_SL (*entier*)
 T_SR (*entier*)
 T_INC (*entier*)
 T_DEC (*entier*)
 T_INT_CAST (*entier*)
 T_DOUBLE_CAST (*entier*)
 T_STRING_CAST (*entier*)
 T_ARRAY_CAST (*entier*)
 T_OBJECT_CAST (*entier*)
 T_BOOL_CAST (*entier*)
 T_UNSET_CAST (*entier*)
 T_NEW (*entier*)
 T_EXIT (*entier*)
 T_IF (*entier*)
 T_ELSEIF (*entier*)
 T_ELSE (*entier*)
 T_ENDIF (*entier*)
 T_LNUMBER (*entier*)
 T_DNUMBER (*entier*)
 T_STRING (*entier*)
 T_STRING_VARNAME (*entier*)
 T_VARIABLE (*entier*)
 T_NUM_STRING (*entier*)
 T_INLINE_HTML (*entier*)
 T_CHARACTER (*entier*)
 T_BAD_CHARACTER (*entier*)
 T_ENCAPSED_AND_WHITESPACE (*entier*)
 T_CONSTANT_ENCAPSED_STRING (*entier*)
 T_ECHO (*entier*)
 T_DO (*entier*)
 T_WHILE (*entier*)
 T_ENDWHILE (*entier*)
 T_FOR (*entier*)
 T_ENDFOR (*entier*)
 T_FOREACH (*entier*)
 T_ENDFOREACH (*entier*)
 T_DECLARE (*entier*)
 T_ENDDECLARE (*entier*)
 T_AS (*entier*)
 T_SWITCH (*entier*)
 T_ENDSWITCH (*entier*)
 T_CASE (*entier*)
 T_DEFAULT (*entier*)

T_BREAK (*entier*)
 T_CONTINUE (*entier*)
 T_OLD_FUNCTION (*entier*)
 T_FUNCTION (*entier*)
 T_CONST (*entier*)
 T_RETURN (*entier*)
 T_USE (*entier*)
 T_GLOBAL (*entier*)
 T_STATIC (*entier*)
 T_VAR (*entier*)
 T_UNSET (*entier*)
 T_ISSET (*entier*)
 T_EMPTY (*entier*)
 T_CLASS (*entier*)
 T_EXTENDS (*entier*)
 T_OBJECT_OPERATOR (*entier*)
 T_DOUBLE_ARROW (*entier*)
 T_LIST (*entier*)
 T_ARRAY (*entier*)
 T_LINE (*entier*)
 T_FILE (*entier*)
 T_COMMENT (*entier*)
 T_ML_COMMENT (*entier*)
 T_OPEN_TAG (*entier*)
 T_OPEN_TAG_WITH_ECHO (*entier*)
 T_CLOSE_TAG (*entier*)
 T_WHITESPACE (*entier*)
 T_START_HEREDOC (*entier*)
 T_END_HEREDOC (*entier*)
 T_DOLLAR_OPEN_CURLY_BRACES (*entier*)
 T_CURLY_OPEN (*entier*)
 T_PAAMAYIM_NEKUDOTAYIM (*entier*)
 T_DOUBLE_COLON (*entier*)

7.104.5 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

```

<?php
$source = file_get_contents("un_script.php");
$tokens = token_get_all($source);
foreach ($tokens as $token) {
    if (is_string($token)) {
        // token d'un seul caractère
        echo $token;
    } else {
        // tableau de tokens
        list($id, $text) = $token;
        switch($id) {
            case T_COMMENT:
            case T_ML_COMMENT:
                // aucune action sur les commentaires
                break;
            default:
                // tout le reste doit être affiché
                echo $text;
                break;
        }
    }
}

```

```
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [token_get_all](#)
- [token_name](#)

7.104.7 token_name() : Lit le nom d'un élément de code source

string **token_name** (int token)

[token_name](#) retourne le nom de l'élément de code ^{PHP token}. Ce nom correspond au nom de la constante équivalente.

Exemple avec [token_name](#)

```
<?php
// 260 est la valeur correspondante à l'élément T_REQUIRE
echo token_name(260);          // -> "T_REQUIRE"

// Une constante d'élément correspond à son propre nom
echo token_name(T_FUNCTION); // -> "T_FUNCTION"
?>
```

7.105 URL

7.105.1 Introduction

Fonctions permettant de traiter les URL : encoder, décoder, et analyser.

7.105.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.105.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.105.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.105.5 Types de ressources

Cette extension ne définit aucune ressource.

7.105.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [base64_decode](#)
- [base64_encode](#)
- [get_meta_tags](#)
- [http_build_query](#)
- [parse_url](#)
- [rawurldecode](#)
- [rawurlencode](#)
- [urldecode](#)
- [urlencode](#)

7.105.8 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, grosso modo, 33% de plus que les données initiales.

Exemple avec base64_encode

```
<?php
$str = 'Ceci est une chaîne encodée';
echo base64_encode($str);
?>
```

Cet exemple va afficher :

Q2VjaSB1c3QgdW5lIGNoYWluZSB1bmNvZGVl

Voir aussi base64_decode , chunk_split et la RFC 2045 section 6.8.

7.105.9 get_meta_tags() : Extrait tous les méta tags d'un fichier

array **get_meta_tags** (string filename , int use_include_path)

get_meta_tags ouvre le fichier `filename` et l'analyse ligne par ligne à la recherche de balises <meta>, de la forme

Exemple de Meta Tags

```
<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.

En donnant la valeur de 1 au paramètre optionnel `use_include_path` fera que get_meta_tags recherchera aussi le fichier dans l'include path.

Résultat de get_meta_tags

```
<?php
// Supposons que les balises ci-dessus sont disponibles sur exemple.com
$tags = get_meta_tags('http://www.exemple.com/');

// Notez que les clés sont en minuscule, et
// le . a été remplacé par _ dans la clé
print $tags['author'];      // nom
print $tags['keywords'];    // documentation php
print $tags['description']; // un manuel PHP
print $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](#).

7.105.10 http_build_query() : Génère une chaîne 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 les index sont numériques, ils sont lus dans le tableau de base, et si `numeric_prefix` est fourni, il sera utilisé pour préfixer les noms des index pour les éléments du tableau de base. Cela permet de générer des noms de variables valides, si les données sont ensuite décodées par ^{PHP} ou une application ^{CGI}.

Exemple avec [http_build_query](#)

```
<?php
$data = array('foo'=>'bar',
              'baz'=>'boom',
              'cow'=>'milk',
              'php'=>'hypertext processor');

echo http_build_query($data);
/* Affiche :
    foo=bar&baz=boom&cow=milk&php=hypertext+processor
*/
?>
```

Exemple avec [http_build_query](#) et un index numérique

```
<?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
*/
?>
```

Exemple avec [http_build_query](#) et des tableaux complexes

```
<?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_');
/* Affiche : (word wrapped for readability)
    user[name]=Bob+Smith&user[age]=47&user[sex]=M&user[dob]=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: Only the numerically indexed element in the base array "CEO" received a prefix. The other numeric indices, found under

```

    */
    pastimes, do not require a string prefix to be legal variable names.
?>

```

Using [http_build_query](#) with an object

```

<?php
class myClass {
    var $foo;
    var $baz;

    function myClass() {
        $this->foo = 'bar';
        $this->baz = 'boom';
    }
}

$data = new myClass();

echo http_build_query($data);
/* Affiche :
    foo=bar&baz=boom
*/
?>

```

Voir aussi [parse_str](#), [parse_url](#), [urlencode](#) et [array_walk](#)

7.105.11 [parse_url\(\)](#) : Analyse une URL et retourne ses composants

array [parse_url](#) (string url)

[parse_url](#) retourne un tableau associatif contenant les composants de l'URL. Si l'un des composants n'est pas présent dans l'URL, l'entrée correspondante dans le tableau ne sera pas créée. Les composants sont :

- scheme – e.g. http
- host
- port
- user
- pass
- path
- query – après le point d'interrogation ?
- fragment – après le signe dièse #

Cette fonction **n'est pas** destinée à valider des URL, mais à les scinder en parties. Les URL partielles sont aussi acceptées, [parse_url](#) essaie alors de l'analyser correctement.

Note

[parse_url](#) ne fonctionne que avec les URL absolues.

Exemple avec [parse_url](#)

```

$ php -r 'print_r( parse_url("http://username:password@hostname/path?arg=value#anchor"));'
Array
(
    [scheme] => http
    [host] => hostname
    [user] => username
    [pass] => password
    [path] => /path
    [query] => arg=value
    [fragment] => anchor
)

$ php -r 'print_r( parse_url("http://invalid_host..name/"));'
Array
(
    [scheme] => http
    [host] => invalid_host..name
    [path] => /
)

```


Voir aussi [pathinfo](#) , [parse_str](#) , [dirname](#) et [basename](#) .

7.105.12 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.

base64_decode example

```
<?php
$str = 'foo%20bar%40baz';
echo rawurldecode($str);
?>
```

This example will produce :

foo bar@baz

Voir aussi [rawurlencode](#) , [urldecode](#) et [urlencode](#) .

7.105.13 rawurlencode() : Encode une chaîne en URL, selon la RFC1738

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 RFC1738 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.my.com/x.txt">';
```

Ou, si vous transmettez un chemin dans une URL

Exemple avec [rawurlencode](#)

```
echo '<a href="http://x.com/department_list_script/',
rawurlencode('ventes et marketing/Bruxelles'), '>';
```

Voir aussi [rawurldecode](#) , [urldecode](#) et [urlencode](#) , et la [RFC 2045](#) section 6.8 .

7.105.14 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 = split('&', $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>";
    $i++;
}
?
```

Voir aussi [urlencode](#) , [rawurlencode](#) et [rawurldecode](#) .

7.105.15 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 RFC1738 (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 par exemple `&copy`, `&copy` et `&pound`, 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.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 avec [htmlentities](#) .

Exemple avec [urlencode](#) et [htmlentities](#)

```
<?php
echo '<A HREF="moncgi?foo=', htmlentities (urlencode ($userinput) ), '>';
?>
```

Voir aussi [urldecode](#) , [htmlentities](#) , [rawurldecode](#) et [rawurlencode](#) .

7.106 Variables

7.106.1 Introduction

Pour plus de détails sur le comportement des variables, reportez vous à la section [Variables](#) du chapitre [Référence du langage](#) .

7.106.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.106.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.106.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<code>unserialize_callback_func</code>	<code>""</code>	<code>PHP_INI_ALL</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

```
unserialize_callback_func string
```

La fonction de callback d'unserialize sera appelée (avec le nom de la classe non définie en paramètre), si le délinéarisateur 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](#) .

7.106.5 Types de ressources

Cette extension ne définit aucune ressource.

7.106.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)

- [Constantes prédefinies](#)
- [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](#)

7.106.8 empty() : Détermine si une variable contient une valeur non nulle

bool **empty** (mixed var)

Note

`empty` est une structure de langage.

`empty` retourne la valeur `FALSE` si la variable `var` est affectée ou bien a une valeur différente de 0; la valeur `TRUE` dans les autres cas.

Exemple avec `empty`

```
<?php
$var = 0;
if (empty($var)) { // retourne
<TT>TRUE</TT>

    print 'soit $var vaut 0, soit il n\'est pas défini';
}
if (!isset($var)) { // retourne
<TT>FALSE</TT>

    print '$var n\'est pas définie';
}
?>
```

Notez que cette fonction n'a pas de sens si elle est utilisée sur autre chose qu'une variable. i.e. ^{empty} (addslashes (\$name)) n'a pas de sens, car cela revient à vérifier une entité qui n'est pas une variable.

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 isset et unset .

7.106.9 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} .

^{var} peut être n'importe quel type scalaire. Vous ne pouvez pas utiliser floatval sur un tableau ou un objet.

Exemple avec floatval

```
<?php
$var = '122.34343Le';
$float_value_of_var = floatval ($var);
print $float_value_of_var; // Affiche 122.34343
?>
```

Voir aussi intval , strval , settype et Transtypage .

7.106.10 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'environnement, de serveurs ou définies par l'utilisateur.

Exemple avec get_defined_vars

```
<?php
$b = array(1,1,2,3,5,8);

$arr = get_defined_vars();

// affiche $b
print_r($arr["b"]);

// affiche le chemin jusqu'à l'interpréteur
// CGI PHP (si PHP est utilisé en CGI)
// i.e. /usr/local/bin/php
echo $arr["_"];

// affiche la ligne de commande, s'il y en a une
print_r($arr["argv"]);

// affiche toutes les variables serveurs
print_r($arr["HTTP_SERVER_VARS"]);

// affiche toutes les clés disponibles dans les tableaux de variables
print_r(array_keys(get_defined_vars()));
?>
```

Voir aussi get_defined_functions et get_defined_constants .

7.106.11 `get_resource_type()` : Retourne le type de ressource

string **get_resource_type** (resource handle)

`get_resource_type` retourne une chaîne représentant le type de ressources de `handle`. Si le paramètre n'est pas une ressource valide, une erreur est générée.

Exemple avec `get_resource_type`

```
<?php
$c = mysql_connect();
echo get_resource_type($c)."\n";
// affiche : mysql link
// (lien mysql)
$f = fopen("foo", "w");
echo get_resource_type($f)."\n";
// affiche : file
// (fichier)
$doc = new_xmldoc("1.0");
echo get_resource_type($doc->doc)."\n";
// affiche : domxml document
// (document domxml)
?>
```

7.106.12 `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)
- "integer "
- "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 integer`, `is null`, `is numeric`, `is object`, `is resource`, `is scalar` et `is string`.

7.106.13 `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 sensible la casse, et vous pouvez les combiner entre eux. POST inclus les fichiers uploadés. 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 autre lettre, autre que GPC 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 par exemple `$_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 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_");

print $rvar_foo;
?>
```

Voir aussi `$_REQUEST` , `register_globals` , `variables prédéfinies` et `extract` .

7.106.14 `intval()` : Retourne la valeur numérique entière équivalent d'une variable

```
int intval ( mixed var , int base )
```

`intval` retourne la valeur numérique entière (integer) de la variable `var` , en convertissant la valeur dans la base spécifiée (par défaut en base 10).

`var` peut être de type scalaire. Vous ne pouvez pas utiliser la fonction `intval` avec un tableau ou un objet.

Voir aussi `doubleval` , `strval` , `settype` et `Transtypage` .

7.106.15 `is_array()` : Détermine si une variable est un tableau

`bool is_array ( mixed var )`

`is_array` renvoie la valeur `TRUE` si la variable `var` est un `array`, `FALSE` sinon.

Voir aussi [is_float](#), [is_int](#), [is_integer](#), [is_string](#) et [is_object](#).

7.106.16 `is_bool()` : Détermine si une variable est un booléen

`bool is_bool ( mixed var )`

`is_bool` retourne `TRUE` si `var` est un `boolean`.

Exemple avec `is_bool`

```
<?php
$a = false;
$b = 0;

// Comme $a est un booléen, ceci est vrai
if (is_bool($a)) {
    print "Oui, c'est un booléen";
}

// Comme $b n'est pas un booléen, ceci est faux
if (is_bool($b)) {
    print "Oui, c'est un booléen";
}
?>
```

Voir aussi [is_array](#), [is_float](#), [is_int](#), [is_integer](#), [is_string](#) et [is_object](#).

7.106.17 `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 stocké 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 call back. 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 ("callable name"). Dans l'exemple ci-dessous, il vaut `"uneClasse:uneMethode"`. Notez que, bien que `uneClasse:uneMethode()` puisse être appelé sous forme static, 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

?>
```

7.106.18 `is_double()` : Alias de `is_float`

`is_double` est un alias de `is_float` .

7.106.19 `is_float()` : Détermine si une variable est de type float

bool `is_float` (mixed var)

Cette fonction est un alias de la fonction `is_double` .

Voir aussi `is_double` , `is_bool` , `is_real` , `is_int` , `is_integer` , `is_string` , `is_object` , `is_array` , et `is_long` .

7.106.20 `is_int()` : Détermine si une variable est de type integer

bool `is_int` (mixed var)

`is_int` est un alias de la fonction `is_long` .

Voir aussi `is_bool` , `is_double` , `is_float` , `is_integer` , `is_string` , `is_real` , `is_object` , `is_array` , et `is_long` .

7.106.21 `is_integer()` : Détermine si une variable est de type int

bool `is_integer` (mixed var)

Cette fonction est un alias de la fonction `is_long` .

Voir aussi [is_bool](#) , [is_double](#) , [is_float](#) , [is_int](#) , [is_string](#) , [is_real](#) , [is_object](#) , [is_array](#) , et [is_long](#) .

7.106.22 is_long() : Alias de [is_int](#)

[is_long](#) est un [alias](#) de [is_int](#) .

7.106.23 is_null() : Indique si une variable vaut ^{NULL}

bool **is_null** (mixed var)

[is_null](#) retourne ^{TRUE} , si var est ^{NULL} , et ^{FALSE} .

Reportez-vous au type ^{NULL} pour savoir quand une variable est considérée comme contenant la valeur ^{NULL} ou non.

Voir aussi [NULL](#) , [is_bool](#) , [is_numeric](#) , [is_float](#) , [is_int](#) , [is_string](#) , [is_object](#) , [is_array](#) , [is_integer](#) et [is_real](#)

7.106.24 is_numeric() : Détermine si une variable est un type numérique

bool **is_numeric** (mixed var)

[is_numeric](#) retourne ^{TRUE} si var est un nombre, ou une chaîne numérique, ou ^{FALSE} sinon.

Voir aussi [is_bool](#) , [is_float](#) , [is_int](#) , [is_string](#) , [is_object](#) , [is_array](#) et [is_integer](#) .

7.106.25 is_object() : Détermine si une variable est de type object

bool **is_object** (mixed var)

[is_object](#) renvoie ^{TRUE} si la variable var est un objet, ^{FALSE} sinon.

Voir aussi [is_bool](#) , [is_int](#) , [is_integer](#) , [is_float](#) , [is_string](#) et [is_array](#) .

7.106.26 is_real() : Détermine si une variable est de type real

bool **is_real** (mixed var)

Cette fonction est un alias de la fonction [is_double](#) .

Voir aussi [is_bool](#) , [is_long](#) , [is_int](#) , [is_integer](#) , [is_float](#) , [is_double](#) , [is_object](#) , [is_string](#) , et [is_array](#) .

7.106.27 is_resource() : Détermine si une variable est une ressource

bool **is_resource** (mixed var)

[is_resource](#) retourne ^{TRUE} si la variable var est une ressource PHP, sinon ^{FALSE} .

Les ressources peuvent être des pointeurs de fichiers, des identifiants de résultats SQL, qui sont allouées et libérées en interne, par PHP, et qui peuvent demander un peu de nettoyage lorsqu'elles sont devenues inutiles, mais pas encore supprimées.

7.106.28 `is_scalar()` : Indique si une variable est un scalaire

`bool is_scalar ( mixed var )`

`is_scalar` retourne `TRUE` si la variable `var` est scalaire, et `FALSE` sinon.

Les variables scalaires sont celles qui contiennent des entiers, des nombres à virgule flottante, des chaînes de caractères ou des booléens. Par exemple :

Exemple avec `is_scalar`

```
<?php
function show_var($var) {
    if (is_scalar($var))
        echo $var;
    else
        var_dump($var);
}
$pi = 3.1416;
$proteines = array("hemoglobine", "cytochrome c oxidase", "ferredoxine");
show_var($pi);
// affiche : 3.1416
show_var($proteines)
// affiche:
// array(3) {
//   [0]=>
//     string(10) "hemoglobine"
//   [1]=>
//     string(20) "cytochrome c oxidase"
//   [2]=>
//     string(10) "ferredoxine"
// }
?>
```

Voir aussi `is_bool` , `is_numeric` , `is_float` , `is_int` , `is_real` , `is_string` , `is_object` , `is_array` et `is_integer` .

7.106.29 `is_string()` : Détermine si une variable est de type chaîne

`bool is_string ( mixed var )`

`is_string` renvoie `TRUE` si la variable `var` est du type chaîne de caractères (`string`), `FALSE` sinon.

Voir aussi `is_bool` , `is_int` , `is_integer` , `is_float` , `is_real` , `is_object` et `is_array` .

7.106.30 `isset()` : Détermine si une variable est affectée

`bool isset ( mixed var , mixed var , ... )`

Note

`isset` est une structure du langage.

`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` .

Exemple avec `isset`

```
<?php
$var = '';

// Ceci est vrai, alors le texte est affiché
if (isset($var)) {
    print "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` , `array_key_exists` et l'opérateur de contrôle de rapport d'erreur `@` .

7.106.31 `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 sera 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.

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>

```

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.

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 `array` ou un `object` 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`.

7.106.32 serialize() : Linéarise une variable

string **serialize** (mixed value)

`serialize` retourne une chaîne contenant une représentation linéaire de `value`, 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 les types `integer`, `double`, `string`, `array` (multidimensionnels) et `object` (les propriétés des objets seront linéarisées, mais pas les méthodes).

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)) {
        /* Grosse bourde! Souffre et potasse! */
    }
}
?>
```

7.106.33 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 .

7.106.34 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.

`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 .

7.106.35 unserialize() : Crée une variable PHP à partir d'une valeur linéarisée

mixed **unserialize** (string *str* , string *callback*)

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 `integer` , `double` , `string` , `array` ou `object` . Les objets linéarisés perdent leurs méthodes.

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 instantiée, 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.

Note

Le paramètre `callback` est disponible 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 : moncallbackfunction

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 bojet. PHP 4 lève cette limitation, 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 sauveées sont dasn $tmp[0].
    $session_data = unserialize ($tmp[0]);
    if (!is_array ($session_data)) {
        // Erreur... initialisation à tableau vide
        $session_data = array();
    }
}
?>
```

Voir aussi [serialize](#) .

7.106.36 unset() : Détruit une variable

void **unset** (mixed var , mixed var , ...)

[unset](#) détruit les variables `var` . Notez qu'en PHP 3, [unset](#) retournait toujours `TRUE` (en fait, la valeur entière 1). [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 affichera :

bar

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 dernière 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'intérieur d'une fonction [unset](#) détruira la référence à la variable statique, plutôt que la variable statique elle-même.

Exemple avec [unset](#) avec une variable statique

```
<?php
function foo() {
    static $a;
    $a++;
    echo "$a\n";
    unset($a);
}
foo();
foo();
foo();
?>
```

L'affichage du script ci-dessus donnera :

```
1
2
3
```

Si vous voulez détruire une variable globale, depuis une fonction, vous pouvez utiliser le tableau `$globals` .

Exemple avec [unset](#) avec une variable globale

```
<?php
function foo() {
    unset($GLOBALS['bar']);
}
$bar = "truc";
foo();
?>
```

Note

[unset](#) est une structure du langage et non pas une fonction.

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](#) .

7.106.37 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 sont explorés récursivement, avec des indentations, pour mettre en valeur leur structure.

Array

Comparez [var_dump](#) et [print_r](#) .

Exemple avec [var_dump](#)

```
<PRE>
<?php
```

```

$a = array (1, 2, array ("a", "b", "c"));
var_dump ($a);

/* affichage :
array(3) {
  [0]=>
  int(1)
  [1]=>
  int(2)
  [2]=>
  array(3) {
    [0]=>
    string(1) "a"
    [1]=>
    string(1) "b"
    [2]=>
    string(1) "c"
  }
}
*/

$b = 3.1;
$c = TRUE;
var_dump($b,$c);

/* affichage :
float(3.1)
bool(true)
*/
?>
</PRE>

```

7.106.38 var_export() : Retourne le code PHP utilisé pour générer une variable

`mixed var_export (mixed expression, bool return)`

`var_export` retourne des données structurés sur la variable `expression`. C'est le même principe que `var_dump` mais le résultat retourné est du code PHP.

Par défaut, le code sera affiché, mais vous pouvez retourner ce code dans une variable en passant `TRUE` pour le paramètre de `return`.

Comparez `var_export` avec `var_dump`.

Exemple avec var_export

```

<pre>
<?php
$a = array (1, 2, array ("a", "b", "c"));
var_export ($a);

/* affiche :
array (
  0 => 1,
  1 => 2,
  2 =>
    array (
      0 => 'a',
      1 => 'b',
      2 => 'c',
    ),
)
*/

$b = 3.1;
$v = var_export($b, TRUE);
echo $v;

/* affiche :
3.1
*/
?>
</pre>

```

7.107 vpopmail

7.107.1 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.

7.107.2 Installation

En ^{PHP} 4, ces fonctions ne sont disponibles que si ^{PHP} a été configuré avec `--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](#)

7.107.4 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.

7.107.5 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.

7.107.6 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.

7.107.7 vpopmail_add_user() : Ajoute un nouvel utilisateur à un domaine virtuel

```
bool vpopmail_add_user ( string user , string domain , string password , string gecost , 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.

7.107.8 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.

7.107.9 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.

7.107.10 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.

7.107.11 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.

7.107.12 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.

7.107.13 vpopmail_auth_user() : Valide le mot de passe d'un utilisateur pour un domaine. Retourne ^{TRUE} / ^{FALSE}

```
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.

7.107.14 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.

7.107.15 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.

7.107.16 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.

7.107.17 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.

7.107.18 vpopmail_passwd() : Remplace le mot de passe d'un utilisateur virtuel

```
bool vpopmail_passwd ( string user , string domain , string password )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.107.19 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.

7.108 API windows

7.108.1 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_defctype](#).

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.

7.108.2 Pré-requis

Cette extension ne fonctionne qu'avec le système Windows.

7.108.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

7.108.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.108.5 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".

7.108.6 Constantes prédefinies

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*)

7.108.7 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

<?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
$sticks = GetTickCount();

// Conversion en texte lisible
$secs = floor($sticks / 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.",
              $sticks,
              $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);
?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [Exemples](#)
- [w32api_deftype](#)
- [w32api_init_dtype](#)
- [w32api_invoke_function](#)
- [w32api_register_function](#)
- [w32api_set_call_method](#)

7.108.9 w32api_init_dtype() : Crée une instance de type de données 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 `val1` , `val2` ,... 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` .

7.108.10 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 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` .

7.108.11 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 librairie `library` , et essaie de l'importer en `PHP` . La fonction sera alors enregistrée avec un type de valeur de retour donné. Ce 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.

7.108.12 w32api_set_call_method() : Modifie le type d'appel de la fonction

`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` .

7.109 WDDX

7.109.1 Introduction

Ces fonctions fonctionnent grâce à WDDX .

7.109.2 Pré-requis

Pour utiliser WDDX, vous devez installer la librairie EXPAT (qui est fournie avec la distribution d'Apache 1.3.7 ou plus récent).

7.109.3 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 librairie supplémentaire pour disposer de ces fonctions.

7.109.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.109.5 Types de ressources

Cette extension ne définit aucune ressource.

7.109.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.109.7 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.

Enregistrer une valeur simple

```
<?php
print 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 paquet de PHP à WDDX</string></data></wddxPacket>
```

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);
print $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'
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [wddx_add_vars](#)
- [wddx_deserialize](#)
- [wddx_packet_end](#)
- [wddx_packet_start](#)
- [wddx_serialize_value](#)
- [wddx_serialize_vars](#)

7.109.9 wddx_deserialize() : Lire un paquet WDDX

wddx_deserialize (*string* *packet*)

[wddx_deserialize](#) prend la chaîne *packet* et la lit. Cette fonction retourne un résultat qui peut être une chaîne, un nombre ou un tableau. Notez que les structures sont lues sous la forme de tableaux associatifs.

7.109.10 wddx_packet_end() : Clos un paquet WDDX

wddx_packet_end (*int* *packet_id*)

[wddx_packet_end](#) clos un paquet WDDX repéré par son identifiant *packet_id* .

7.109.11 wddx_packet_start() : Commencer un nouveau paquet WDDX avec une structure

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.

7.109.12 wddx_serialize_value() : Enregistrer 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.

7.109.13 wddx_serialize_vars() : Enregistrer 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.

```
wddx_serialize_vars

<?php
$a = 1;
$b = 5.5;
$c = array("bleu", "orange", "violet");
$d = "colors";
$clvars = array("c", "d");
print wddx_serialize_vars("a", "b", $clvars);
?>
```

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>
```

7.110 Analyseur syntaxique XML

7.110.1 Introduction

Le langage XML (eXtensible Markup Language (Langage à Balises Etendu)) 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 supporte la librairie `expat` de James Clark sous PHP. Cela vous permettra d'analyser mais pas de valider les documents XML. Il supporte trois types de codage 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 **points d'entrée** pour chaque événement XML. Les analyseurs XML disposent de quelques paramétrages .

7.110.2 Pré-requis

Cette extension de PHP utilise `expat` , disponible à <http://www.jclark.com/xml/expat.html> . Le fichier Makefile livré avec `expat` ne construit pas par défaut de librairie : il faut utiliser la ligne suivante :

```
libexpat.a: $(OBJJS)
ar -rc $@ $(OBJJS)
ranlib $@
```

Les sources RPM de `expat` sont disponibles à <http://sourceforge.net/projects/expat/> .

7.110.3 Installation

Ces fonctions sont activées par défaut, et utilisent la librairie `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 Apache et avec les versions 1.3.9 ou plus récent, PHP va automatiquement utiliser la librairie fournie par Apache. Si vous ne souhaitez pas utiliser la librairie `expat` fournie avec PHP, il faut que vous compiliez PHP avec l'option `--with-expat-dir=DIR` , où DIR est le dossier d'installation de votre librairie `expat`.

La version Windows de `PHP` dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

7.110.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.110.5 Types de ressources

7.110.5.1 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 l'extension.

7.110.6 Constantes prédefinies

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)
```

7.110.7 gestionnaires d'évènements

Les gestionnaires d'évènements XML sont:

Fonction PHP de configuration du gestionnaire	Description de l'évènement
<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.
<code>xml_set_processing_instruction_handler</code>	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 spécifiques, (sauf le programme cible, qui est réservé à XML).
<code>xml_set_default_handler</code>	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.
<code>xml_set_unparsed_entity_decl_handler</code>	Ce gestionnaire est appelé pour gérer les déclaration des entités non analysés.
<code>xml_set_notation_decl_handler</code>	Ce gestionnaire est appelé pour gérer les notations.

<code>xml_set_external_entity_ref_handler</code>	Ce gestionnaire est appelé lorsque l'analyseur XML trouve une référence à un fichier externe. Cela peut être un fichier, ou une URL. Reportez-vous à <u>entité externe</u> pour un exemple.

7.110.8 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 caractère 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`, respectivement.

7.110.9 Codes d'erreurs

Les constantes suivantes sont définies comme des codes d'erreurs XML : (retournée 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

7.110.10 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 doit 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 derniers 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).

7.110.11 Exemples

Voici une liste d'exemple de code PHP qui analyse un document XML.

7.110.11.1 Exemple de structure XML

Ce premier exemple affiche la structure de l'élément de début dans un document avec indentation.

Afficher une structure XML

```
<?php
$file = "donnees.xml";
$depth = array();

function debutElement($parser, $name, $attrs) {
    global $depth;
    for ($i = 0; $i < $depth[$parser]; $i++) {
        print "  ";
    }
    print "$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);
?>
```

7.110.11.2 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 $map_array;
    if ($htmltag == $map_array[$name]) {
        print "<$htmltag>";
    }
}

function endElement($parser, $name) {
```

```

    global $map_array;
    if ($htmltag == $map_array[$name]) {
        print "</$htmltag>";
    }
}

function characterData($parser, $data) {
    print $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);
?>

```

7.110.11.3 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 (!ereg("^[a-z]+://", $file)
        && fileowner($file) == getmyuid()) {
        return true;
    }
    return false;
}

function startElement($parser, $name, $attrs) {
    print "<<font color=\"#0000cc\">$name</font>";
    if (sizeof($attrs)) {
        while (list($k, $v) = each($attrs)) {
            print " <font color=\"#009900\">$k</font>=<font
                color=\"#990000\">$v</font>>";
        }
    }
    print ">";
}

function endElement($parser, $name) {
    print "</<font color=\"#0000cc\">$name</font>>";
}

function characterData($parser, $data) {
    print "<b>$data</b>";
}

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: <i>%s</i>",
                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");
}

print "<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)));
    }
}
print "</pre>";
print "parse complete\n";
xml_parser_free($xml_parser);
?>

```

xmltest.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>
    </tbody>
  </tgroup>
</informaltable>
</para>
</chapter>

```

```
<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

<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 prédéfinies](#)
- [gestionnaires d'évènements](#)
- [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](#)

7.110.13 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 caractères de grande taille dans des flôts 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 flôt 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.

7.110.14 xml_error_string() : Lit le message d'erreur de l'analyseur XML

string **xml_error_string** (int code)

code

Un message d'erreur, issu de xml_get_error_code.

xml_error_string retourne la chaîne avec un message textuel, décrivant l'erreur ^{code}, ou ^{FALSE} si aucune description n'a été trouvée.

7.110.15 xml_get_current_byte_index() : Retourne l'index de l'octet courant d'un analyseur XML

int **xml_get_current_byte_index** (resource parser)

parser

Une référence sur un analyseur XML valide.

xml_get_current_byte_index retourne ^{FALSE} si ^{parser} n'est pas valide, ou sinon, retourne l'index de l'octet d'analyse courante de l'analyseur XML.

7.110.16 xml_get_current_column_number() : Retourne le nombre courant de colonne d'un analyseur XML.

int **xml_get_current_column_number** (resource parser)

parser

Une référence sur un analyseur XML valide.

xml_get_current_column_number retourne ^{FALSE} si ^{parser} n'est pas valide, ou sinon, retourne le numéro de colonne courante de la ligne courante de l'analyseur, qui correspond à la position d'analyse courante de l'analyseur XML.

7.110.17 `xml_get_current_line_number()` : Retourne le numéro de ligne courant d'un analyseur XML.

```
int xml_get_current_line_number ( resource parser )
```

parser

Une référence sur un analyseur XML valide.

`xml_get_current_line_number` retourne `FALSE` si `parser` n'est pas valide, ou sinon, retourne le numéro de la ligne en cours d'analyse.

7.110.18 `xml_get_error_code()` : Retourne le nombre courant de colonne d'un analyseur XML

```
int xml_get_error_code ( resource parser )
```

`xml_get_error_code` retourne `FALSE` si `parser` n'est pas valide, ou sinon, retourne le numéro de colonne courante de la ligne courante de l'analyseur, qui correspond à la position d'analyse courante de l'analyseur XML.

7.110.19 `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` array. Ces deux paramètres sont passés par références.

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);
?>
```

Lors de l'exécution du code, l'affichage sera :

```
Tableau d'index
Array
(
    [PARA] => Array
        (
            [0] => 0
            [1] => 2
        )

    [NOTE] => Array
        (
            [0] => 1
        )
)
```

7.110.17 `xml_get_current_line_number()` : Retourne le numéro de ligne courant d'un analyseur XML.

```

)

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 moléculaire

```

<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) {

```

7.110.17 [xml_get_current_line_number\(\)](#) : Retourne le numéro de ligne courant d'un analyseur XML.


```

        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
    )
)

```

7.110.20 xml_parse() : Commence l'analyse d'un fichier XML

```
bool xml_parse ( resource parser , string data , bool is_final )
```

`parser`

une référence sur l'analyseur XML à utiliser.

`data`

Une partie des données à analyser. Un document peut être analysé morceau par morceau, en appelant `xml_parse` plusieurs fois, tant que le paramètre `is_final` est mis à `TRUE` pour le dernier morceau.

`is_final` (optional)

S'il vaut `TRUE`, `data` est la dernière partie à analyser.

Lorsqu'un document XML est analysé, les gestionnaires d'événements sont appelés aussi souvent que nécessaire, et retournent `TRUE` ou `FALSE`.

`TRUE` est retourné lorsque l'analyse a été concluante, et `FALSE` en cas d'échec, ou si `parser` n'est pas un analyseur valide. Lors d'un échec d'analyse, la cause de l'erreur peut être obtenue grâce aux fonctions `xml_get_error_code`, `xml_error_string`, `xml_get_current_line_number`, `xml_get_current_column_number` et `xml_get_current_byte_index`.

7.110.21 `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 que le XML doit traiter. Les jeux de caractères supportés sont `"ISO-8859-1"`, qui est aussi la valeur par défaut si `encoding` est omis, `"UTF-8"` et `"US-ASCII"`.

Voir aussi `xml_parser_create_ns` et `xml_parser_free`.

7.110.22 `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. `xml_parser_create` retourne `FALSE` en cas d'erreur.

`encoding` (optionnel)

Le codage de caractère de l'analyseur. Les codages suivants sont supportés :

`ISO-8859-1` (par défaut)

`US-ASCII`

`UTF-8`

Voir aussi `xml_parser_create_ns` et `xml_parser_free`.

7.110.23 `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`.

7.110.24 `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. Reportez-vous à `xml_parser_set_option` pour avoir la liste des options disponibles.

`xml_parser_get_option` retourne `FALSE` si `parser` n'est pas valide, ou sinon, retourne la valeur de l'option demandée.

Reportez-vous à `xml_parser_set_option` pour avoir la liste des options disponibles.

7.110.25 `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 casse des balises de cet analyseur XML. Par défaut, activé.
<code>XML_OPTION_TARGET_ENCODING</code>	string	Modifie le codage à la cible 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.

7.110.26 `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 en train d'être 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.

7.110.27 `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'autre données pour qui aucun gestionnaire n'est prévu.

Si un gestionnaire reçoit une chaîne vide ou `FALSE`, c'est qu'il est en train d'être désactivé.

`xml_set_default_handler` retourne `TRUE` si le gestionnaire est actif, et `FALSE` sinon, ou si `parser` n'est pas un analyseur.

Il n'est pas pour l'instant possible d'utiliser des objets pour servir de gestionnaire. Reportez-vous à `xml_set_object` pour utiliser l'analyseur XML depuis un objet.

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.

7.110.28 `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`.

`startelementhandler` et `endelementhandler` sont des chaînes qui contiennent les noms de fonctions qui existent lorsque `xml_parse` est appelé pour créer `parser`.

La fonction `startelementhandler` doit accepter trois paramètres: `start_element_handler resource parser ( string name , array attribs )`

`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.

`attribs`
Le troisième paramètre, `attribs`, 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 `attribs`, avec la fonction `each`. La première clé sera la première clé du tableau.

La fonction `end_element_handler` 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.

7.110.29 xml_set_end_namespace_decl_handler() : Configure le gestionnaire XML de données

`bool xml_set_end_namespace_decl_handler ( resource pind , 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.

7.110.30 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 `end_element_handler` 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` (comme 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.

`openentitynames`
Le deuxième paramètre, `openentitynames`, 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 de cet entité externe. Actuellement, ce paramètre est toujours une chaîne vide.

systemid

Identifiant système pour cet entité externe.

publicid

Le cinquième paramètre, `publicid`, 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.

7.110.31 `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`.

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.

`notationname`

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 de cette entité externe. Actuellement, ce paramètre est toujours une chaîne vide.

`systemid`

Identifiant système pour cet entité externe.

`publicid`

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.

7.110.32 `xml_set_object()` : Utilise un analyseur XML à l'intérieur d'un objet.

`void 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.

```
<?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>");
?>
```

7.110.33 `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

```
<?
<TT>target</TT>
<IT>data</IT>
```

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 en train d'être 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.

7.110.34 xml_set_start_namespace_decl_handler() : Configure le gestionnaire de caractères

`bool xml_set_start_namespace_decl_handler ( resource pind , callback hdl )`

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.

7.110.35 xml_set_unparsed_entity_decl_handler() : Affecte les gestionnaires d'entités non déclarés

`bool xml_set_unparsed_entity_decl_handler ( resource parser , callback handler )`

`xml_set_unparsed_entity_decl_handler` affecte les gestionnaires d'entité non déclaré 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 )`

<code>parser</code>	Le premier paramètre, <code>parser</code> , est une référence sur l'analyseur XML qui appelle cette fonction.
<code>entityname</code>	Le nom de l'entité qui va être définie
<code>base</code>	La meilleure base de résolution de l'identifiant système de cette entité externe. Actuellement, ce paramètre est toujours une chaîne vide.
<code>systemid</code>	Identifiant système pour cet entité externe.
<code>publicid</code>	Identifiant public pour cet entité externe.
<code>notationname</code>	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.

7.111 XML-RPC

7.111.1 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.

7.111.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

7.111.3 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.

7.111.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
<code>xmlrpc_errors</code>	<code>"0"</code>	<code>PHP_INI_SYSTEM</code>
<code>xmlrpc_error_number</code>	<code>"0"</code>	<code>PHP_INI_ALL</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini set](#) .

7.111.5 Types de ressources

Cette extension ne définit aucune ressource.

7.111.6 Constantes prédefinies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)

- [xmlrpc_decode_request](#)
- [xmlrpc_decode](#)
- [xmlrpc_encode_request](#)
- [xmlrpc_encode](#)
- [xmlrpc_get_type](#)
- [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](#)

7.111.8 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.

7.111.9 xmlrpc_encode_request() : Génère le code XML d'une requête de méthode

```
string xmlrpc_encode_request ( string method , mixed params )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

7.111.10 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.

7.111.11 xmlrpc_get_type() : Retourne le type XMLRPC d'une valeur PHP. Particulièrement pratique pour les types chaînes de type base64 et datetime

```
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.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.111.12 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.

7.111.13 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.

7.111.14 xmlrpc_server_call_method() : Analyse une requête XML et appelle les méthodes associées

```
mixed 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.

7.111.15 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.

7.111.16 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.

7.111.17 xmlrpc_server_register_introspection_callback() : Enregistre une fonction PHP pour générer la documentation

```
bool xmlrpc_server_register_introspection_callback ( resource server , string 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.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.111.18 xmlrpc_server_register_method() : Enregistre une fonction PHP avec une méthode

```
bool xmlrpc_server_register_method ( resource server , string method_name , string 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.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.111.19 xmlrpc_set_type() : Modifie le type XMLRPC, base64 ou datetime, pour une chaînes 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.

7.112 XSLT

7.112.1 Introduction

Cette extension PHP fournit 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.

7.112.2 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.

7.112.3 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 que vous avez 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.

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).

7.112.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.112.5 Types de ressources

Cette extension ne définit aucune ressource.

7.112.6 Constantes prédefinies

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_SILENT` (*entier*)

Ignore tous les rapports d'erreurs. C'est une option générique pour toutes les interfaces qui seront ajoutées plus tard.

`XSLT_SAB_PARSE_PUBLIC_ENTITIES` (*entier*)

Commande à ^{Sablotron} de faire l'analyse des entités publics. Par défaut, cette option est désactivée.

`XSLT_SAB_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_SAB_DISABLE_STRIPPING` (*entier*)

Arrête la suppression des espaces (pour les fichiers de données uniquement).

`XSLT_SAB_IGNORE_DOC_NOT_FOUND` (*entier*)

Considère les documents non résolus (la fonction `document()`) comme non-bloquant.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [xslt_create](#)
- [xslt_errno](#)
- [xslt_error](#)
- [xslt_free](#)
- [xslt_process](#)
- [xslt_set_base](#)
- [xslt_set_encoding](#)
- [xslt_set_error_handler](#)
- [xslt_set_log](#)
- [xslt_set_sax_handler](#)
- [xslt_set_sax_handlers](#)
- [xslt_set_scheme_handler](#)
- [xslt_set_scheme_handlers](#)

7.112.8 `xslt_errno()` : Retourne le numéro d'erreur ^{XSLT} courant

`int xslt_errno ( resource xh )`

`xslt_errno` le numéro courant d'erreur, pour l'analyseur ^{XSLT} `xh`.

7.112.9 xslt_error() : Retourne le message d'erreur courant

mixed **xslt_error** (resource xh)

xslt_errno retourne le message d'erreur courant, pour l'analyseur ^{xh}.

Gestion d'erreurs avec xslt_error et xslt_errno

```
<?php
$xh = xslt_create();
$result = xslt_process($xh, 'dog.xml', 'pets.xsl');
if (!$result) {
    die(sprintf("Impossible de traiter le document XSLT [%d]: %s",
        xslt_errno($xh), xslt_error($xh)));
}
print($result);
xslt_free($xh);
?>
```

7.112.10 xslt_free() : D truit un analyseur ^{XSLT}

void **xslt_free** (resource xh)

xslt_free d truit l'analyseur ^{XSLT} xh .

7.112.11 xslt_process() : Transforme des donn es ^{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 r aliser des transformations ^{XSLT} en utilisant n'importe quel type de source : des conteneurs. Cela se fait gr ce   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 d faut des fichiers qui contiennent le document   traiter. Le conteneur de r sultat est par d faut un fichier qui recevra le document trait . Si le conteneur de r sultat n'est pas fourni (l'argument vaut alors ^{NULL}), le r sultat sera retourn  par la fonction.

Attention

Cette fonction a vu ses arguments  tre modifi  depuis la version ^{PHP} 4.0.6. Ne placez pas le code ^{XML} ou XSL dans les deuxi me et troisi me arguments, car cela engendrera une erreur de segmentation (^{segmentation fault}), dans les versions de ^{Sablotron} 0.95 et plus r centes.

Les conteneurs peuvent aussi  tre modifi s via le tableau ^{\$arguments}.

Le type le plus simple de transformation r alisable avec xslt_process est la transformation d'un fichier ^{XML} avec un fichier ^{XSLT}, en pla ant le r sultat dans un troisi me fichier. Faire cela avec Sablotron est tr s 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($xh, 'sample.xml', 'sample.xsl', 'result.xml')) {
    print "Réussi. sample.xml a été transformé par sample.xsl en result.xml";
    print ", result.xml contient ceci : \n<br>\n";
    print "<pre>\n";
    readfile('result.xml');
    print "</pre>\n";
}
else {
    print "Désolé, sample.xml n'a pu être transformé par sample.xsl en ";
    print " result.xml. La raison est " . xslt_error($xh) . " et ";
    print " 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) {
    print "Réussi. sample.xml a été transformé par sample.xsl en result.xml";
    print ", result.xml contient ceci : \n<br>\n";
    print "<pre>\n";
    print $result;
    print "</pre>\n";
}
else {
    print "Désolé, sample.xml n'a pu être transformé par sample.xsl en ";
    print " result.xml. La raison est " . xslt_error($xh) . " et ";
    print " 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) {
    print "Réussi. sample.xml a été transformé par sample.xsl en result.xml";
    print ", result.xml contient ceci : \n<br>\n";
    print "<pre>\n";
}

```

```

    print $result;
    print "</pre>\n";
}
else {
    print "Désolé, sample.xml n'a pu être transformé par sample.xsl en ";
    print " result.xml. La raison est " . xslt_error($xh) . " et ";
    print " le code d'erreur est " . xslt_errno($xh);
}
xslt_free($xh);
?>

```

Finalement, 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}.

Note

Notez que **file://** est nécessaire au début des chemin si vous utilisez Windows.

7.112.12 `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.

Note

Notez bien que **file://** est nécessaire dans tous les chemins que vous utiliserez, si vous êtes sous Windows.

7.112.13 `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.

7.112.14 `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}.

7.112.15 `xslt_set_log()` : Configure le fichier d'historique pour les messages ^{XSLT}

void **xslt_set_log** (resource *xh* , mixed *log*)

xh
log Une ressource d'analyseur ^{XSLT} .

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 :

Utilisation de l'enregistrement de messages XSLT

```
<?php
$xh = xslt_create();
xslt_set_log($xh, true);
xslt_set_log($xh, getcwd() . 'mon_fichier.log');

$result = xslt_process($xh, 'chien.xml', 'animal.xsl');
print($result);

xslt_free($xh);
?>
```

Note

Notez bien que **file://** est nécessaire dans tous les chemins que vous utiliserez, si vous êtes sous Windows .

7.112.16 `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 `xslt_set_sax_handler`

```
array(
[document] =>
    array(
        start document handler,
        end document handler
    ),
[element] =>
    array(
        start element handler,
        end element handler
    ),
[namespace] =>
    array(
        start namespace handler,
        end namespace handler
    ),
)
```

```
[comment] => comment handler,
[pi] => processing instruction handler,
[character] => character data handler
)
```

7.112.17 `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 )
```

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

7.112.18 `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
)
```

7.112.19 `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.

7.113 YAZ

7.113.1 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 peristantes 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/>.

7.113.2 Installation

Compilez YAZ (ANSI/NISO Z39.50 support) et installez-le. Compilez ^{PHP} avec vos modules et ajoutez l'option `--with-yaz[=DIR]`. Les instructions sont :

Installation de YAZ

```
gunzip -c php-4.3.X.tar.gz|tar xf -
gunzip -c yaz-1.9.Y.tar.gz|tar xf -
cd yaz-1.9.Y
./configure --prefix=/usr
make
make install
cd ../php-4.3.X.
./configure --with-yaz=/usr/bin
make
make install
```

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`

Sous Windows, `php_yaz.dll` dépend de `yaz.dll`. Vous trouverez la librairie `yaz.dll` dans un sous dossier du dossier `dlls` de l'archive Zip Win32. Copiez `yaz.dll` dans un dossier de votre variable d'environnement `PATH` (`c:\winnt\system32` ou `c:\windows\system32`).

Attention

L'extension IMAP ne peut être utilisée simultanément avec l'extension recode ou l'extension YAZ. Ceci est dû au fait que ces deux librairies utilisent un symbole interne commun.

Note

Le problème ci-dessus est résolu en version 2.0 de YAZ.

7.113.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable
yaz.max_links	"100"	PHP_INI_ALL
yaz.log_file	""	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [ini_set](#).

7.113.4 Types de ressources

Cette extension ne définit aucune ressource.

7.113.5 Constantes prédefinies

Cette extension ne définit aucune constante.

7.113.6 Exemples

PHP /YAZ conserve les connexions aux serveurs. Un entier positif représente l'ID d'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.

Recherche parallèle utilisant YAZ

```
<?php
$num_hosts = count($host);
if (empty($term) || count($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="1"
    name="host[]" value="z3950.bell-labs.com/books">
        BELL Labs Library
    <br>
    RPN Query:
    <input type="text" size="30" name="term">
    <input type="submit" name="action" value="Search">
    ';
} else {
    echo 'Vous avez recherché '.htmlspecialchars($term).'\n';
    for ($i = 0; $i < $num_hosts; $i++) {
        $id[] = yaz_connect($host[$i]);
        yaz_syntax($id[$i], "sutrs");
        yaz_search($id[$i], "rpn", $term);
    }
    yaz_wait();
    for ($i = 0; $i < $num_hosts; $i++) {
        echo '<hr>'. $host[$i].':\n';
        $error = yaz_error($id[$i]);
        if (!empty($error)) {
            echo "Erreur: $error";
        } else {
            $hits = yaz_hits($id[$i]);
            echo "Nombre de résultats : $hits";
        }
        echo '\n';
        for ($p = 1; $p <= 10; $p++) {
            $rec = yaz_record($id[$i], $p, "string");
            if (empty($rec)) continue;
            echo "<dt><B>$p</B></dt><dd>";
            echo ereg_replace("\n", "<br>\n", $rec);
            echo "</dd>";
        }
        echo '</dl>';
    }
}
?>
```

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédefinies](#)
- [Exemples](#)
- [yaz_addinfo](#)
- [yaz_ccl_conf](#)
- [yaz_ccl_parse](#)
- [yaz_close](#)
- [yaz_connect](#)
- [yaz_database](#)
- [yaz_element](#)
- [yaz_errno](#)
- [yaz_error](#)
- [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](#)

7.113.8 `yaz_ccl_conf()` : Configure l'analyseur CCL

`int yaz_ccl_conf ( resource id , array config )`

`yaz_ccl_conf` configure l'analyseur CCL de requête de l'hôte `id`, avec les définitions de points d'accès (CCL qualifiers) et leur équivalent en RPN. Pour cabler une requête spécifique vers un appel RPN, utilisez `yaz_ccl_parse`. Chaque index du tableau `config` est un nom de champ CCL et la valeur correspondante contient une chaîne spécifiant le code RPN. Ce code est une séquence de paires "attribut-type, attribut-valeur". Les "attribut-type" et "attribut-valeur" sont séparés par le signe égal (=). Chaque paire est séparée par un espace (" ").

Exemple de configuration CCL

Dans l'exemple ci-dessous, l'analyseur CCL est configuré pour supporter trois champs CCL : `ti`, `au` et `isbn`. Chaque champ correspond à son équivalent BIB-1. On suppose que chaque variable `$id` est un hôte de destination.

```
<?php
$field["ti"] = "1=4";
$field["au"] = "1=1";
$field["isbn"] = "1=7";
yaz_ccl_conf($id,$field);
?>
```


7.113.9 `yaz_ccl_parse()` : Appelle l'analyseur CCL

`bool` **`yaz_ccl_parse`** (`resource` `id` , `string` `query` , `array` & `result`)

`yaz_ccl_parse` 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 celle-ci. Si la requête `query` a pu être convertie en RPN, `yaz_ccl_parse` retourne `TRUE`, et l'index `rpn` du tableau `result` contient une requête RPN valide. Si la requête n'a pas pu être convertie, (pour n'importe quelle raison, comme syntaxe invalide, champ inconnu...), `yaz_ccl_parse` retourne `FALSE`. Trois index sont alors créés dans le tableau de résultat : `errorcode` (code d'erreur CCL, un entier), `errorstring` (message d'erreur CCL), et `errorpos` position estimée de l'erreur dans la requête (entier, position en nombre de caractères).

Configuration CCL

Dans l'exemple ci-dessous, l'analyseur CCL est configuré pour supporter trois champs CCL : `ti`, `au` et `isbn`. Chaque champ correspond à son équivalent BIB-1. On suppose que la variable `$id` est la ressource de connexion.

```
<?php
$fields["ti"] = "1=4";
$fields["au"] = "1=1";
$fields["isbn"] = "1=7";
yaz_ccl_conf($id,$fields);
?>
```

7.113.10 `yaz_close()` : Ferme une connexion YAZ

`bool` **`yaz_close`** (`resource` `id`)

`yaz_close` ferme une connexion à un hôte YAZ. L'application ne pourra plus utiliser l'identifiant de connexion `id`. `id` est un identifiant d'hôte, retourné par `yaz_connect`.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

7.113.11 `yaz_connect()` : Prépare une connexion à un hôte YAZ

`resource` **`yaz_connect`** (`string` `zurl` , `mixed` `options`)

`yaz_connect` retourne un identifiant positif en cas de succès, et `FALSE` sinon.

`yaz_connect` prépare une connexion à un serveur Z39.50. `zurl` est de la forme "host[:port]/[database]". Si port est omis, 210 est utilisé. Si `database` est omis, `default` est utilisé. `yaz_connect` n'est pas bloquante, et ne tente pas d'établir une socket. En fait, elle ne fait que préparer la connexion pour exécution ultérieure par `yaz_wait`.

Si le second argument `options` est fourni sous la forme d'une chaîne, il est traité comme une chaîne d'authentification Z39.50 V2 (OpenAuth).

Si `options` est fourni sous la forme d'un tableau, le contenu du tableau servira d'options. Notez que ce tableau d'options est supporté depuis PHP 4.1.0.

Options `yaz_connect`

user

Nom d'utilisateur pour l'authentification.

group

Groupe d'authentification.

password

Mot de passe d'authentification.

cookie

Cookie de session (proxy YAZ).

proxy

Proxy de connexion (proxy YAZ).

*persistent*Un booléen . Si il vaut ^{TRUE} la connexion sera persistante; Si il vaut ^{FALSE} , la connexion ne le sera pas. Par défaut, les connexions sont persistantes.*piggyback*Un booléen . Si il vaut ^{TRUE} , piggyback est activé pour les recherches. Si il vaut ^{FALSE} , piggyback est désactivé. Par défaut, piggyback est activé. Activer piggyback est plus efficace, et généralement, cela économise des aller-retours sur le réseau pour les premières lectures de résultat. Mais peu de serveurs Z39.50 supportent piggyback, ou bien ils ignorent ce jeu de noms. Pour ceux là, il faut que piggyback soit désactivé.

Note

L'utilisation d'un proxy améliore souvent les performances. Un proxy Z39.50 fait partie du package libre YAZ++ .

7.113.12 `yaz_database()` : Spécifie la base d'une session YAZ

```
bool yaz_database ( resource id , string databases )
```

yaz_database spécifie ^{databases} , la ou les bases utilisées lors des recherches, lectures, etc, en remplaçant celles spécifiées lors de l'appel à yaz_connect . Pour indiquer plusieurs bases de données, séparez les noms par des ⁺ .yaz_database vous permet d'utiliser différents jeux de bases durant une session.Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.113.13 `yaz_element()` : Spécifie le type d'éléments YAZ à lire

```
bool yaz_element ( resource id , string elementset )
```

yaz_element est à utiliser en conjonction avec yaz_search et yaz_present pour spécifier le type d'éléments à lire. La majorité des serveurs supporte ^F (full, tous), et ^B (brief, bref).Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.113.14 `yaz_errno()` : Retourne le numéro d'erreur

```
int yaz_errno ( resource id )
```

yaz_errno retourne le numéro d'erreur de la dernière requête. Une valeur positive est retournée si le serveur a retourné un diagnostic. La valeur 0 est retournée si aucune erreur n'est survenue. Une

valeur négative indique une erreur sans diagnostic (connexion perdue,...).

yaz_errno doit être appelée après chaque requête. (après la fin de yaz_wait), pour savoir si la transaction a réussi ou échoué.

7.113.15 yaz_error() : Retourne une description de l'erreur

```
string yaz_error ( resource id )
```

yaz_error retourne un message d'erreur pour la dernière requête. Une chaîne vide est retournée si la dernière requête a réussi.

yaz_error retourne un message en anglais, qui correspond au numéro d'erreur retourné par yaz_errno .

7.113.16 yaz_get_option() : Retourne la valeur d'une option de connexion YAZ

```
string yaz_get_option ( resource id , string name )
```

yaz_get_option retourne la valeur de l'option ^{name} pour la connexion YAZ ^{name} . Si cette option n'est pas définie, une chaîne vide est retournée.

Voyez la description de yaz_set_option pour connaître les options disponibles.

7.113.17 yaz_hits() : Retourne le nombre de résultats de la dernière recherche

```
int yaz_hits ( resource id )
```

yaz_hits retourne le nombre de résultats de la dernière recherche.

7.113.18 yaz_itemorder() : Prépare une requête Z39.50 Item Order avec le package ILL-Request

```
int yaz_itemorder ( resource id , array args )
```

yaz_itemorder 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 . Le paramètre ^{args} doit être un tableau associatif, contenant les informations "Item Order" à envoyer. L'index du tableau est le nom ASN.1 correspondant au "tag path". Par exemple, le numéro ISBN sous l'Item-ID est la clé item-id,ISBN.

Les paramètres de ILL-Request sont :

```
protocol-version-num
transaction-id,initial-requester-id,person-or-institution-symbol,person
transaction-id,initial-requester-id,person-or-institution-symbol,institution
transaction-id,initial-requester-id,name-of-person-or-institution,name-of-person
transaction-id,initial-requester-id,name-of-person-or-institution,name-of-institution
transaction-id,transaction-group-qualifier
```

transaction-id, transaction-qualifier
 transaction-id, sub-transaction-qualifier
 service-date-time, this, date
 service-date-time, this, time
 service-date-time, original, date
 service-date-time, original, time
 requester-id, person-or-institution-symbol, person
 requester-id, person-or-institution-symbol, institution
 requester-id, name-of-person-or-institution, name-of-person
 requester-id, name-of-person-or-institution, name-of-institution
 responder-id, person-or-institution-symbol, person
 responder-id, person-or-institution-symbol, institution
 responder-id, name-of-person-or-institution, name-of-person
 responder-id, name-of-person-or-institution, name-of-institution
 transaction-type
 delivery-address, postal-address, name-of-person-or-institution, name-of-person
 delivery-address, postal-address, name-of-person-or-institution, name-of-institution
 delivery-address, postal-address, extended-postal-delivery-address
 delivery-address, postal-address, street-and-number
 delivery-address, postal-address, post-office-box
 delivery-address, postal-address, city
 delivery-address, postal-address, region
 delivery-address, postal-address, country
 delivery-address, postal-address, postal-code
 delivery-address, electronic-address, telecom-service-identifier
 delivery-address, electronic-address, telecom-service-address
 billing-address, postal-address, name-of-person-or-institution, name-of-person
 billing-address, postal-address, name-of-person-or-institution, name-of-institution
 billing-address, postal-address, extended-postal-delivery-address
 billing-address, postal-address, street-and-number
 billing-address, postal-address, post-office-box
 billing-address, postal-address, city
 billing-address, postal-address, region
 billing-address, postal-address, country
 billing-address, postal-address, postal-code
 billing-address, electronic-address, telecom-service-identifier
 billing-address, electronic-address, telecom-service-address
 ill-service-type
 requester-optional-messages, can-send-RECEIVED
 requester-optional-messages, can-send-RETURNED
 requester-optional-messages, requester-SHIPPED
 requester-optional-messages, requester-CHECKED-IN
 search-type, level-of-service
 search-type, need-before-date
 search-type, expiry-date
 search-type, expiry-flag
 place-on-hold
 client-id, client-name
 client-id, client-status
 client-id, client-identifier
 item-id, item-type
 item-id, call-number
 item-id, author
 item-id, title
 item-id, sub-title
 item-id, sponsoring-body
 item-id, place-of-publication
 item-id, publisher
 item-id, series-title-number
 item-id, volume-issue
 item-id, edition
 item-id, publication-date
 item-id, publication-date-of-component
 item-id, author-of-article
 item-id, title-of-article
 item-id, pagination
 item-id, ISBN

item-id, ISSN
 item-id, additional-no-letters
 item-id, verification-reference-source
 copyright-compliance
 retry-flag
 forward-flag
 requester-note
 forward-note

Il y a quelques paramètres du package Extended Services Request et ItemOrder :

package-name
 user-id
 contact-name
 contact-phone
 contact-email
 itemorder-item

7.113.19 `yaz_present()` : Prépare à la lecture (Z39.50 present).

```
int yaz_present ( resource id )
```

`yaz_present` prépare ^{PHP} à la lecture des résultats, après une recherche. `yaz_range` doit être appelée avant celle-ci pour spécifier la plage de résultats à lire.

7.113.20 `yaz_range()` : Spécifie le nombre maximal de résultat à lire

```
bool yaz_range ( resource id , int start , int number )
```

`yaz_range` est utilisée conjointement à `yaz_search`, pour spécifier le nombre maximal ^{number} de résultats à lire, ainsi que la position de début de lecture avec ^{start}. Si `yaz_range` n'est pas utilisée, ^{start} vaudra 1 et ^{number} vaudra 10.

Cette fonction retourne ^{TRUE} en cas de succès, ^{FALSE} en cas d'échec.

7.113.21 `yaz_record()` : Retourne un résultat

```
string yaz_record ( resource id , int pos , string type )
```

`yaz_record` retourne un résultat à la position ^{pos}, ou une chaîne vide si aucun résultat n'est disponible à la position ^{pos}.

`yaz_record` recherche une ligne dans le résultat, à la position spécifiée. Si aucune ligne n'existe à la position donnée, une chaîne vide est retournée. L'argument ^{type} spécifie la forme du résultat retourné : si ^{type} vaut "string", la ligne est retournée sous la forme d'une chaîne, prête à l'affichage. Si ^{type} vaut "array", la ligne sera retournée sous la forme d'un tableau structuré.

7.113.22 `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 ^{id}, lors de la dernière requête de scan. Le tableau commence à l'index 0. Chaque valeur est une paire, où le premier élément est le terme, et le second est un compte de résultat. Si ^{result} est fourni, il sera rempli avec les informations générales de la requête de scan : ^{number} (nombre d'entrée retournées), ^{stepsize} (taille du pas), ^{position} (position du terme), ^{status} (statut du Scan).

7.113.23 `yaz_scan()` : Prépare un scan

```
int yaz_scan ( resource id , string type , string startterm , array flags )
```

`yaz_scan` prépare une requête "Z39.50 Scan Request". ^{id} spécifie l'hôte cible. Le point de départ est donné avec ^{startterm}. La forme de spécification du point de départ est donné par ^{type}. Actuellement, le type ^{rpn} est supporté. Le paramètre optionnel ^{flags} donne des informations supplémentaires pour contrôler le comportement de la requête de scan. Actuellement, trois index sont lus dans ce paramètre : ^{number} (nombre de termes requis), ^{position} (position préférée du terme) et ^{stepSize} (taille du pas préférée). Pour réellement envoyer la requête de recherche à l'hôte, et recevoir la réponse, `yaz_wait` doit être appelée. A la fin de `yaz_wait`, `yaz_error` et `yaz_scan_result` auront les résultats.

La syntaxe de ^{startterm} est similaire aux requêtes RPN, décrites dans `yaz_search`. ^{startterm} est constitué de zéro ou plus spécifications, avec les opérateurs ^{@attr}, suivi par exactement un token.

Fonction PHP qui scanne les titres sur un serveur YAZ

```
<?php
function scan_titles($id, $startterm) {
    yaz_scan($id, "rpn", "@attr 1=4 " . $startterm);
    yaz_wait();
    $errno = yaz_errno($id);
    if ($errno == 0) {
        $ar = yaz_scan_result($id, &$options);
        echo 'Scan réussi: ';
        $ar = yaz_scan_result($id, &$options);
        while(list($key, $val)=each($options)) {
            echo "$key = $val &nbsp;";
        }
        echo '<br><table><tr><td>';
        while(list($key, list($k, $term, $tcount))=each($ar)) {
            if (empty($k)) continue;
            echo "<tr><td>$term</td><td>";
            echo $tcount;
            echo "</td></tr>";
        }
        echo '</table>';
    } else {
        echo "Echec du scan. Erreur: " . yaz_error($id) . "<br />";
    }
}
?>
```

7.113.24 `yaz_schema()` : Spécifie le schema de lecture

```
int yaz_schema ( resource id , string schema )
```

`yaz_schema` spécifie le schema de lecture ^{schema} pour la connexion ^{id}. Le schéma doit être spécifié sous la forme d'un OID (Object Identifier), au format points (comme par exemple ^{1.2.840.10003.13.4}) ou sous la forme d'un schéma enregistré : ^{GILS-schema}, ^{Holdings}, ^{Zthes}, etc. Cette fonction doit être appelée avant `yaz_search` et `yaz_present`.

7.113.25 `yaz_search()` : Prépare une recherche

```
int yaz_search ( resource id , string type , string query )
```

`yaz_search` prépare une recherche sur le serveur identifié par `id`. `type` représente le type de requête : seul RPN est supporté actuellement, et dans ce cas, le troisième argument est un préfixe de notation de requête utilisé par YAZ. Comme pour `yaz_connect`, `yaz_search` n'est pas bloquante, et ne fait que préparer la recherche pour exécution ultérieure, avec `yaz_wait`.

7.113.26 `yaz_set_option()` : Modifie une ou plusieurs options de connexion YAZ

```
string yaz_set_option ( resource id , string name , string value )
```

`yaz_set_option` modifie l'option `name`, en lui affectant la nouvelle valeur `value`, pour la connexion au serveur représentée par `id`.

Nom	Description
<code>implementationName</code>	nom de l'implémentation du serveur
<code>implementationVersion</code>	version de l'implémentation du serveur
<code>implementationId</code>	ID de l'implémentation du serveur
<code>schema</code>	schema de lecture. Par défaut, aucun schéma n'est utilisé. Modifier cette option est l'équivalent de l'utilisation de la fonction <code>yaz_schema</code>
<code>preferredRecordSyntax</code>	Syntaxe des enregistrements en lecture. Par défaut, il n'y a pas de syntaxe. Modifier cette option est l'équivalent de l'utilisation de la fonction <code>yaz_syntax</code>
<code>start</code>	Décalage du premier enregistrement à lire, via la fonction <code>yaz_search</code> ou <code>yaz_present</code> . Les enregistrements sont numérotés à partir de 0. Modifier cette option, en combinaison avec l'option <code>count</code> , est l'équivalent de l'utilisation de la fonction <code>yaz_range</code> , hormis le fait que les enregistrements sont numérotés à partir de 1 avec <code>yaz_range</code>
<code>count</code>	Nombre maximal d'enregistrements à lire via <code>yaz_search</code> et <code>yaz_present</code> .
<code>elementSetName</code>	element-set-name en lecture. Modifier cette option est l'équivalent de l'utilisation de la fonction <code>yaz_element</code> .

7.113.27 `yaz_sort()` : Configure les critères de tri

```
int yaz_sort ( resource id , string criteria )
```

`yaz_sort` permet de configurer les critères de tri, et d'activer le tri Z39.50. Utilisez cette fonction conjointement avec `yaz_search` ou `yaz_present`. Utiliser cette fonction seule n'a aucun effet. 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. Le paramètre `criteria` prend la forme suivante :

```
field1 flags1 field2 flags2 ...
```

où `field1` spécifie l'attribut primaire de tri, `field2` le second, etc... Le champ spécifie soit une combinaison d'attributs numériques, constituée de paire `type=valeur`, séparée par des virgules (e.g. `1=4,2=1`), soit le champ spécifie un critère en chaîne de caractères (e.g. `title`). Les options (flags) sont des séquences de caractères, choisis parmi les valeurs suivantes. Ces caractères ne peuvent être séparés par des espaces :

Options de tri

```
a          Tri ascendant
d          Tri descendant
i
```

Tri insensible à la casse

s

Tri sensible à la casse

Critères de tri

Pour effectuer des tris avec l'attribut Bib1 du champ title, de manière insensible à la casse, vous pouvez utiliser le critère suivant :

1=4 ia

Si le second critère de tri doit être l'auteur, en tenant compte de la casse, et dans l'ordre alphabétique, vous pouvez utiliser :

1=4 ia 1=1003 sa

7.113.28 yaz_syntax() : Spécifie la syntaxe de lecture des lignes

```
int yaz_syntax ( resource id , string syntax )
```

yaz_syntax est utilisée conjointement avec yaz_search pour spécifier la méthode de lecture des lignes. La syntaxe est spécifiée comme un OID (Identifiant d'Objet), en notation brute, séparée par des points (i.e. ^{1.2.840.10003.5.10}), ou bien avec une des valeurs prédéfinies : sutrs, usmarc, grs1, xml, etc... yaz_syntax doit être utilisée en conjonction avec yaz_search et yaz_present pour spécifier la méthode de lecture des résultats.

7.113.29 yaz_wait() : Attend l'exécution d'une requête

```
int yaz_wait ( array options )
```

yaz_wait exécute les requêtes préparée par les fonctions 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).

Si le paramètre ^{options} est fourni, il doit contenir les options suivantes, pour modifier le comportement de yaz_wait.

timeout

Configure le délais maximal d'attente, en secondes. Si le serveur n'a pas répondu durant ce délai, la connexion est considérée comme morte et yaz_wait se termine. Par défaut, ce délai est de 15 secondes.

7.114 NIS

7.114.1 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 n'est pas disponible sur les plate-formes Windows.

7.114.2 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).

7.114.3 Installation

Pour ajouter ces fonctionnalités, vous devez compiler PHP avec l'option `--enable-yp`.

7.114.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.114.5 Constantes prédefinies

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_YPSESV (entier)
YPERR_VERS (entier)

```

- [Introduction](#)

- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes prédefinies](#)
- [yp_all](#)
- [yp_cat](#)
- [yp_err_string](#)
- [yp_errno](#)
- [yp_first](#)
- [yp_get_default_domain](#)
- [yp_master](#)
- [yp_match](#)
- [yp_next](#)
- [yp_order](#)

7.114.7 yp_cat() : Retourne un tableau contenant toute la carte

array **yp_cat** (string domain , string map)

[yp_cat](#) retourne toutes les entrées de la carte sous la forme d'un tableau, dont les index sont les clés de la carte, et les entrées sont les valeurs du tableau.

7.114.8 yp_err_string() : Retourne la chaîne correspondante 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
<pre><?php echo "Erreur : " . yp_err_string(yp_errno()); ?></pre>

Voir aussi [yp_errno](#) .

7.114.9 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](#) .

7.114.10 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
$entree = yp_first($domain, "passwd.byname");
$cle = $entry ["key"];
$valeur = $entry ["value"];
echo "La première entrée de cette carte est " . $key
    . " et sa valeur est " . $entry[$key];
?>
```

Voir aussi [yp-get-default-domain](#) .

7.114.11 yp_get_default_domain() : Retourne le domaine NIS par défaut

int **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;
?>
```

7.114.12 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 "Master for this map is: " . $master;
?>
```

Voir aussi [yp-get-default-domain](#) .

7.114.13 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 de recherche NIS

```
<?php
$entry = yp_match($domain, "passwd.byname", "joe");
echo "La valeur trouvée est : " . $entry;
?>
```

Dans le cas présent, ce pourrait être: joe:##joe:11111:100:Joe User:/home/j/joe:/usr/local/bin/bash

Voir aussi [yp-get-default-domain](#) .

7.114.14 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";
}
$key = key($entry);
echo "L'entrée suivante après \"joe\" a la clé \" . $key
    . \" et la valeur \" . $entry[$key];
?>
```

Voir aussi [yp-get-default-domain](#) .

7.114.15 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 d'ordre NIS

```
<?php
$number = yp_order($domain,$mapname);
echo "Le numéro d'ordre de cette carte est : " . $number;
?>
```

Voir aussi [yp-get-default-domain](#) .

7.115 Accès aux fichiers Zip (Lecture seule)

7.115.1 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.

7.115.2 Pré-requis

Ce module utilise les fonctions de la librairie ZZIPLib , écrite par Guido Draheim. Vous aurez besoin de ZZIPLib version >= 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.

7.115.3 Installation

Le support ZIP de PHP n'est pas activé par défaut. Vous aurez besoin de l'activer avec l'option `--with-zip` dans le fichier de configuration de PHP.

Note
Le support Zip est expérimental avant PHP 4.1.0. Ce manuel est valable pour les versions 4.1.0 et plus récentes de PHP.

7.115.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

7.115.5 Types de ressources

Cette extension ne définit aucune ressource.

7.115.6 Constantes prédefinies

Cette extension ne définit aucune constante.

7.115.7 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
<pre><?php \$zip = zip_open("/tmp/test2.zip");</pre>

```

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 prédefinies](#)
- [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](#)

7.115.9 zip_entry_close() : Ferme un dossier d'archive

void **zip_entry_close** (resource zip_entry)

[zip_entry_close](#) ferme le dossier d'archive [zip_entry](#) . [zip_entry](#) doit être un dossier d'archive valide, fournit par [zip_entry_open](#) .

Cette fonction ne retourne aucune valeur.

Voir aussi [zip_entry_open](#) et [zip_entry_read](#) .

7.115.10 zip_entry_compressedsize() : Lit la taille compressée d'un dossier d'archive

int **zip_entry_compressedsize** (resource zip_entry)

[zip_entry_compressedsize](#) retourne la taille compressée du dossier d'archive [zip_entry](#). Le paramètre [zip_entry](#) est un dossier d'archive valide, retourné par [zip_read](#).

Voir aussi [zip_open](#) et [zip_read](#).

7.115.11 zip_entry_compressionmethod() : Lit la méthode de compression utilisée sur un dossier d'archive

string **zip_entry_compressionmethod** (resource [zip_entry](#))

[zip_entry_compressionmethod](#) retourne la méthode de compression utilisée sur le dossier d'archive [zip_entry](#). Le paramètre [zip_entry](#) est un dossier d'archive valide, retourné par [zip_read](#).

Voir aussi [zip_open](#) et [zip_read](#).

7.115.12 zip_entry_filesize() : Lit la taille décompressée d'un dossier d'archive

int **zip_entry_filesize** (resource [zip_entry](#))

[zip_entry_filesize](#) retourne la taille décompressée du dossier d'archive [zip_entry](#). Le paramètre [zip_entry](#) est un dossier d'archive valide, retourné par [zip_read](#).

Voir aussi [zip_open](#) et [zip_read](#).

7.115.13 zip_entry_name() : Lit le nom du dossier d'archive

string **zip_entry_name** (resource [zip_entry](#))

[zip_entry_name](#) retourne le nom du dossier d'archive [zip_entry](#). Le paramètre [zip_entry](#) est un dossier d'archive valide, retourné par [zip_read](#).

Voir aussi [zip_open](#) et [zip_read](#).

7.115.14 zip_entry_open() : Ouvre un fichier d'archive en lecture

bool **zip_entry_open** (resource [zip](#), resource [zip_entry](#), string [mode](#))

[zip_entry_open](#) ouvre un fichier dans une archive ZIP. Le paramètre [zip](#) est une ressource d'archive valide, retournée par [zip_open](#). Le paramètre [zip_entry](#) est une ressource de fichier retournée par [zip_read](#). Le paramètre optionnel [mode](#) peut prendre les mêmes valeurs que ceux de la fonction [fopen](#).

Note

Actuellement, [mode](#) est ignoré et prend toujours la valeur "rb". C'est lié au fait que cette extension PHP propose uniquement l'accès en lecture. Reportez vous à la fonction [fopen](#) pour plus de détails sur les différents modes, y compris le mode "rb".

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note

Contrairement à [fopen](#) et les autres fonctions de fichiers, la valeur retournée par [zip_entry_open](#) indique uniquement le résultat de l'opération, et n'est pas nécessaire pour la lecture ou la fermeture du fichier de dossier d'archive.

Voir aussi [zip_entry_read](#) et [zip_entry_close](#).

7.115.15 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 jusqu'à `length` octets dans le fichier représenté par `zip_entry`. Si `length` est omis, alors [zip_entry_read](#) essaiera de lire un bloc de 1024 octets. Le paramètre `zip_entry` est un dossier d'archive valide, retourné par [zip_read](#).

Note

Le paramètre `length` doit être la taille non-compressée que vous voulez lire.

[zip_entry_read](#) retourne les données lues, ou bien `FALSE` si la fin du fichier est atteinte.

Voir aussi [zip_entry_open](#), [zip_entry_close](#) et [zip_entry_filesize](#).

7.115.16 zip_open() : Ouvre une archive ZIP

```
resource zip_open ( string filename )
```

[zip_open](#) ouvre une nouvelle archive ZIP. Le paramètre `filename` est le nom du fichier ZIP à ouvrir.

[zip_open](#) retourne une ressource, à utiliser plus tard avec les fonctions [zip_read](#) et [zip_close](#), ou bien retourne `FALSE` si `filename` n'a pu être ouvert.

Voir aussi [zip_read](#) et [zip_close](#).

7.115.17 zip_read() : Lit la prochaine entrée dans une archive ZIP

```
resource zip_read ( resource zip )
```

[zip_read](#) lit la prochaine entrée dans l'archive ZIP. Le paramètre `zip` est une archive valide, retournée par [zip_open](#).

[zip_read](#) retourne une ressource de dossier d'archive, à utiliser plus tard avec les autres fonctions de la librairie, ou `FALSE` s'il n'y a plus d'entrées à lire dans l'archive ZIP.

Voir aussi [zip_open](#), [zip_close](#), [zip_entry_open](#) et [zip_entry_read](#).

7.116 Compression Zlib

7.116.1 Introduction

Ce module donne accès aux fichiers compressés avec l'utilitaire gzip, grâce à des fonctions semblables aux fonctions d'accès aux fichiers, qui fonctionnent avec les fichiers compressés ou non (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.

En version 4.3.0, ce préfixe spécial est remplacé par `zlib://` pour éviter les ambiguïtés avec les fichiers contenant des deux points `:`.

Cette fonctionnalité requiert une librairie C qui fournisse la fonction `fopencookie()`. A ma connaissance, la librairie GNU libc est la seule à fournir cette fonctionnalité.

7.116.2 Pré-requis

Ce module utilise les fonctions de la librairie zlib de Jean-loup Gailly et Mark Adler. Vous devez utiliser zlib version `>= 1.0.9` avec ce module.

7.116.3 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 librairie supplémentaire pour disposer de ces fonctions.

Note

Le support intégré de zlib est disponible depuis PHP 4.3.0.

7.116.4 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 comprimer de manière transparente les pages PHP à la volée, si le navigateur du visiteur le support. Voici donc les trois options à utiliser dans le `php.ini`.

Nom	Par défaut	Modifiable
<code>zlib.output_compression</code>	<code>"Off"</code>	<code>PHP_INI_ALL</code>
<code>zlib.output_compression_level</code>	<code>"-1"</code>	<code>PHP_INI_ALL</code>
<code>zlib.output_handler</code>	<code>""</code>	<code>PHP_INI_ALL</code>

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [ini_set](#).

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.

Vous pouvez utiliser `ini_set` pour désactiver cette option dans votre script, si les en-têtes n'ont pas encore été envoyés. Si vous émettez un en-tête de type "Content-Type: image/", la compression sera désactivée aussi (afin de parer à un bug de Netscape). Vous pouvez le réactiver en ajoutant la ligne `"ini_set('zlib.output_compression', 'On')"` après que les en-têtes aient été envoyés.

Cette option accepte aussi des valeurs entières au lieu des booléens, "On"/"Off", ce qui vous permet de configurer la taille du buffer de sortie (par défaut, il vaut 4ko).

Note

`output_handler` doit être laissé à vide si cette option est activée. Sinon, vous devez utiliser `zlib.output_handler`.

`zlib.output_compression_level` integer

Niveau de compression utilisé pour la compression de sortie.

`zlib.output_handler` string

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.

7.116.5 Types de ressources

Cette extension ne définit aucune ressource.

7.116.6 Constantes prédefinies

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*)

7.116.7 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

```
<?php
$filename = tempnam ('/tmp', 'zlibtest').'.gz';
print "<html>\n<head></head>\n<body>\n<pre>\n";
$s = "C'est juste un test, test, test, test, test, test, test, test!\n";

// ouvre le fichier en écriture avec la compression maximale
$zp = gzopen($filename, "w9");

// écrit uen chaîne dans le fichier
gzwrite($zp, $s);
```

```
// ferme le fichier
gzclose($zp);

// ouvre le fichier en lecture
$zp = gzopen($filename, "r");

// lit 3 caractères
print gzread($zp, 3);

// affiche le reste du fichier, puis le ferme
gzpassthru($zp);

print "\n";

// ouvre le fichier et affiche le fichier une seconde fois
if (readgzfile($filename) != strlen($s)) {
    echo "Erreur avec les fonctions zlib!";
}
unlink($filename);
print "</pre>\n</h1></body>\n</html>\n";

?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes prédéfinies](#)
- [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](#)

7.116.9 gzcompress() : Comprime une chaîne

string **gzcompress** (string data , int level)

gzcompress comprime la chaîne ^{data}, avec le niveau de de compression ^{level}, en utilisant le format de compression zlib. La fonction retourne ^{FALSE} si une erreur survient. Le paramètre optionnel ^{level} peut prendre des valeurs comprises entre 0 et 9.

Pour plus de détails sur l'algorithme de compression ZLIB, reportez-vous à la RFC "[ZLIB Compressed Data Format Specification version 3.3](#)" (RFC 1950).

Note

Ce **n'est pas** la même chose qu'une compression gzip, qui inclut aussi des en-têtes. Voyez la fonction `gzencode` pour cela.

Voir aussi `gzdeflate`, `gzinflate`, `gzuncompress` et `gzencode`.

7.116.10 gzdeflate() : Comprime une chaîne avec la méthode DEFLATE

```
string gzdeflate ( string data , int level )
```

`gzdeflate` comprime la chaîne `data` avec la méthode DEFLATE, ou `FALSE` si une erreur survient. Le paramètre optionnel `level` peut prendre des valeurs comprises entre 0 (pas de compression) et 9 (compression maximale). Par défaut, le niveau de compression est le niveau de la librairie Zlib.

Pour plus de détails sur l'algorithme de compression DEFLATE, reportez-vous à la RFC [DEFLATE Compressed Data Format Specification version 1.3](#) (RFC 1951).

Voir aussi `gzinflate`, `gzcompress`, `gzuncompress` et `gzencode`.

7.116.11 gzencode() : Crée une chaîne compressée

```
string gzencode ( string data , int level , int encoding_mode )
```

`gzencode` retourne la version compressée de la chaîne `data`, compatible avec le résultat de la commande `gzip`. Cette fonction retourne `FALSE` si une erreur survient. Le paramètre optionnel `level` peut prendre des valeurs comprises entre 0 (pas de compression) et 9 (compression maximale). Par défaut, le niveau de compression est le niveau de la librairie Zlib.

Vous pouvez aussi utiliser les constantes `FORCE_GZIP` (la valeur par défaut) ou `FORCE_DEFLATE` comme troisième paramètre optionnel `encoding_mode`. Si vous utilisez `FORCE_DEFLATE`, vous obtiendrez une chaîne réduite (deflated) avec les en-têtes Zlib ad hoc, mais sans la somme de contrôle finale CRC32.

Note

`level` a été ajouté en PHP 4.2. Avant cette version `gzencode` n'utilisait que le paramètre `data` et optionnellement `encoding_mode`.

La chaîne résultante contient les en-têtes et la structure de données appropriée pour en faire un fichier .gz standard, c'est à dire :

Créer un fichier gzip

```
<?php
    $data = implode("", file("bigfile.txt"));
    $gzdata = gzencode($data, 9);
    $fp = fopen("bigfile.txt.gz", "w");
    fwrite($fp, $gzdata);
    fclose($fp);
?>
```

Pour plus de détails sur le format de fichier GZIP, reportez-vous à la RFC [GZIP file format specification version 4.3](#) (RFC 1952).

Voir aussi [gzcompress](#) [gzuncompress](#) [gzdeflate](#) et [gzinflate](#) .

7.116.12 gzeof() : Indique si la fin d'un fichier compressé est atteinte

```
int gzeof ( resource  zp )
```

[gzeof](#) retourne ^{TRUE} si le pointeur du fichier compressé ^{zp} a atteint la fin du fichier, ou si une erreur s'est produite. La fonction retourne ^{FALSE} sinon.

^{zp} doit être un fichier compressé valide, retourné par la fonction [gzopen](#) .

7.116.13 gzfile() : Lit la totalité d'un fichier compressé

```
array gzfile ( string  filename , int  use_include_path )
```

[gzfile](#) est identique à la fonction [readgzfile](#) , hormis le fait que [gzfile](#) retourne le fichier sous forme de tableau.

Vous pouvez utiliser le second paramètre optionnel en lui donnant la valeur de ¹ , si vous voulez rechercher le fichier dans l' [include_path](#) .

Voir aussi [readgzfile](#) et [gzopen](#) .

7.116.14 gzgetc() : Lit un caractère dans un fichier compressé

```
string gzgetc ( resource  zp )
```

[gzgetc](#) retourne une chaîne contenant un caractère unique (décompressé), lu dans le fichier représenté par ^{zp} . [gzgetc](#) retourne ^{FALSE} si la fin du fichier est atteinte (contrairement à [gzeof](#)).

^{zp} doit être un fichier compressé valide, retourné par la fonction [gzopen](#) .

Voir aussi [gzopen](#) et [gzgets](#) .

7.116.15 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).

Si une erreur survient, [gzgets](#) retourne ^{FALSE} .

^{zp} doit être un fichier compressé valide, retourné par la fonction [gzopen](#) .

Voir aussi [gzopen](#) , [gzgetc](#) et [fgets](#) .

7.116.16 gzgetss() : Lit une ligne dans un fichier compressé, et supprime les balises

```
string gzgetss ( resource $p , int $length , string $allowable_tags )
```

[gzgetss](#) est identique à la fonction [gzgets](#) , hormis le fait que [gzgetss](#) supprime toutes les balises HTML et PHP dans le texte lu.

Vous pouvez utiliser le troisième paramètre optionnel `$allowable_tags` pour indiquer les balises que vous souhaitez conserver.

Note
<code>\$allowable_tags</code> a été ajouté en PHP 3.0.13, et PHP 4.0b3.

Voir aussi [gzgets](#) , [gzopen](#) et [strip_tags](#) .

7.116.17 gzinflate() : Décompresse une chaîne

```
string gzinflate ( string $data , int $length )
```

[gzinflate](#) prend la chaîne `$data` compressée avec [gzdeflate](#) et retourne la chaîne originale, décompressée, ou bien `FALSE` en cas d'erreur. [gzinflate](#) retourne une erreur si la chaîne décompressée est plus de 256 fois plus longue que la chaîne compressée `$data` ou plus grande que la taille de `$length` octets, passé comme paramètre optionnel.

Voir aussi [gzcompress](#) , [gzuncompress](#) , [gzdeflate](#) et [gzencode](#) .

7.116.18 gzopen() : Ouvre un fichier compressé avec gzip

```
resource gzopen ( string $filename , string $mode , int $use_include_path )
```

[gzopen](#) ouvre le fichier `$filename` , compressé avec l'utilitaire gzip, en écriture ou en lecture. Le paramètre de mode `$mode` prend les mêmes valeurs que dans la fonction [fopen](#) ("rb" ou "wb"), mais il accepte aussi un entier, qui représente un niveau de compression ("wb9") ou une heuristique : "f" pour filtrer les données "wb6f", 'h' pour la compression Huffman seulement "wb1h". (voir aussi la description de `deflateInit2` dans `zlib.h` pour plus d'informations sur le paramètre d'heuristique).

[gzopen](#) peut aussi servir pour lire des fichiers qui ne sont pas compressés. Dans ce cas, [gzread](#) lira le fichier, sans décompression.

[gzopen](#) retourne une ressource de fichier. Tout ce que vous lirez à partir de ce pointeur sera décompressé à la volée de manière transparente.

Si l'ouverture du fichier échoue, [gzopen](#) retourne `FALSE` .

Vous pouvez utiliser le troisième paramètre optionnel en lui donnant la valeur de ¹ , si vous voulez rechercher le fichier dans l' `$include_path` .

Exemple avec gzopen
<pre><?php \$fp = gzopen ("/tmp/file.gz", "r"); ?></pre>

Voir aussi [gzclose](#) .

7.116.19 gzpassthru() : Affiche le contenu d'un fichier en une seule fois

`int gzpassthru ( resource zp )`

[gzpassthru](#) lit toutes les données du fichier `zp` jusqu'à la fin, puis affiche le résultat (décompressé).

Si une erreur survient, la fonction retourne `FALSE` , sinon, elle retourne `TRUE` .

`zp` doit être un fichier compressé valide, retourné par la fonction [gzopen](#) .

Le fichier est alors fermé lorsque [gzpassthru](#) a terminé sa lecture. `zp` n'est plus utilisable après cela.

7.116.20 gzputs() : Ecrit une chaîne dans un fichier compressé

`int gzputs ( resource zp , string str , int length )`

[gzputs](#) est un alias de la fonction [gzwrite](#) , et lui est identique en tous points.

7.116.21 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 (décompressé) ont été lus, ou bien que la fin du fichier a été atteinte.

Exemple avec [gzread](#)

```
<?php
// Lit le contenu d'un fichier dans une chaîne
$filename = "/usr/local/something.txt.gz";
$zd = gzopen($filename, "r");
$content = gzread($zd, 10000);
gzclose($zd);
?>
```

Voir aussi [gzwrite](#) , [gzopen](#) , [gzgets](#) , [gzgetss](#) , [gzfile](#) et [gzpassthru](#) .

7.116.22 gzrewind() : Recommence la lecture du fichier

`int gzrewind ( resource zp )`

[gzrewind](#) replace le pointeur de position du fichier `zp` au début du fichier.

Si une erreur survient, la fonction retourne 0.

`zp` doit être un fichier compressé valide, retourné par la fonction [gzopen](#) .

Voir aussi [gzseek](#) et [gztell](#) .

7.116.23 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`, compté en octets depuis le début du fichier. C'est l'équivalent de la fonction `gzseek( zp, offset, SEEK_SET )`, en 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 position indiquée.

gzseek retourne 0 en cas de réussite, sinon, elle retourne -1. Notez que placer le pointeur au delà de la fin du fichier n'est pas une erreur.

Voir aussi gztell et gzrewind.

7.116.24 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`, c'est à dire sa position dans le fichier.

Si une erreur survient, gztell retourne `FALSE`.

`zp` doit être un fichier compressé valide, retourné par la fonction gzopen.

Voir aussi gzopen, gzseek et gzrewind.

7.116.25 gzuncompress() : Décompresse une chaîne compressée

`string gzuncompress ( string data , int length )`

gzuncompress prend la chaîne `data`, compressée avec la fonction gzcompress et retourne la chaîne originale, ou bien `FALSE` en cas d'erreur. gzuncompress retourne une erreur si la chaîne décompressée est plus de 256 fois plus longue que la chaîne compressée `data` ou plus grande que la taille de `length` octets, passé comme paramètre optionnel.

Voir aussi gzdeflate, gzinflate, gzcompress et gzencode.

7.116.26 gzwrite() : Ecrit dans un fichier compressé, en gzip

`int gzwrite ( resource zp , string string , int length )`

gzwrite écrit le contenu de la chaîne `string` dans le fichier compressé `zp`. Si le paramètre `length` est fourni, l'écriture s'arrêtera lorsque `length` octets décompressés ont été écrit, ou à la fin de la chaîne `string` : le premier qui arrive.

Notez que si `length` est donné, la directive magic_quotes_runtime sera ignorée, et les slashes ne seront pas supprimés de la chaîne `string`.

Voir aussi [gzread](#) , [gzopen](#) et [gzputs](#) .

7.116.27 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.

[readgzfile](#) retourne le nombre d'octets décompressés qui ont été lus dans le fichier. Si une erreur survient, [readgzfile](#) retourne `FALSE` . Un message d'erreur sera alors affiché, à moins que la fonction n'ait été appelée avec `@readgzfile` .

Le fichier `filename` sera ouvert dans le système de fichier, et son contenu sera affiché.

Vous pouvez utiliser le troisième paramètre optionnel en lui donnant la valeur de ¹ , si vous voulez rechercher le fichier dans l' `include_path` .

Voir aussi [gzpassthru](#) , [gzfile](#) et [gzopen](#) .

7.116.28 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. Les valeurs possibles sont `gzip` , `deflate` et `FALSE` .

Voir aussi la directive `zlib.output\_compression` .

8 Développer PHP 4.0

Travailler au Coeur de PHP

Introduction

Parfois, PHP tel qu'il est n'est simplement pas suffisant. Même si ces situations sont très rares pour les utilisateurs standard, les applications professionnelles de PHP l'amèneront inmanquablement aux limites de ses possibilités, en terme de vitesse ou de fonctionnalité. Les nouvelles fonctionnalités ne peuvent pas toujours être implémentées nativement, étant donné les limitations et restrictions du langage lorsque l'on essaie d'inclure une énorme librairie dans chaque script. Une autre méthode doit donc être utilisée.

Dès que ce point est atteint, il est temps de plonger au coeur de PHP, et de regarder son contenu, ainsi que le code C qui s'y trouve.

8.1 Introduction

"Améliorer PHP" est plus vite dit que fait. PHP est devenu un outil mature, constitué de plusieurs méga octets de code source, et travailler dans un système comme celui-ci impose d'apprendre quelque peu avant de s'y lancer. Lorsque nous avons constitué ce chapitre, nous avons choisi une approche d'apprentissage par l'exemple. Ce n'est pas l'approche la plus scientifique ou professionnelle, mais c'est celle qui est la plus passionnante et qui donne les meilleurs résultats. Dans les sections suivantes, nous allez apprendre à utiliser les fonctionnalités avancées de l'API Zend. L'alternative serait de présenter les fonctionnalités, le design, les astuces, les trucs.. comme un tout, d'une seule pièce, afin de donner une vue d'ensemble avant de commencer à bidouiller. Même si c'est la "meilleure" méthode, cela se révèle très frustrant, et très gourmand en énergie et en temps, ce qui fait que nous avons décidé de prendre l'autre approche.

Notez que même si ce chapitre essaie de fournir autant de connaissances que possible sur les rouages internes de PHP, il est impossible de donner une vue complète du processus d'amélioration de PHP, qui fonctionne à 100% dans tous les cas. PHP est tellement vaste et complexe que vous ne pourrez comprendre son fonctionnement qu'en vous familiarisant par la pratique. Nous vous encourageons donc à travailler dans les sources.

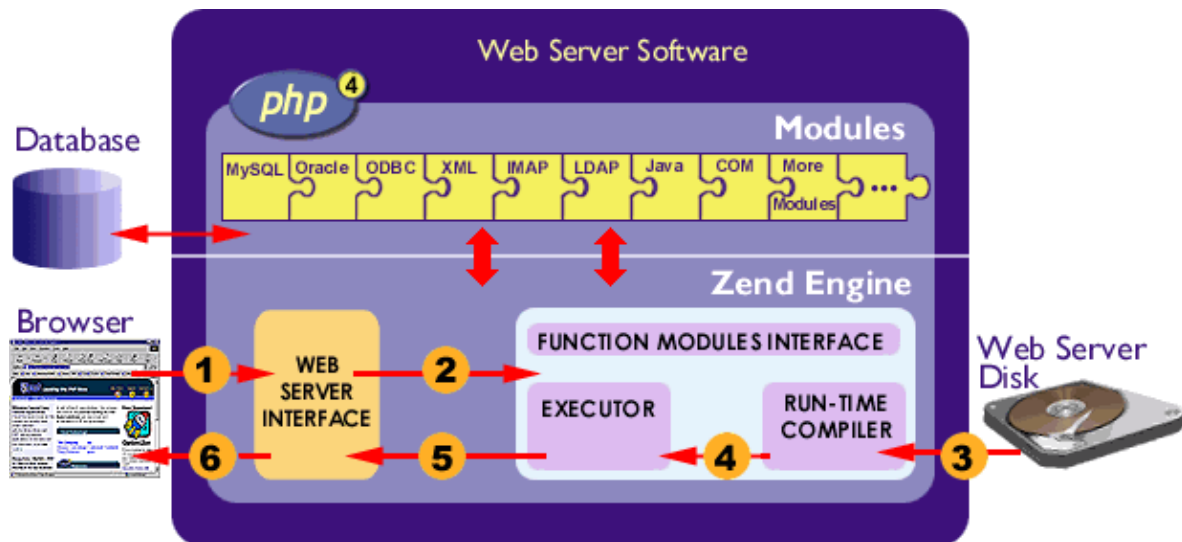
8.1.1 Qu'est ce que Zend? et qu'est ce que PHP?

Le nom de **Zend** fait référence au moteur du langage, le coeur de PHP. Le terme de **PHP** fait référence au système complet, tel qu'il apparait à l'extérieur. Cela peut paraître confus, mais ce n'est pas si compliqué (voyez [The internal structure of PHP](#)). Pour implémenter un interpréteur de scripts web, il vous faut :

1. Un **interpréteur** qui analyse le code source, le traduit et l'exécute.
2. Une couche **fonctionnelle** qui implémente les fonctionnalités du langage (les fonctions, etc...).
3. Une **interface** qui dialogue avec le serveur web, etc...

Zend prend en charge la partie 1 complètement, et une partie de la partie 2; PHP prend en charge la partie 2 et la partie 3. Ensemble, ils forment le logiciel PHP. Zend lui-même ne représente que le coeur du langage, implémentant les fonctions les plus basiques. PHP contient tous les modules, et les fonctionnalités incroyables du langage.

The internal structure of PHP.



Les sections suivantes présentent comment PHP peut être amélioré, et comment le faire.

8.2 Possibilités d'extension

Comme illustré sur la figure The internal structure of PHP, ci-dessus, PHP peut être amélioré en trois endroits : les modules externes, les modules internes et le moteur Zend. Nous allons présenter ces trois cas.

8.2.1 Modules externes

Les modules externes peuvent être chargés dans le script au moment de l'exécution, en utilisant la fonction `dl`. Cette fonction charge un objet partagé depuis le disque, et rend ses fonctionnalités disponibles au script exécutant. A la fin du script, le module est déchargé de la mémoire. Cette méthode a des avantages et des inconvénients, comme décrits dans la table qui suit :

Avantages	Inconvénients
Les modules externes n'imposent pas la recompilation de PHP.	Les objets partagés doivent être chargés à chaque invocation de script, ce qui peut être très lent.
La taille de PHP reste faible, en exportant certaines fonctionnalités.	Les fichiers externes encombre votre disque.
	Chaque script qui a besoin du module externe doit l'inclure explicitement dans le script avec la fonction <code>dl</code> , ou bien le fichier <code>php.ini</code> doit être modifié (ce qui n'est pas toujours une solution viable).

Pour résumer, les modules externes sont excellents pour les produits de tierce parties, pour les fonctionnalités rarement utilisées, ou dans le cadre de test ou de développement. Pour développer de nouvelles fonctionnalités, les modules externes sont les plus efficaces. Dans le cadre d'une utilisation fréquente, d'implémentation de grande taille, ou de code complexe, les inconvénients deviennent majeurs.

Les tierces parties devraient utiliser la balise `extension` dans le fichier `php.ini` pour créer des modules externes additionnels à PHP. Ces modules sont complètement détachés de l'exécutable principal, ce qui est très pratique en environnement commercial. Les distributeurs commerciaux n'ont plus qu'à envoyer les modules additionnels compilés, sans avoir à se préoccuper de compiler PHP lui-même.

8.2.2 Modules internes

Les modules internes sont compilés directement dans PHP, et chargés avec chaque processus PHP. Leur fonctionnalités sont immédiatement disponibles avec chaque script. Comme les modules externes, les modules internes ont leurs avantages et leurs inconvénients :

Avantages	Inconvénients
Pas besoin de charger spécifiquement un module. Les fonctionnalités sont immédiatement disponibles.	Les modifications du module imposent la recompilation de PHP.
Pas de fichiers externes encombrants. Tout est inclus dans l'exécutable PHP.	Le programme PHP grossit, et prend plus de mémoire.

Les modules PHP sont les plus efficaces lorsque vos librairies de fonctions sont stables, et qu'elles requièrent des performances au dessus de la moyenne, ou qu'elles sont utilisées de manière intensive. Le besoin de recompiler PHP à chaque modification est compensé par le fait que PHP est alors plus efficace, et plus rapide. Cependant, les modules internes ne sont pas idéaux pour le développement.

8.2.3 Zend Engine

Bien sur, les extensions peuvent aussi être implémentées directement dans le moteur Zend. Cette stratégie est excellent si vous avez besoin de modification dans le comportement du langage, ou si vous avez besoin de fonctions spéciales, bâties directement dans le cœur du langage. En général, les modifications du moteur Zend sont à éviter. Ces modifications peuvent conduire à des incompatibilités avec le reste du monde PHP, et personne ne pourra alors s'adapter à votre version spéciale de Zend. Les modifications risquent d'être écrasées à la prochaine version de PHP, et vous imposera de tout refaire pour suivre l'évolution. De ce fait, cette méthode est considérée comme mauvaise, et, étant donné son caractère exceptionnel, elle ne sera pas couverte dans ce livre.

8.3 Source Layout

Note
Avant de progresser dans ce chapitre, nous vous conseillons de récupérer une version propre et non polluée des sources de votre serveur web préféré. Nous travaillerons avec Apache (disponible à http://www.apache.org/) et, bien entendu, avec PHP (disponible à http://www.php.net/ – est-il vraiment besoin de le préciser?). Assurez vous que vous pouvez compiler PHP par vous même. Nous ne traiterons pas cet aspect du problème ici, car vous devriez être déjà en mesure de le faire avant de vous plonger dans ce chapitre.

Avant que nous ne commençons à discuter du code lui-même, vous devez vous familiariser avec l'arbre des sources PHP, pour que vous puissiez naviguer rapidement dans les fichiers. C'est une compétence nécessaire pour implémenter et déboguer des modules.

La table suivante décrit le contenu des dossiers les plus importants :

Dossier	Contenu
php4	Le dossier principal de PHP, ses sources et fichiers d'entêtes. Vous trouverez là toutes les définitions des API PHP, les macros, etc... Tout se trouve rangé dans ce dossier. Ce dossier est important.
php4/ext	Dépôt des modules, statiques ou dynamiquement chargés. Par défaut ce sont les modules PHP 'officiels' qui sont intégrés dans la distribution principale. Depuis PHP 4.0, il est possible de compiler ces extensions standards comme module dynamiquement chargeable (tout au moins, ceux qui le supportent).
php4/main	Ce dossier contient les principales macros PHP, et leur définition. Ce dossier est important.
php4/pear	Dossier du Repository à Extensions et Applications de PHP (alias PEAR en anglais, PHP Extension and Application Repository). Ce dossier contient les fichiers de base de PEAR.

php4/sapi	Contient le code pour les différentes interfaces avec les serveurs HTTP.
php4/TSRM	Dossier du "Thread Safe Resource Manager" (TSRM) pour Zend et PHP.
php4/Zend	Dossier du moteur Zend. Vous y trouverez notamment les définitions des macros et de l'API. Ce dossier est important.

Détailler tous les fichiers inclus dans les sources PHP dépasse le cadre de ce chapitre. Cependant, nous allons en observer certains :

- `php4/main/php.h` , situé dans le dossier principal de PHP. Ce fichier contient la plupart des définitions de macros et API de PHP.
- `php4/Zend/zend.h` , situé dans le dossier principal Zend. Ce fichier contient la plupart des définitions de macros et API de Zend.
- `php4/Zend/zend_API.h` , est aussi situé dans le dossier Zend, et définit les API Zend.

Nous vous recommandons aussi d'étudier les sous-inclusions de ces fichiers. Par exemple, celles qui sont liées à l'exécuteur Zend, au fichier d'initialisation de PHP, etc... Après avoir lu ces fichiers, prenez un moment pour naviger dans les sources, pour étudier les inter dépendances des fichiers et modules : comment ils sont liés, et particulièrement, comment ils s'utilisent mutuellement. Cela vous aidera à vous adapter au style de codage de PHP. Pour ajouter des fonctionnalités à PHP, il vous faudra vous adapter à ce style.

8.3.1 Conventions sur les extensions

Zend est bâti autour de certaines conventions. Pour éviter d'enfreindre les standards, il est recommandé de suivre les règles suivantes.

8.3.2 Macros

Pour presque toutes les tâches importantes, Zend fournit des macros prédéfinies, qui sont très pratiques. Les tables et illustrations des sections qui suivent décrivent les fonctions, structures et macros de base. Les définitions des macros sont disponibles en général dans les fichiers d'entête `zend.h` et `zend_API.h` . Nous vous suggérons d'étudier ces fichiers après avoir lu ce chapitre. Mais vous pouvez aussi les lire dès maintenant, même si tout ne sera pas compréhensible au premier abord.

8.3.3 Gestion de la mémoire

La gestion des ressources est un problème crucial, notamment sur des serveurs. Une de ces ressources est la mémoire, qui doit être gérée avec beaucoup de soin. La gestion de la mémoire est partiellement laissée à Zend, et il est recommandé de l'exploiter, pour deux raisons : avec une couche d'abstraction, Zend prend le contrôle complet des allocations mémoire. Zend est capable de déterminer si un bloc est utilisé, de libérer automatiquement des blocs inutilisés, et les blocs dont les références sont perdues. Cela évite les fuites mémoire. Les fonctions à utiliser sont les suivantes :

Fonction	Description
<code>emalloc</code>	Remplacement pour <code>malloc</code> .
<code>efree</code>	Remplacement pour <code>free</code> .
<code>estrdup</code>	Remplacement pour <code>strdup</code> .
<code>estrndup</code>	Remplacement pour <code>strndup</code> . Plus rapide que <code>estrdup</code> et compatible avec les données binaires. C'est la fonction recommandée si vous connaissez la taille de votre chaîne avant de la dupliquer.
<code>ecalloc</code>	Remplacement pour <code>calloc</code> .
<code>erealloc</code>	Remplacement pour <code>realloc</code> .

`emalloc` , `estrdup` , `estrndup` , `ecalloc` , et `erealloc` allouent de la mémoire interne; `efree` libère cette mémoire. La mémoire gérée par les fonctions `e*` est considérée comme locale au processus courant, et détruite

aussitôt que le script est terminé.

Attention

Pour allouer de la mémoire résident qui survivra à la fin du script courant, vous pouvez utiliser `malloc` et `free`. Mais cela doit être fait avec beaucoup de soin, et en conjonction avec les exigences de l'API Zend; Sinon, vous risquez des fuites mémoire.

Zend dispose aussi d'un gestionnaire de ressources compatible avec les threads, pour exploiter au mieux les serveurs web multi-threads. Cela vous impose d'allouer les structures locales de toutes vos variables globales pour permettre aux threads concurrents de s'exécuter. Etant donné que le mode compatible avec les threads n'est pas terminé lorsque ceci a été écrit, il ne sera pas détaillé ici.

8.3.4 Fonctions de fichiers et dossiers

Les fonctions de dossiers et fichiers suivantes doivent être utilisées dans les modules Zend. Ils fonctionnent exactement comme leur équivalent en C, mais fournissent un dossier de travail virtuel, au niveau du thread.

Fonction Zend	Fonction C classique
<code>v_getcwd</code>	<code>getcwd</code>
<code>v_fopen</code>	<code>fopen</code>
<code>v_open</code>	<code>open</code>
<code>v_chdir</code>	<code>chdir</code>
<code>v_getwd</code>	<code>getwd</code>
<code>v_chdir_file</code>	Prend le chemin comme argument, et modifie le dossier de travail, en le remplaçant par ce chemin.
<code>v_stat</code>	<code>stat</code>
<code>v_lstat</code>	<code>lstat</code>

8.3.5 Gestion des chaînes de caractères

Les chaînes de caractères sont gérées un peu différemment par le Zend Engine, par rapport aux autres types comme entiers, booléens, etc..., qui ne requièrent pas de mémoire additionnelle pour stocker leur valeurs. Si vous souhaitez retourner une chaîne depuis une fonction, introduire une nouvelle chaîne dans la table des symboles, ou faire encore autre chose, vous devez vous assurer que la mémoire utilisée par la chaîne a bien été allouée, en utilisant les fonctions sus-citées ². Cela n'est peut être pas encore évident mais gardez le à l'esprit : vous y reviendrons bientôt.

8.3.6 Types complexes

Les types complexes, comme les tableaux et objets nécessitent un traitement spécifique. Zend dispose d'une API unique pour ces types, qui sont stockés dans des tables de hash.

Note

Pour réduire la complexité des exemples, nous travaillerons qu'avec des types simples, comme les entiers. Une section sur la création des types complexes est disponible plus loin.

8.4 Le système de squelette automatique de PHP

PHP 4 dispose d'un système de squelette automatique très souple. Tous les modules sont placés dans un sous-dossier du dossier ^{ext}. En plus de ses propres sources, chaque module est constitué d'un fichier config.m4, qui sert à la configuration des extensions (par exemple, consultez

http://www.gnu.org/manual/m4/html_mono/m4.html).

Tout ces fichiers sont générés automatiquement, ainsi que le fichier `.cvsignore`, par un script shell appelé `ext_skel` qui est installé dans le dossier `ext`. En argument, il prend le nom du module à créer. Le script va alors bâtir un dossier complet, du même nom.

Etape par étape, voici le processus :

```
~/cvs/php4/ext:> ./ext_skel --extname=mon_module
Creating directory mon_module
Creating basic files: config.m4 .cvsignore mon_module.c php_mon_module.h CREDITS EXPERIMENTAL tests/001.phpt mon_module.php [done].
```

Pour utiliser votre nouvelle extension, il suffit d'exécuter les étapes suivantes :

```
1. $ cd ..
2. $ vi ext/mon_module/config.m4
3. $ ./buildconf
4. $ ./configure --[with|enable]-my_module
5. $ make
6. $ ./php -f ext/my_module/my_module.php
7. $ vi ext/my_module/my_module.c
8. $ make
```

Recommencez les étapes de 3 à 6 jusqu'à ce que vous soyez satisfait par votre fichier `ext/my_module/config.m4` et l'étape 6 confirmera que votre module est compilé dans PHP. Maintenant, commencez à écrire le code de votre extensions, et répétez le processus aussi souvent que nécessaire.

Ces instructions crée les fichiers sus-cités. Pour inclure le module dans le processus de configuration automatique et de compilation, vous devez exécuter le script `buildconf`, qui génère le script `configure` en fouillant le dossier `ext`, pour y inclure tous les fichiers `config.m4` qu'on y trouve.

Le fichier par défaut `config.m4` qui sert d'illustration, Le fichier config.m4 par défaut est un peu plus complexe :

Le fichier config.m4 par défaut

```
dnl $Id: Extending_Zend_Build.xml,v 1.7 2002/10/06 21:43:02 images Exp $
dnl config.m4 for extension my_module

dnl Comments in this file start with the string 'dnl'.
dnl Remove where necessary. This file will not work
dnl without editing.

dnl If your extension references something external, use with:

dnl PHP_ARG_WITH(my_module, for my_module support,
dnl Make sure that the comment is aligned:
dnl [ --with-my_module          Include my_module support])

dnl Otherwise use enable:

dnl PHP_ARG_ENABLE(my_module, whether to enable my_module support,
dnl Make sure that the comment is aligned:
dnl [ --enable-my_module       Enable my_module support])

if test "$PHP_MY_MODULE" != "no"; then
    dnl Write more examples of tests here...

    dnl # --with-my_module -> check with-path
    dnl SEARCH_PATH="/usr/local /usr"      # you might want to change this
    dnl SEARCH_FOR="include/my_module.h"  # you most likely want to change this
    dnl if test -r $PHP_MY_MODULE/; then # path given as parameter
    dnl     MY_MODULE_DIR=$PHP_MY_MODULE
    dnl else # search default path list
    dnl     AC_MSG_CHECKING([for my_module files in default path])
    dnl     for i in $SEARCH_PATH ; do
    dnl         if test -r $i/$SEARCH_FOR; then
    dnl             MY_MODULE_DIR=$i
    dnl             AC_MSG_RESULT(found in $i)
    dnl         fi
    dnl     done
    dnl fi
    dnl if test -z "$MY_MODULE_DIR"; then
    dnl     AC_MSG_RESULT([not found])
    dnl     AC_MSG_ERROR([Please reinstall the my_module distribution])
    dnl fi

    dnl # --with-my_module -> add include path
    dnl PHP_ADD_INCLUDE($MY_MODULE_DIR/include)

    dnl # --with-my_module -> check for lib and symbol presence
```

```

dn1 LIBNAME=my_module # you may want to change this
dn1 LIBSYMBOL=my_module # you most likely want to change this

dn1 PHP_CHECK_LIBRARY($LIBNAME,$LIBSYMBOL,
dn1 [
dn1 PHP_ADD_LIBRARY_WITH_PATH($LIBNAME, $MY_MODULE_DIR/lib, MY_MODULE_SHARED_LIBADD)
dn1 AC_DEFINE(HAVE_MY_MODULELIB,1,[ ])
dn1 ],[
dn1 AC_MSG_ERROR([wrong my_module lib version or lib not found])
dn1 ],[
dn1 -L$MY_MODULE_DIR/lib -lm -ldl
dn1 ])
dn1
dn1 PHP_SUBST(MY_MODULE_SHARED_LIBADD)

PHP_NEW_EXTENSION(my_module, my_module.c, $ext_shared)
fi

```

Si vous n'êtes pas familier avec les fichiers M4, cela peut paraître un peu désorientant à première vue. Il serait certainement utile que vous vous y mettiez. Mais en fait, c'est très facile :

Note : Tout ce qui est préfixé par `dn1` est un commentaire, et ne sera pas utilisé.

Le fichier `config.m4` est responsable de l'analyse des options de ligne de commande passé à `configure`, lors de la configuration. Cela signifie qu'il doit faire les vérifications de présence des fichiers externes, les configurations et les mises en place requises.

Le fichier `config.m4` crée deux directives dans le script principal : `--with-mon_module` et `--enable-mon_module`. Utilisez la première option pour faire référence à des fichiers externes (comme par exemple `--with-apache` qui fait référence au dossier Apache). Utilisez la seconde option si l'utilisateur n'a qu'à choisir d'activer ou pas l'extension. Indépendamment de l'option de vous choisissez, vous devez mettre en commentaire l'autre, qui est inutile. Par exemple, si vous choisissez `--enable-mon_module`, vous devez enlever le support pour `--with-my_module`, et vice versa.

Par défaut, le fichier `config.m4` créé par `ext_skel` accepte les deux directives et active automatiquement votre extension. L'activation de l'extension est fait en utilisant la macro `PHP_EXTENSION`. Pour changer ce comportement par défaut, et le remplacer par l'inclusion sur demande explicite (en utilisant `--enable-mon_module` ou `--with-mon_module`), changez le test de `$PHP_MY_MODULE` avec `== "yes"` :

```

if test "$PHP_MY_MODULE" == "yes"; then dn1
    Action.. PHP_EXTENSION(mon_module, $ext_shared)
fi

```

. Cela vous imposera d'utiliser l'option `--enable-mon_module` à chaque fois que vous reconfigurerez et recompilerez PHP.

Note: Assurez vous de bien exécuter `buildconf` à chaque fois que vous changez `config.m4` !

Nous nous plongerons plus profondément dans les macros M4 plus loin. Pour le moment, les fichiers par défaut font très bien l'affaire.

8.5 Créer une extension

Nous commencerons avec la création d'une extension très simple au début, et qui ne fera pratiquement rien de plus qu'implémenter une fonction qui retourne l'entier qu'elle reçoit comme paramètre. Une extension simple montre le code source :

```

Une extension simple

/* inclusion des entêtes standards */
#include "php.h"

/* déclaration des fonctions exportées */

```

```

ZEND_FUNCTION(first_module);

/* liste des fonctions compilées pour que Zend sache ce qu'il y a dans ce module */
zend_function_entry firstmod_functions[] =
{
    ZEND_FE(first_module, NULL)
    {NULL, NULL, NULL}
};

/* Informations sur le module */
zend_module_entry firstmod_module_entry =
{
    STANDARD_MODULE_HEADER,
    "First Module",
    firstmod_functions,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NO_VERSION_YET,
    STANDARD_MODULE_PROPERTIES
};

/* implémentation standard "stub" routine pour nous présenter à Zend */
#ifdef COMPILE_DL_FIRST_MODULE
ZEND_GET_MODULE(firstmod)
#endif

/* implémentation de la fonction qui sera accessible depuis PHP */
ZEND_FUNCTION(first_module)
{
    long parameter;

    if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "l", &parameter) == FAILURE) {
        return;
    }

    RETURN_LONG(parameter);
}

```

Ce code contient un module PHP complet. Nous expliquerons le code source en détail ultérieurement, mais nous devons commencer par présenter le code avant le processus de création. (Cela permettra aux plus impatientes d'expérimenter avant de se plonger dans les discussions d'API).

Note

Le source d'exemple fait référence aux fonctionnalités introduites dans les versions de Zend utilisées en PHP 4.1.0 et plus récent, et ne fonctionnera pas avec les vieilles versions de PHP 4.0.x.

8.5.2 Compiler le module

Il y a simplement deux méthodes pour compiler les modules :

- Utiliser le mécanisme de "make", présent dans le dossier ^{ext}, qui permet aussi de compiler les modules chargeables dynamiquement.
- Compiler les sources manuellement.

La première méthode doit toujours être privilégiée, car, depuis PHP 4.0, un mécanisme standardisé et sophistiqué de compilation a été mis en place. Le fait qu'il soit sophistiqué a des inconvénients, malheureusement : il est difficile à comprendre au premier abord. Nous allons fournir une introduction détaillée de ce processus ultérieurement, mais commençons par regarder le fonctionnement avec les fichiers par défaut.

La seconde méthode est bonne pour ceux qui, pour une raison ou une autre, n'ont pas les sources complètes de PHP sous la main, ou n'ont pas accès à tous les fichiers, ou qui aiment simplement pianoter sur leur clavier. Ces cas sont extrêmement rares, mais afin d'être exhaustif, nous allons aussi décrire cette méthode.

Pour compiler le code d'exemple en utilisant le mécanisme standard, copiez le dossier et tous les sous dossier dans le dossier ^{ext} de vos sources PHP. Puis exécutez le script ^{buildconf}, qui va créer une version mise à jour du script ^{configure}, avec les options appropriées pour votre extension. Par défaut, les nouvelles extensions sont désactivées. Comme cela, vous ne démolirez pas votre script de configuration.

Après avoir exécuté ^{buildconf}, ^{configure --help} montre les modules additionnels :

```
--enable-array_experiments BOOK: Enables array experiments
--enable-call_userland BOOK: Enables userland module
--enable-cross_conversion BOOK: Enables cross-conversion module
--enable-first_module BOOK: Enables first module
--enable-infoprint BOOK: Enables infoprint module
--enable-reference_test BOOK: Enables reference test module
--enable-resource_test BOOK: Enables resource test module
--enable-variable_creation BOOK: Enables variable-creation module
```

Le module présenté auparavant Une extension simple peut être activé avec ^{--enable-first_module} ou ^{--enable-first_module=yes}.

Pour compiler vos modules manuellement, vous avez besoin des commandes suivantes :

Action	Commande
Compiler	cc -fpic -DCOMPILE_DL=1 -I/usr/local/include -I. -I.. -I../Zend -c -o <your_object_file> <your_c_file>
Lier	cc -shared -L/usr/local/lib -rdynamic -o <your_module_file> <your_object_file(s)>

La commande pour compiler le module indique simplement au compilateur de générer du code indépendant des positions (^{-fpic} ne doit pas être omis), et de plus, il définit la constante ^{COMPILE_DL} qui indique au code du module qu'il est compilé comme module dynamiquement chargeable (le module de test commence par vérifier cela en premier; nous en reparlerons plus tard). Après ces options, il spécifie un certain nombre de chemin d'inclusion qui devraient être utilisés comme liste minimale pour compiler le tout.

Note: Tous les chemins d'inclusion dans l'exemple sont relatifs au dossier ^{ext}. Si vous compilez depuis un autre dossier, changez les chemins de manière appropriée. Les chemins obligatoires sont le dossier PHP, le dossier Zend, et (si nécessaire) le dossier de votre module.

La commande de liaison est aussi une commande standard de liaison du module comme module dynamique.

Vous pouvez inclure des options d'optimisation dans la commande de compilation, même si nous les avons omis dans cet exemple (mais certaines sont incluses dans le fichier makefile, décrit un peu plus haut).

Note: Compiler et lier manuellement comme module statique de PHP implique une liste d'insctructions très longue, et non discutée ici (ce n'est pas très efficace de taper toutes ces commandes).

8.6 Utiliser des extensions

Suivant le processus de compilation que vous avez choisi, vous vous retrouvez avec un nouvel exécutable PHP lié à votre serveur web (sous forme de CGI), ou bien avec une librairie partagée, sous la forme d'un fichier .so. Si vous avez compilé le fichier d'exemple ^{first_module.c} comme objet partagé, le fichier résultant doit être nommé ^{first_module.so}. Pour l'utiliser, vous devez commencer par copier cette librairie dans un dossier accessible à PHP. Pour un simple test, vous pouvez copier le fichier dans le dossier ^{htdocs} et essayer PHP avec le script Un fichier test pour first_module.so. . Si vous avez compilé l'extension dans l'exécutable PHP, il faut enlever l'appel à la fonction ^{dl}, car les

fonctionnalités du module sont déjà accessibles dans PHP.

Attention

Pour des raisons de sécurité, vous **ne devez pas** mettre vos modules dynamiques dans un dossier accessible à tous. Même si cela peut se faire et que cela vous simplifiera les tests, il est recommandé de les placer dans un dossier séparé, pour les environnements de production.

Un fichier test pour `first_module.so`.

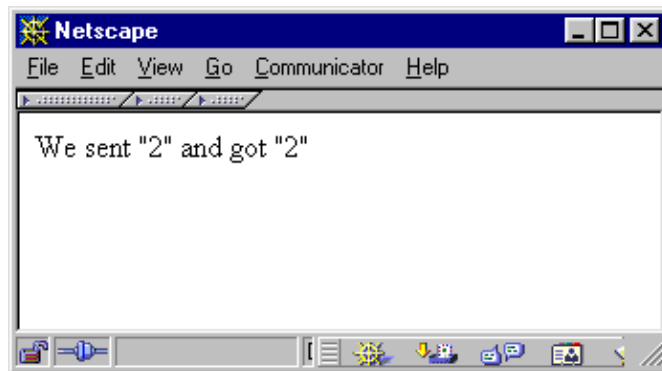
```
<?php
// Décommentez la ligne suivante si nécessaire
// dl("first_module.so");

$params = 2;
$return = first_module($params);

print("Nous avons envoyé \"$params\" et reçu \"$return\"");
?>
```

En exécutant ce script PHP dans votre navigateur web, vous devriez obtenir l'affichage suivant : Affichage résultant de `first_module.php`.

Affichage résultant de `first_module.php`.



Si nécessaire, les modules dynamiques peuvent être appelés dans le script avec la fonction `dl`. Cette fonction recherche la bibliothèque partagée, la charge, et inclut les nouvelles fonctions dans PHP. Ici, le module exporte la fonction `first_module`, qui accepte un paramètre unique, le convertit en entier, et retourne le résultat de la conversion.

Si vous avez réussi à arriver aussi loin, félicitations! Vous avez réussi à compiler votre première extension.

8.7 Résolution de problèmes

En réalité, très peu de problèmes sont possibles avec les modules statiques et dynamiques. Le seul problème qui peut survenir est que le compilateur va se plaindre de définitions qui manquent, ou quelque chose comme ça. Dans ce cas, assurez-vous que les fichiers d'entêtes sont accessibles, et que vous en avez spécifié correctement le chemin. Pour être sûr que tout est bien localisé, installez une version source propre de PHP, et utilisez le script de compilation automatique situé dans `ext`. Cela vous garantira un environnement de compilation propre. Si cela échoue, passez en compilation manuelle.

PHP peut aussi se plaindre que certaines fonctions manquent dans votre module (ce qui ne devrait pas arriver avec les sources d'exemples si vous ne les modifiez pas). Si les noms des fonctions externes que vous utilisez dans votre module sont erronés, ils seront conservés comme "unlinked".

symbols" dans la table des symboles. Durant le chargement dynamique et le linkage, ils ne seront pas résolus, à cause de cette erreur de nommage. Recherchez les déclarations erronées dans votre fichier, ainsi que les références externes incorrectes. Notez que ce problème est spécifique aux modules chargeables dynamiquement. Les erreurs des modules statiques seront détectées au moment de la compilation.

8.8 Présentation des sources

Maintenant que vous avez bâti un environnement de compilation propre, et que vous êtes capables d'inclure des modules PHP dans vos scripts, il est temps de détailler un peu le fonctionnement interne.

8.8.1 Structure de module

Tous les modules PHP ont la structure suivante :

- Fichiers d'entête à inclure (pour inclure les macros, les API, les définitions, etc...)
- Les déclarations C des fonctions exportées (nécessaire pour déclarer le bloc de fonction Zend)
- La déclaration du bloc de fonction Zend
- La déclaration du bloc de module Zend
- L'implémentation de la fonction `get_module`
- L'implémentation de toutes les fonctions exportées

8.8.2 Inclusions des fichiers d'entête

Le seul fichier d'entête que vous devez réellement inclure dans vos modules est le fichier `php.h`, situé dans le dossier de PHP. Ce fichier inclut toutes les macros et API nécessaires pour bâtir le module.

Tip: C'est une bonne pratique que de créer un fichier d'entête séparé pour votre module, qui contient les définitions spécifiques à votre module. Ce fichier d'entête doit contenir toutes les définitions ultérieures des fonctions exportées, et inclure le fichier `php.h`. Si vous créez votre module en utilisant le script `ext_skel`, vous devriez avoir un tel fichier déjà préparé.

8.8.3 Déclaration des fonctions exportées

Pour déclarer des fonctions qui seront exportées (c'est à dire, rendue disponibles dans PHP comme fonction native), Zend fournit un jeu de macros. Une simple déclaration comme celle-ci :

```
ZEND_FUNCTION ( my_function );
```

`ZEND_FUNCTION` déclare une nouvelle fonction C, qui est compatible avec les standards internes de Zend. Cela signifie que la fonction est de type `void` et accepte `INTERNAL_FUNCTION_PARAMETERS` (une autre macro) comme paramètres. De plus, elle préfixe le nom de la fonction avec `zif`. La version longue de la définition ci-dessus est la suivante :

```
void zif_my_function ( INTERNAL_FUNCTION_PARAMETERS );
```

Après remplacement de `INTERNAL_FUNCTION_PARAMETERS`, cela devient :

```
void zif_my_function( int ht
                    , zval * return_value
                    , zval * this_ptr
                    , int return_value_used
                    , zend_executor_globals * executor_globals
                    );
```

Comme l'interpréteur et l'exécuteur ont été séparé l'un de l'autre, dans le package principal de PHP, un nouveau jeu de macros et de fonction a évolué : les API Zend. Comme les API Zend ont pris en charge certaines responsabilités qui appartenait auparavant à PHP, un bon nombre de fonction de PHP ont été réduites à de simple alias de macros, qui effectuent les appels à l'API Zend. La politique recommandée est d'utiliser l'API Zend à chaque fois que possible, car l'ancienne API n'est conservée que pour des raisons de compatibilité. Par exemple, les types `zval` et `pval` sont identiques. `zval` est la définition pour Zend. `pval` est la définition pour PHP (en fait, `pval` est actuellement un alias de `zval`). Comme la macro `INTERNAL_FUNCTION_PARAMETERS` est une macro Zend, la déclaration ci-dessus contient un `zval`. Lorsque vous écrivez du code, il faut toujours utiliser les `zval` pour être compatible avec la nouvelle API Zend API.

La liste de paramètres de cette déclaration est très importante. Gardez bien ces paramètres en tête, et voyez [Paramètres Zend pour les fonctions PHP](#) pour plus de détails.

Paramètre	Description
ht	Le nombre d'arguments passé à la fonction Zend. N'indiquez pas cette valeur vous même, mais utilisez <code>ZEND_NUM_ARGS()</code> pour obtenir la valeur.
return_value	Cette variable est utilisée pour passer les valeurs retournées par votre fonction, à PHP. L'utilisation de cette variable peut être réalisée avec des macros prédéfinies très pratiques. Pour une description de ces macros, voyez plus loin dans le chapitre.
this_ptr	En utilisant cette variable, vous pouvez accéder à l'objet qui contient votre fonction, si elle fait partie d'un objet. Utilisez la fonction <code>getthis</code> pour en obtenir un pointeur.
return_value_used	Cette option indique si la valeur retournée par la fonction sera utilisée par le script. 0 indique que la valeur retournée n'est pas utilisée, et 1 indique que le script s'attend à recevoir une valeur. Cette option permet de s'assurer de l'utilisation correcte de la fonction, mais aussi de faire des optimisations si jamais retourner la valeur requiert des opérations compliquées (par exemple, voyez comment <code>array.c</code> utilise ce paramètre).
executor_globals	Cette variable pointe sur la configuration générale du moteur Zend. Vous la trouverez pratique lorsque vous créez de nouvelles variables, par exemple (nous verrons cela plus tard). Les variables globales de l'exécuteur peuvent aussi être introduites dans la fonction avec la macro <code>TSRMLS_FETCH()</code> .

8.8.4 Déclaration du bloc de fonctions Zend

Maintenant que vous avez déclaré les fonctions qui seront exportées vers PHP, il faut les introduire à Zend lui-même. Présenter la liste des fonctions est fait en utilisant un tableau de `zend_function_entry`. Ce tableau contient toutes les fonctions qui sont rendues disponibles à l'extérieur, avec le nom de la fonction qui doit apparaître dans PHP, et son nom de fonction C. En interne, `zend_function_entry` est déclarés comme dans [Déclaration interne de zend_function_entry](#).

Déclaration interne de zend_function_entry .	
<pre>typedef struct _zend_function_entry { char *fname; void (*handler)(INTERNAL_FUNCTION_PARAMETERS); unsigned char *func_arg_types; } zend_function_entry;</pre>	
Entrée	Description
fname	Le nom de la fonction, tel qu'il sera utilisable dans PHP (par exemple, <code>fopen</code> , <code>mysql_connect</code> , ou bien, dans notre exemple <code>first_module</code>).
handler	Un pointeur sur la fonction C, responsable du traitement de cette fonction. Par exemple, voyez la macro standard <code>INTERNAL_FUNCTION_PARAMETERS</code> présenté plus tôt.
func_arg_types	Vous permet de forcer certains paramètres, pour qu'ils soient passés par référence. Généralement, vous configurez cette option à NULL.

Dans l'exemple ci-dessus, la déclaration ressemble à :

```
zend_function_entry firstmod_functions[] =
{
    ZEND_FE(first_module, NULL)
    {NULL, NULL, NULL}
};
```

Vous pouvez voir que la dernière entrée de la liste est `{NULL, NULL, NULL}`. C'est un jalon, destiné à Zend, pour qu'il puisse reconnaître la fin de la liste de fonctions exportées.

Note

Vous **ne pouvez pas** utiliser les macros prédéfinies pour créer le jalon final, car cela conduira à ce qu'il se réfère à la fonction "NULL"!

La macro `ZEND_FE` (abréviation de 'Zend Function Entrée') est remplacé par une structure dans `zend_function_entry`. Notez que ces macros introduisent un plan de nommage spécial pour vos fonctions : toutes vos fonctions C seront préfixées avec `zif_`, ce qui signifie que `ZEND_FE(first_module)` se référera à la fonction C appelée `zif_first_module`. Si vous voulez mélanger des appels aux macros avec des entrées codées en dur (une très mauvaise pratique), gardez ça en tête.

Tip: Les erreurs de compilation qui font références à des fonctions nommés `zif_*` font référence aux fonctions définies avec `ZEND_FE`.

Macros de définition de fonctions montre une liste de macros que vous pouvez utiliser pour définir des fonctions.

Nom de la macro	Description
<code>ZEND_FE(name, arg_types)</code>	Définit une fonction du nom de <code>name</code> dans <code>zend_function_entry</code> . Requiert une fonction C correspondante. <code>arg_types</code> doit valoir <code>NULL</code> . Cette fonction utilise automatiquement la fonction C dont le nom est la fonction PHP, préfixé par <code>zif_</code> . Par exemple, <code>ZEND_FE("first_module", NULL)</code> introduit la fonction <code>first_module</code> dans PHP et la relie avec la fonction C <code>zif_first_module</code> . Utilisez cette macro en conjonction avec <code>ZEND_FUNCTION</code> .
<code>ZEND_NAMED_FE/php_name, name, arg_types)</code>	Définit une fonction qui sera disponible dans PHP sous le nom de <code>php_name</code> et la relie avec la fonction C de nom <code>name</code> . <code>arg_types</code> doit valoir <code>NULL</code> . Utilisez cette fonction si vous ne souhaitez pas utiliser le préfixage automatique de fonction, introduit par <code>ZEND_FE</code> . Utilisez cette macro en conjonction avec <code>ZEND_NAMED_FUNCTION</code> .
<code>ZEND_FALIAS(name, alias, arg_types)</code>	Définit un alias appelé <code>alias</code> pour la fonction <code>name</code> . <code>arg_types</code> doit valoir <code>NULL</code> . Cette macro ne demande pas de fonction C correspondante. Elle utilisera l'alias.
<code>PHP_FE(name, arg_types)</code>	Équivalent de <code>ZEND_FE</code> dans les vieilles API PHP.
<code>PHP_NAMED_FE(runtime_name, name, arg_types)</code>	Équivalent de <code>ZEND_NAMED_FE</code> dans les vieilles API PHP.

Note: Vous ne pouvez pas utiliser `ZEND_FE` en conjonction avec `PHP_FUNCTION`, ou `PHP_FE` en conjonction avec `ZEND_FUNCTION`. Cependant, il est parfaitement valide de mélanger des appels à `ZEND_FE` et `ZEND_FUNCTION` avec `PHP_FE` et `PHP_FUNCTION` lorsque vous restez dans le même jeu de macro pour chaque fonction déclarée. Mais mélanger les appels **n'est pas** recommandé; il est recommandé de n'utiliser que les macros `ZEND_*`.

8.8.5 Déclaration du bloc module de Zend

Ce bloc est stocké dans la structure `zend_module_entry` et contient toutes les informations nécessaires pour décrire le contenu du module à Zend. Vous pouvez voir la définition interne du module dans Déclaration interne de `zend_module_entry`.

Déclaration interne de `zend_module_entry`.


```
typedef struct _zend_module_entry zend_module_entry;

struct _zend_module_entry {
    unsigned short size;
    unsigned int zend_api;
    unsigned char zend_debug;
    unsigned char zts;
    char *name;
    zend_function_entry *functions;
    int (*module_startup_func)(INIT_FUNC_ARGS);
    int (*module_shutdown_func)(SHUTDOWN_FUNC_ARGS);
    int (*request_startup_func)(INIT_FUNC_ARGS);
    int (*request_shutdown_func)(SHUTDOWN_FUNC_ARGS);
    void (*info_func)(ZEND_MODULE_INFO_FUNC_ARGS);
    char *version;
    int (*global_startup_func)(void);
    int (*global_shutdown_func)(void);
};

[ Le reste de la structure n'est pas abordé ici ]
```

Entrée	Description
size , zend_api , zend_debug et zts	Généralement remplit par "STANDARD_MODULE_HEADER" , qui fournit la taille totale de zend_module_entry, le ZEND_MODULE_API_NO , si c'est une version de débogage ou une version normale (ZEND_DEBUG) et si ZTS est activé (USING_ZTS).
name	Contient le nom du module (par exemple, "File functions" , "Socket functions" , "Crypt" , etc...). Ce nom sera utilisé dans la phpinfo , à la section
functions	Pointe sur le bloc de fonction Zend, présenté juste ci-dessus.
module_startup_func	Cette fonction est appelée au moment de l'initialisation du module, et sert pour réaliser des initialisations uniques (telles que des allocations initiales de mémoire, etc...). Pour indiquer un échec durant cette initialisation, retournez FAILURE ; sinon, SUCCESS . Pour marquer ce champs comme inutilisé, utilisez la valeur NULL . Pour déclarer une fonction, utilisez la macro ZEND_MINIT .
module_shutdown_func	Cette fonction est appelée au moment de la terminaison du module, et sert pour réaliser des terminaisons uniques (telles que des désallocations finales de mémoire, etc...). Pour indiquer un échec durant cette terminaison, retournez FAILURE ; sinon, SUCCESS . Pour marquer ce champs comme inutilisé, utilisez la valeur NULL . Pour déclarer une fonction, utilisez la macro ZEND_MSHUTDOWN .
request_startup_func	Cette fonction est appelée au moment de l'initialisation des requêtes, et sert à réaliser des initialisations nécessaires à chaque requêtes Pour indiquer un échec durant cette initialisation, retournez FAILURE ; sinon, SUCCESS . Note: comme les modules dynamiques sont chargés uniquement durant le script, la fonction de démarrage de requête est appelée juste après celle de démarrage du module (en fait, l'initialisation de module et celle de requête interviennent au même instant). Pour marquer ce champs comme inutilisé, utilisez la valeur NULL . Pour déclarer une fonction, utilisez la macro ZEND_RINIT .
request_shutdown_func	Cette fonction est appelée au moment de la terminaison des requêtes, et sert à réaliser des terminaisons nécessaires à chaque requêtes Pour indiquer un échec durant cette terminaison, retournez FAILURE ; sinon, SUCCESS . Note: comme les modules dynamiques sont déchargés uniquement durant le script, la fonction de terminaison de requête est appelée juste après celle de terminaison du module (en fait, la terminaison de module et celle de requête interviennent au même instant). Pour marquer ce champs comme inutilisé, utilisez la valeur NULL . Pour déclarer une fonction, utilisez la macro ZEND_RSHUTDOWN .
info_func	Lorsque la fonction phpinfo est appelée dans un script, Zend passe en revue tous les modules, et appelle cette fonction. Chaque module a donc une chance pour fournir son propre grain de sel dans le résultat. Généralement, cela se fait en fournissant des informations d'environnement ou de statistiques. Pour marquer ce champs comme inutilisé, utilisez la valeur NULL . Pour déclarer une fonction, utilisez la macro ZEND_MINFO .
version	Le numéro de version du module. Vous pouvez utiliser la macro NO_VERSION_YET si vous ne voulez pas donner de version à votre module, mais nous recommandons d'ajouter toujours une chaîne de version ici. Une telle chaîne peut ressembler à ceci, en ordre chronologique : "2.5-dev" , "2.5RC1" , "2.5" ou "2.5pl3" .
Les autres éléments de la structure	Les autres éléments de la structures sont générés en interne, et peuvent être remplis automatiquement par la macro STANDARD_MODULE_PROPERTIES_EX . Ne leur donnez aucun valeur. Utilisez STANDARD_MODULE_PROPERTIES_EX uniquement si vous utilisez des fonctions de démarrage et de terminaison au niveau du module; sinon, utilisez directement STANDARD_MODULE_PROPERTIES .

Dans notre exemple, la structure est implémentée comme suit :

```
zend_module_entry firstmod_module_entry =
{
    STANDARD_MODULE_HEADER,
    "First Module",
    firstmod_functions,
    NULL, NULL, NULL, NULL,
    NO_VERSION_YET,
    STANDARD_MODULE_PROPERTIES,
};
```

C'est une structure qui est sûrement la plus facile et la plus minimaliste que vous puissiez utiliser. Le nom du module est choisi à ^{First Module} , puis vient la liste des fonctions référencées, puis la fonctions de démarrage et de terminaison, qui sont inutilisées.

Pour information, vous pouvez trouver la liste des macros utilisées dans les fonctions de démarrage et de terminaison dans la table [Macros utilisées pour déclarer les fonctions de démarrage et de terminaison](#) . Elles ne sont pas utilisées dans notre exemple de base, mais elles

seront exploitées plus tard. Il est recommandé d'utiliser ces fonctions pour déclarer vos fonctions de démarrage et terminaison, car elles réclament des arguments spéciaux (`INIT_FUNC_ARGS` et `SHUTDOWN_FUNC_ARGS`), qui sont automatiquement inclus dans la fonction, si vous utilisez les macros. Si vous déclarez vos fonctions manuellement, et que les développeurs PHP décident de changer la liste des arguments, vous devrez aussi changer vos propres sources pour rester compatible.

Macro	Description
<code>ZEND_MINIT(module)</code>	Déclare la fonction de démarrage de module. Le nom généré sera <code>zend_init_<module></code> (par exemple, <code>zend_init_first_module</code>). Utilisez-la en conjonction avec <code>ZEND_MINIT_FUNCTION</code> .
<code>ZEND_MSHUTDOWN(module)</code>	Déclare la fonction de terminaison de module. Le nom généré sera <code>zend_mshutdown_<module></code> (par exemple, <code>zend_mshutdown_first_module</code>). Utilisez-la en conjonction avec <code>ZEND_MSHUTDOWN_FUNCTION</code> .
<code>ZEND_RINIT(module)</code>	Déclare la fonction de démarrage de requête. Le nom généré sera <code>zend_rinit_<module></code> (par exemple, <code>zend_rinit_first_module</code>). Utilisez-la en conjonction avec <code>ZEND_RINIT_FUNCTION</code> .
<code>ZEND_RSHUTDOWN(module)</code>	Déclare la fonction de terminaison de requête. Le nom généré sera <code>zend_rshutdown_<module></code> (par exemple, <code>zend_rshutdown_first_module</code>). Utilisez-la en conjonction avec <code>ZEND_RSHUTDOWN_FUNCTION</code> .
<code>ZEND_MINFO(module)</code>	Déclare la fonction d'affichage des informations dans <code>phpinfo</code> . Le nom généré sera <code>zend_info_<module></code> (par exemple, <code>zend_info_first_module</code>). Utilisez-la en conjonction avec <code>ZEND_MINFO_FUNCTION</code> .

8.8.6 Création de la fonction `get_module`

Cette fonction est spéciale à tous les modules dynamiques. Etudiez donc sa création via la macro `ZEND_GET_MODULE` .

```
#if COMPILE_DL_FIRSTMOD
    ZEND_GET_MODULE(firstmod)
#endif
```

L'implémentation de la fonction est encadrée par des commandes de compilation conditionnelle. C'est nécessaire car la fonction `get_module` n'est utile que si votre module est compilé comme extension dynamique. En spécifiant la définition de `COMPILE_DL_FIRSTMOD` dans la commande de compilateur, (voir ci-dessus la présentation de la compilation sous forme d'extension dynamique), vous pouvez indiquer au module si vous allez le compiler comme extension dynamique, ou comme module statique. Si vous voulez un module statique, l'implémentation de la fonction `get_module` est simplement ignorée.

`get_module` est appelée par Zend au moment du chargement du module. Vous pouvez la considérer comme étant appelée par la fonction `dl` , dans le script PHP. Son objectif est de passer les informations du bloc de module à Zend, afin de lui présenter le contenu.

Si vous n'implémentez pas la fonction `get_module` dans vos modules dynamiques, Zend vous gratifiera d'un message d'erreur lorsque vous essayez d'y accéder.

8.8.7 Implémentation de toutes les fonctions exportées

L'implémentation de toutes les fonctions exportées est la dernière touche. Un exemple avec la fonction `first_module` ressemble à ceci :

```
ZEND_FUNCTION(first_module)
{
    long parameter;

    if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "l", &parameter) == FAILURE) {
        return;
    }

    RETURN_LONG(parameter);
}
```

La déclaration de la fonction est fait avec la macro `ZEND_FUNCTION`, qui fait écho à `ZEND_FE` dans la table de fonction, présentée plut tôt.

Après la déclaration, la vérification de l'appel, la lecture des arguments, la conversion des arguments et les valeurs retournées sont présentées plus loin dans ce chapitre.

8.8.8 Sommaire

C'est tout, simplement. Il n'y a plus rien d'autre de nécessaire pour implémenter un module PHP. Les modules inclus de PHP sont structurés de manière similaire aux modules dynamiques. Avec les informations de cette section, vous pouvez affronter la création d'un module PHP.

Maintenant, dans les sections suivantes, nous allons voir comment utiliser les rouages de PHP pour bâtir des extensions puissantes.

8.9 Gestion des arguments

Un des points les plus importants pour les extensions au langage, est la gestion des données qui sont passées en argument. La plus part des extensions sont construite pour traiter des données spécifiques, (ou requièrent des paramètres pour exécuter les actions spécifiques), et les arguments représentent le seul moyen de communication entre la fonction et son environnement. Bien sur, il y a aussi la possibilité d'échanger des données via les valeurs globales (ce qui sera discuté plus tard), mais cela doit être évité par tous les moyens, car c'est une mauvaise méthode.

PHP n'utilise aucune fonction de déclaration formelle. C'est pour cela que la syntaxe est toujours complètement dynamique et n'est jamais validée. La validation de la syntaxe est faite par le code du programmeur. Par exemple, il est possible d'appeler une fonction avec un argument, et un peu plus loin, la même fonction avec quatre arguments : les deux utilisations sont parfaitement correctes.

8.9.1 Déterminer le nombre d'arguments

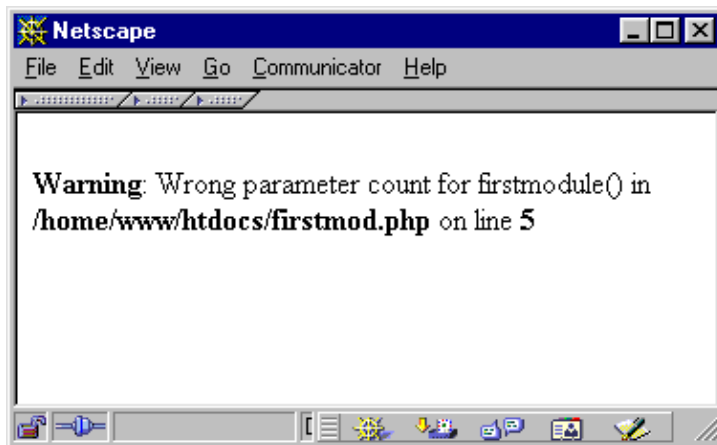
Comme PHP ne dispose pas de définitions de variables contraignantes, avec validation de la syntaxe, et comme PHP propose le nombre variable d'arguments, vous aurez parfois besoin de savoir combien d'arguments ont été passé à votre fonction. Dans cette situation, vous pouvez utiliser la macro `ZEND_NUM_ARGS`. Dans les anciennes versions de PHP, cette macro retournait le nombre d'arguments avec lequel la fonction avait été appelée, en se basant sur la table de hashage d'entrées (`ht`), qui est passée dans la liste `INTERNAL_FUNCTION_PARAMETERS`. Comme `ht` lui-même contient le nombre d'arguments passé à la fonction, `ZEND_NUM_ARGS` a été réduite à une macro simpliste (voyez sa définition dans le fichier `zend_API.h`). Mais c'est une bonne pratique que de l'utiliser, pour rester compatible avec des changements futurs dans l'interface. **Note:** L'ancien équivalent PHP de cette macro est `ARG_COUNT`.

Le code suivant vérifie le nombre d'argument correct :

```
if(ZEND_NUM_ARGS() != 2) WRONG_PARAM_COUNT;
```

Si la fonction n'est pas appelée avec deux arguments, elle se termine avec un message d'erreur. Le code ci-dessus utilise la macro `WRONG_PARAM_COUNT`, qui sert à générer un message d'erreur standard (voyez `WRONG_PARAM_COUNT` en action.).

WRONG_PARAM_COUNT en action.



Cette macro affiche un message d'erreur et retourne à l'appelant. Sa définition est disponible dans le fichier `zend_API.h` et ressemble à ceci :

```
ZEND_API void wrong_param_count(void);
#define WRONG_PARAM_COUNT { wrong_param_count(); return; }
```

Comme vous pouvez le voir, elle appelle une fonction interne appelée `wrong_param_count` qui est responsable de l'affichage du warning. Pour plus de détails sur la génération de messages personnalisés, voyez la section "Afficher des messages."

8.9.2 Lire les arguments

Note

Nouvelle API d'analyse des paramètres

Ce chapitre document la nouvelle API d'analyse des paramètres du moteur Zend, introduite par Andrei Zmievski. Elle a été introduite en développement entre PHP 4.0.6 et 4.1.0 .

Analyser les paramètres est une tâche très commune et qui devient laborieuse. Il serait aussi bien de disposer d'un système de validation standardisé, et des messages d'erreurs commun. Depuis PHP 4.1.0, il existe une nouvelle API qui comble ce manque. Elle simplifie grandement le processus de réception des paramètres, mais elle ne peut être utilisée pour des fonctions qui acceptent un nombre variable d'arguments. Mais comme la vaste majorité des fonctions a un nombre d'arguments fixe, cette API est recommandée comme nouveau standard.

Le prototype de la fonction d'analyse des paramètres ressemble à ceci :

```
int zend_parse_parameters(int num_args TSRMLS_DC, char *type_spec, ...);
```

Le premier argument de cette fonction est le nombre réel de paramètres de votre fonction, et la macro `ZEND_NUM_ARGS()` pourra être utilisée pour cela. Le deuxième paramètre doit toujours être la macro `TSRMLS_CC` . Le troisième argument est une chaîne qui spécifie le nombre et le type des arguments attendus par votre fonction, dans un format qui est proche de celui utilisé par printf. Et finalement, le reste des arguments sont des pointeurs vers les variables qui recevront les valeurs de ces paramètres.

`zend_parse_parameters` effectue des conversions de type lorsque c'est possible, ce qui fait que vous devriez recevoir les données dans le format que vous avez demandé. Tous les types de scalaires peuvent

être convertit dans un autre type de scalaire, mais les conversions entre les types complexes (tableaux, objets, ressources..) et les types scalaires ne sont pas permis.

Si les paramètres ont pu être obtenus et qu'il n'y a pas eu d'erreur durant le traitement, la fonction retourne `SUCCESS`, et sinon `FAILURE`. La fonction affichera des messages d'erreurs informatifs, si le nombre de paramètres ne correspond pas au nombre attendu, ou si les conversions de types ont échoué.

Voici quelques exemples de messages d'erreurs :

Warning - ini_get_all() requires at most 1 parameter, 2 given
Warning - wddx_deserialize() expects parameter 1 to be string, array given

Bien sûr, chaque message d'erreur est accompagné par le nom du fichier et le numéro de ligne.

Voici la liste complète des spécificateurs de type :

- `l` - long
- `d` - double
- `s` - chaîne de caractères (avec potentiellement le caractère null) et sa longueur
- `b` - booléen
- `r` - ressource, stockée en `zval*`
- `a` - tableau, stocké en `zval*`
- `o` - objet (de n'importe quelle classe), stocké en `zval*`
- `O` - objet (de la classe spécifiée par class entry), stocké en `zval*`
- `z` - un `zval*`

Les caractères suivants ont aussi des significations dans la chaîne de spécificateurs :

- `|` - indique que les paramètres restants sont optionnels. Les variables de stockage de ces paramètres doivent être initialisées à leur valeur par défaut (définie par l'extension) car elles seront ignorées par la fonction d'analyse si les paramètres ne sont pas fournis.
- `/` - la fonction d'analyse va appeler `separate_zval_if_not_ref` sur le paramètre suivant, pour fournir une copie du paramètre, à moins que cela ne soit une référence.
- `!` - Le paramètre qui suit peut être du type spécifié ou `NULL` (uniquement pour `a`, `o`, `O`, `r`, et `z`). Si `NULL` est passé par l'utilisateur, le pointeur de stockage utilisera `NULL`.

La meilleure méthode pour illustrer l'utilisation de ces fonctions est de voir un exemple :

```
/* Lit un long, une chaîne et sa longueur, et un zval. */
long l;
char *s;
int s_len;
zval *param;
if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC,
                          "lsz", &l, &s, &s_len, &param) == FAILURE) {
    return;
}

/* Lit un objet de classe spécifiée par my_ce, et un double optionnel. */
zval *obj;
double d = 0.5;
if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC,
                          "O|d", &obj, my_ce, &d) == FAILURE) {
    return;
}

/* Lit un objet ou null, et un tableau. Si null est passé comme objet,
   obj prendra la valeur NULL.*/
zval *obj;
zval *arr;
if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "O!a", &obj, &arr) == FAILURE) {
    return;
}

/* Lit un tableau. */
zval *arr;
if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "a/", &arr) == FAILURE) {
    return;
}
```

```
/* Lit seulement les trois premiers paramètres (utile pour les fonctions à nombre d'arguments variable) */
zval *z;
zend_bool b;
zval *r;
if (zend_parse_parameters(3, "zbr!", &z, &b, &r) == FAILURE) {
    return;
}
```

Notez que dans le dernier exemple, nous utilisons 3 comme nombre d'arguments reçus et non pas zend_num_args. Ce que cela nous permet est de recevoir le nombre minimum de paramètres que notre fonction attend. Bien sur, si vous souhaitez utiliser les autres paramètres, vous devrez utiliser la fonction zend_get_parameters_array_ex pour les obtenir.

La fonction d'analyse des paramètres dispose d'une version plus puissante, qui vous permet de passer des options qui contrôlent son comportement.

```
int zend_parse_parameters_ex(int flags, int num_args TSRMLS_DC, char *type_spec, ...);
```

Actuellement, la seule option utilisable est `ZEND_PARSE_PARAMS_QUIET`, qui commande à la fonction de ne pas afficher de messages d'erreur durant son exécution. Cela peut être utile pour des fonctions qui attendent différents ensembles d'arguments très différents, mais, d'un autre côté, vous devrez afficher vous même les erreurs.

Par exemple, voici comme vous pourriez attendre soit trois entiers, soit une chaîne de caractères.

```
long l1, l2, l3;
char *s;
if (zend_parse_parameters_ex(ZEND_PARSE_PARAMS_QUIET,
    ZEND_NUM_ARGS() TSRMLS_CC,
    "lll", &l1, &l2, &l3) == SUCCESS) {
    /* traiter les longs */
} else if (zend_parse_parameters_ex(ZEND_PARSE_PARAMS_QUIET,
    ZEND_NUM_ARGS(), "s", &s, &s_len) == SUCCESS) {
    /* traiter la chaîne */
} else {
    php_error(E_WARNING, "%s() prend soit trois longs, soit une chaîne comme arguments",
        get_active_function_name(TSRMLS_C));
    return;
}
```

Avec toutes les méthodes sus-citées de réception des paramètres, vous devriez avoir une bonne visibilité sur le processus. Pour d'autres exemples, fouillez donc le code source de PHP, et les extensions qui sont fournies avec PHP : ce sont d'excellentes illustrations.

8.9.3 Ancien méthode de lecture des arguments (obsolète)

Note

API obsolète de lecture des paramètres

Cette API est obsolète et a été remplacée par la nouvelle API Zend.

Après avoir vérifié le nombre d'arguments, vous devez accéder aux arguments eux-mêmes. Cela se fait avec l'aide de la fonction zend_get_parameters_ex :

```
zval **parameter;

if(zend_get_parameters_ex(1, &parameter) != SUCCESS)
    WRONG_PARAM_COUNT;
```

Tous les arguments sont stockés dans des enveloppes `zval`, qui doivent être déréférencés **deux fois**. Le code ci-dessus essaie de lire un argument, et le rend disponible via le pointeur `parameter`.

`zend_get_parameters_ex` accepte au moins deux arguments. Le premier argument est le nombre d'arguments à lire (qui doit correspondre au nombre d'arguments avec lequel la fonction a été appelé. C'est pour cela qu'il est important de vérifier la validité de la syntaxe d'appel). Le second argument (et tous les suivants) sont des pointeurs de pointeurs de pointeurs sur des `zval`. (abscons, n'est ce pas?) Tous ces pointeurs sont nécessaires car Zend fonctionne en interne avec des `**zval`; pour traiter une valeur locale `**zval` dans notre fonction, `zend_get_parameters_ex` requiert un pointeur.

La valeur retournée par `zend_get_parameters_ex` peut être `SUCCESS` ou `FAILURE`, qui indiquent (sans surprise) la réussite ou l'échec du traitement des arguments. Un échec est probablement relié à un mauvais nombre d'arguments, et dans ce cas, il est recommandé de terminer le script avec `WRONG_PARAM_COUNT`.

Pour lire plus d'un argument, vous pouvez utiliser le code suivant :

```
zval **param1, **param2, **param3, **param4;

if(zend_get_parameters_ex(4, &param1, &param2, &param3, &param4) != SUCCESS)
    WRONG_PARAM_COUNT;
```

`zend_get_parameters_ex` vérifie uniquement si vous essayer de lire trop de paramètres. Si la fonction est appelée avec 5 arguments, mais que vous n'en lisez que trois d'entre eux avec `zend_get_parameters_ex`, vous ne générerez pas une erreur, mais nous n'obtiendrez que les trois premiers paramètres. Les appels ultérieurs à `zend_get_parameters_ex` ne liront pas les arguments suivants, mais les mêmes arguments.

8.9.4 Traitement des nombres d'arguments variables et des paramètres optionnels

Si votre fonction doit accepter un nombre variable d'arguments, les exemples de code ci-dessus sont parfois sous-optimaux. Vous devez créer une ligne appelant `zend_get_parameters_ex` pour chaque nombre d'arguments possible, ce qui est souvent frustrant.

Dans ce cas, vous pouvez utiliser la fonction `zend_get_parameters_array_ex`, qui accepte le nombre d'arguments en paramètres, et un tableau dans lequel les stocker.

```
zval **parameter_array[4];

/* lit le nombre de paramètres passés */
argument_count = ZEND_NUM_ARGS();

/* vérification du nombre minimal d'arguments (2 arguments) */
/* et du nombre maximal d'arguments (4 arguments) */
if(argument_count < 2 || argument_count > 5)
    WRONG_PARAM_COUNT;

/* Le nombre d'arguments est valide, nous pouvons les lire */
if(zend_get_parameters_array_ex(argument_count, parameter_array) != SUCCESS)
    WRONG_PARAM_COUNT;
```

Tout d'abord, le nombre d'arguments est testé pour s'assurer qu'il est valide. Après cela, `zend_get_parameters_array_ex` est appelée pour remplir la variable `parameter_array` avec un pointeur valide vers les valeurs des arguments.

Une utilisation très habile de cette technique est accessible dans le code de `fsocketopen`, dans le fichier `ext/standard/fsock.c`, disponible dans [Implémentation en PHP du nombre variable d'arguments avec fsocketopen\(\)](#). Ne paniquez pas si vous ne connaissez pas toutes les fonctions dans ce source :

nous y viendrons progressivement.

Implémentation en PHP du nombre variable d'arguments avec `fsockopen()`.

```

    pval **args[5];
    int *sock=emalloc(sizeof(int));
    int *sockp;
    int arg_count=ARG_COUNT(ht);
    int socketd = -1;
    unsigned char udp = 0;
    struct timeval timeout = { 60, 0 };
    unsigned short portno;
    unsigned long conv;
    char *key = NULL;
    FLS_FETCH();

    if (arg_count > 5 || arg_count < 2 || zend_get_parameters_array_ex(arg_count,args)==FAILURE) {
        CLOSE_SOCK(1);
        WRONG_PARAM_COUNT;
    }

    switch(arg_count) {
        case 5:
            convert_to_double_ex(args[4]);
            conv = (unsigned long) (Z_DVAL_P(args[4]) * 1000000.0);
            timeout.tv_sec = conv / 1000000;
            timeout.tv_usec = conv % 1000000;
            /* fall-through */
        case 4:
            if (!PZVAL_IS_REF(*args[3])) {
                php_error(E_WARNING,"error string argument to fsockopen not passed by reference");
            }
            pval_copy_constructor(*args[3]);
            ZVAL_EMPTY_STRING(*args[3]);
            /* fall-through */
        case 3:
            if (!PZVAL_IS_REF(*args[2])) {
                php_error(E_WARNING,"error argument to fsockopen not passed by reference");
                return;
            }
            ZVAL_LONG(*args[2], 0);
            break;
    }

    convert_to_string_ex(args[0]);
    convert_to_long_ex(args[1]);
    portno = (unsigned short) Z_LVAL_P(args[1]);

    key = emalloc(Z_STRLEN_P(args[0]) + 10);

```

`fsockopen` accepte deux, trois, quatre ou cinq paramètres. Après la déclaration obligatoire des variables, la fonction vérifie que le nombre d'argument est valide. Puis, il utilise le mécanisme de 'fall through' du `switch()` pour traiter ces arguments. `switch()` commence avec le nombre maximal d'arguments, c'est à dire cinq. Puis, il traite le cas de quatres arguments passés, puis de trois, en omettant simplement d'utiliser `break`. Une fois que tous les cas sont passés en revue, `switch()` est terminé, et il traite le nombre minimum d'arguments avec lequel la fonction doit être invoquée, c'est à dire deux.

Ce type de traitement a plusieurs étages, similaire à un escalier, permet de gérer proprement et facilement les fonctions à nombre d'arguments variables.

8.9.5 Accéder aux arguments

Pour accéder aux arguments, il est nécessaire que le type de chaque argument soit clairement défini. Encore une fois, la nature extrêmement dynamique de PHP introduit des couaks. Comme PHP ne fait aucune validation de type, il est possible que le script appelant passe n'importe quel type de données à vos fonctions. Si vous attendez un entier, rien n'empêche le programmeur de vous passer un tableau, et vice-versa. PHP ne s'en préoccupe pas.

Pour résoudre de problème, vous devez utiliser un jeu d'API pour forcer la conversion de type de chaque argument qui est passé (voyez [Fonction de conversion](#)).

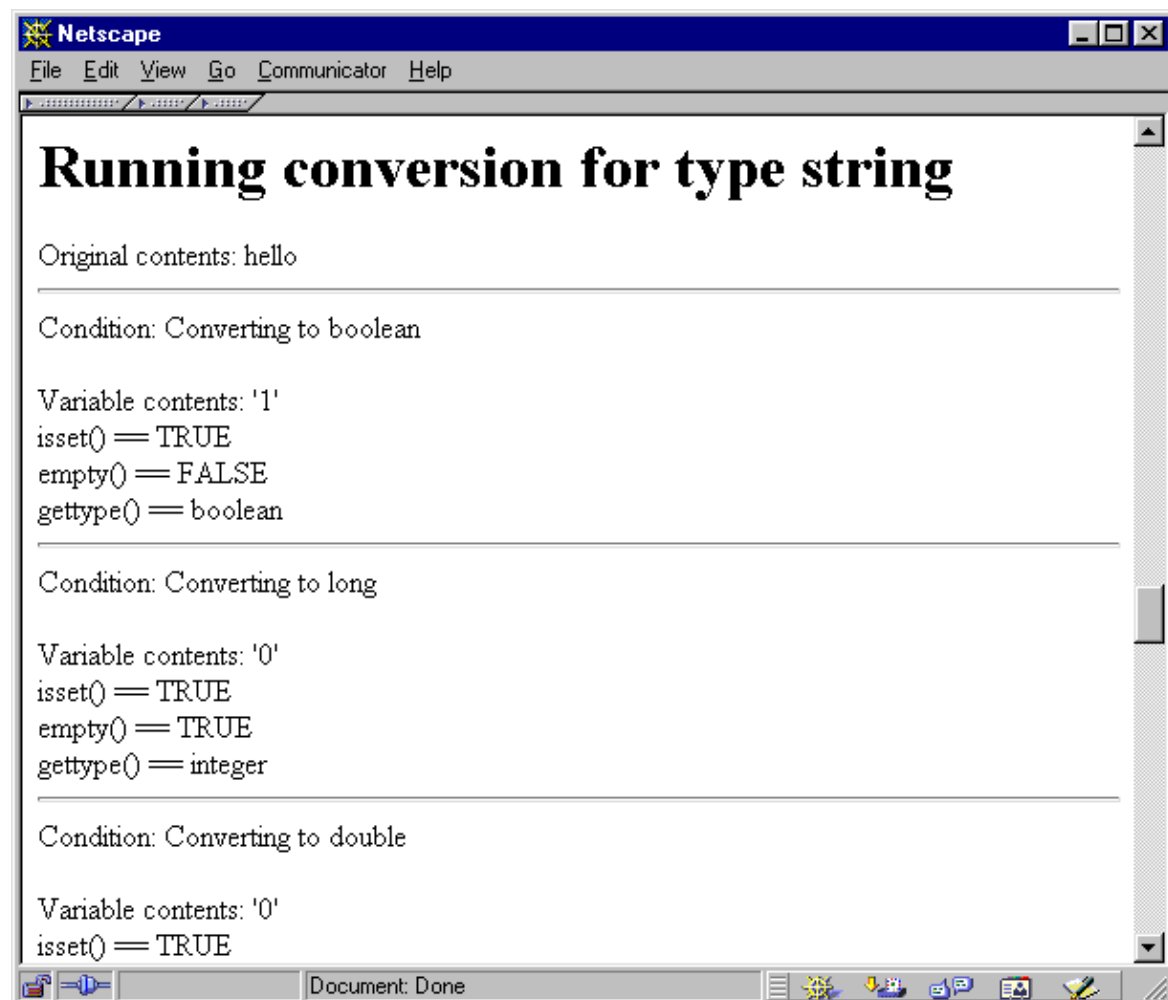
Note: Toutes les fonctions de conversion attendent une enveloppe `**zval` comme paramètre.

Fonction	Description
convert_to_boolean_ex	Force la conversion en type booléen. Les valeurs booléennes sont laissées intactes. Les longs, doubles et chaînes contenant 0 ainsi que les valeurs NULL sont converties en well as NULL values will result in Boolean 0 (FALSE). Arrays and objects are converted based on the number of entries or properties, respectively, that they have. Empty arrays and objects are converted to FALSE; otherwise, to TRUE. All other values result in a Boolean 1 (TRUE).
convert_to_long_ex	Force la conversion en type long, le type entier par défaut. Les valeurs NULL, booléennes ressources, et bien sur les autres longs sont laissés intactes. Les doubles sont tronqués. Les chaînes contenant un entier sont converties en leur équivalent numérique, sinon 0 . Les tables et objets sont convertis en 0 si ils sont vides, 1 sinon.
convert_to_double_ex	Force la conversion en type double, le type de nombres à virgules flottantes par défaut. Les valeurs NULL, booléennes, ressources, longs et bien sur les autres doubles sont laissés intactes. Les chaînes contenant un entier sont converties en leur équivalent numérique, sinon 0.0 . Les tables et objets sont convertis en 0.0 si ils sont vides, 1.0 sinon.
convert_to_string_ex	Force la conversion en type chaîne de caractères. Les chaînes sont laissées intactes. La valeur NULL est convertie en chaîne vide. Les booléens contenant TRUE sont convertis en "1" , et sinon, en chaîne vide. Les longs et doubles sont convertis en leur représentation littérales. Les tableaux sont convertis en la chaîne "Array" et les objets sont convertis en la chaîne "Object" .
convert_to_array_ex(value)	Force la conversion en type tableau. Les tableaux sont laissés intactes. Les objets sont convertis en un tableau en assignant les valeurs des membres dans un élément de tableau avec le nom du membre comme index. La valeur NULL est transformée en tableau vide. Toutes les autres valeurs sont converties en un tableau qui contient un élément, la valeur elle-même, à l'index 0 .
convert_to_object_ex(value)	Force la conversion en type objet. Les objets sont laissés intactes. La valeur NULL est convertie en une objet vide. Les tableaux sont convertis en objets, dont les membres sont les anciens index du tableau, et la valeur du membre est la valeur correspondante dans le tableau. Tous les autres types sont convertis en un objet ayant le membre scalar , qui contient la valeur convertie.
convert_to_null_ex(value)	Convertit la valeur en NULL, qui signifie vide.

Note

Vous pouvez trouver une démonstration de ces conversions dans le fichier `cross_conversion.php` sur le CD ROM d'accompagnement. [Conversions en PHP](#), illustre l'affichage.

Conversions en PHP.



En utilisant ces fonctions sur vos arguments, vous vous assurez du type des données qui vous sont passées. Si le type de la valeur fournie ne correspond pas à celui que vous souhaitez, PHP force la valeur à une valeur de remplacement classique (chaînes vides, tableaux ou objets vides, ⁰ pour les nombres, `FALSE` pour les booléens), afin d'avoir toujours un état défini.

Voici un extrait du module présenté ci-dessus, qui effectue une conversion :

```
zval **parameter;

if((ZEND_NUM_ARGS() != 1) || (zend_get_parameters_ex(1, &parameter) != SUCCESS))
{
    WRONG_PARAM_COUNT;
}

convert_to_long_ex(parameter);

RETURN_LONG(Z_LVAL_P(parameter));
```

Après avoir lu le pointeur du paramètre, la valeur est convertie en long (un entier) qui sera aussi la valeur retournée par la fonction. Comprendre l'accès aux valeurs nécessite une présentation du type `zval`, dont les définitions sont présentées Définition du type zval.

Définition du type zval .	
<pre>typedef pval zval; typedef struct _zval_struct zval; typedef union _zvalue_value { long lval; /* valeur long */ double dval; /* valeur double */ struct { char *val; int len; } str; HashTable *ht; /* table de hash */ struct { zend_class_entry *ce; HashTable *properties; } obj; } zvalue_value; struct _zval_struct { /* Information sur la variable */ zvalue_value value; /* valeur */ unsigned char type; /* type actif */ unsigned char is_ref; short refcount; };</pre>	

En fait, `pval` (défini dans `php.h`) est simplement un alias de `zval` (défini dans `zend.h`), qui lui-même fait référence à `_zval_struct`. C'est une structure très intéressante. `_zval_struct` est la structure "principale" qui contient la valeur, son type et des informations de référence. La sous-structure `zvalue_value` est une union qui contient la valeur de la variable. Suivant le type de la variable, vous pouvez accéder à différents membres de l'union. Pour une description complète des deux structures, voyez Structure Zend zval, Structure Zend zvalue_value et Constantes de type Zend Variable.

Entrée	Description
value	Union contenant la valeur de la variable. Voyez <u>Structure Zend zvalue_value</u> pour plus de détails.
type	Le type de la variable. Pour une liste des types, voyez <u>Constantes de type Zend Variable</u> .
is_ref	0 signifie que cette variable n'est pas une référence. 1 signifie que cette variable est une référence à une autre variable.
refcount	Le nombre de référence qui existe pour cette variable. Pour chaque nouvelle référence à la valeur lue dans cette variable, ce compteur est incrémenté de 1. Pour chaque référence perdue, le compteur est décrémenté de 1. Lors que compteur de référence atteint 0, aucune référence n'existe, et la valeur est alors automatiquement détruite.

Entrée	Description
lval	Utilisez cette propriété si la variable est de type <code>IS_LONG</code> , <code>IS_BOOLEAN</code> , ou <code>IS_RESOURCE</code> .
dval	Utilisez cette propriété si la variable est de type <code>IS_DOUBLE</code> .
str	Utilisez cette propriété si la variable est de type <code>IS_STRING</code> . Le membre <code>len</code> contient la taille de la chaîne. Le membre <code>val</code> pointe sur la chaîne elle-même. Zend utilise des chaînes de type C; ainsi la taille de la chaîne contient un caractère <code>0x00</code> final.
ht	Cette entrée pointe sur la table de hashage si la variable est un tableau.

obj	Utilisez cette propriété si la variable est de type IS_OBJECT .

Constante	Description
IS_NULL	Indique un type NULL.
IS_LONG	Indique un type long (entier).
IS_DOUBLE	Indique un type double (floating point).
IS_STRING	Indique un type chaîne de caractères.
IS_ARRAY	Indique un type tableau.
IS_OBJECT	Indique un type objet.
IS_BOOL	Indique un type booléen.
IS_RESOURCE	Indique un type ressource (pour plus de détails sur les ressources, voyez ci-dessous).
IS_CONSTANT	Indique un type constante.

Pour accéder à la valeur d'un long, vous pouvez y accéder avec `zval.value.lval`, pour un double, vous pouvez y accéder avec `zval.value.dval`, et ainsi de suite... Comme toutes les valeurs sont stockées dans une union, accéder à un membre qui ne coresspond pas au type en cours retourne des valeurs sans aucun sens.

Accéder aux tableaux et objets est un peu plus compliqué, et sera discuté plus tard.

8.9.6 Traiter les arguments passés par référence

Si votre fonction accepte des arguments passés par référence, et que vous souhaitez les modifier, vous devez prendre quelques précautions.

Ce que nous n'avons pas encore dit, c'est que dans les circonstances qui vous ont été présentées, vous n'avez aucun droit d'écriture dans les enveloppes `zval` qui vous ont été passé. Bien entendu, vous pouvez modifier toutes ces enveloppes, mais vous ne devez changer aucun `zval` qui fasse référence aux données interne de Zend.

Nous avons uniquement présenté les fonctions `*_ex` de l'API. Vous pouvez avoir remarqué que cette API s'appelle `zend_get_parameters_ex` au lieu de `zend_get_parameters`, `convert to long_ex` au lieu de `convert to long`, etc... Les fonctions `*_ex` représentent la nouvelle API Zend étendue. Elles apporte une petite amélioration en terme de vitesse d'exécution, mais en échange, elles ne fournissent qu'un accès en lecture aux arguments.

Comment Zend travaille en interne avec des références, ces différentes variables pourrait faire référence à la même valeur. L'accès en écriture aux variables `zval` requiert cette enveloppe pour contenir une valeur isolée, ce qui signifie que la valeur n'est pas référencée par une autre enveloppe. Si une enveloppe `zval` était référencée et que vous modifiez la valeur `zval`, vous changeriez automatiquement le contenu de toutes les enveloppes qui y font référence (car elles ne font que pointer sur une valeur que vous modifiez, et qui donc, modifie leur valeur en même temps).

`zend_get_parameters_ex` ne se préoccupe pas de la situation, mais retourne simplement un pointeur sur l'enveloppe `zval`, que cette dernière soit une référence ou pas. La fonction correspondante de l'API traditionnelle `zend_get_parameters`, vérifie immédiatement les valeurs référencées. Si elle trouve une référence, elle crée une nouvelle enveloppe `zval` isolée, copie les données référencées dans le nouvel espace allouée et retourne un pointeur sur la nouvelle valeur créée.

Cette action est appelée **séparation zval** (ou séparation pval). Mais comme l'API `*_ex` ne fait pas cette séparation, elle est considérablement plus rapide, mais au prix de l'accès en écriture.

Pour modifier les paramètres, il vous faut cependant un accès en écriture. Zend gère cette situation d'un manière spéciale : lorsqu'un paramètre est passé à une fonction par référence, il pratique une séparation zval automatique. Cela signifie qu'à chaque fois que vous appelez une fonction PHP avec un argument passé par référence, Zend va automatiquement s'assurer que `$parameter` est bien

passé comme une valeur isolée, pour le rendre accessible en écriture.

```
ma_function(&$parameter);
```

Mais ce **n'est pas le cas** avec les paramètres réguliers. Tous les autres paramètres qui ne sont pas passés par référence sont dans un état de lecture seule.

Cela vous impose de vous assurer que vous voulez vraiment travailler avec une référence : sinon, vous risquez de générer des effets secondaires nuisibles. Pour vérifier qu'un paramètre est passé par référence, vous pouvez utiliser la macro `PZVAL_IS_REF`. Cette macro accepte un `zval*` pour vérifier si c'est une référence ou pas. Des exemples sont fournis dans Test des paramètres passés par référence.

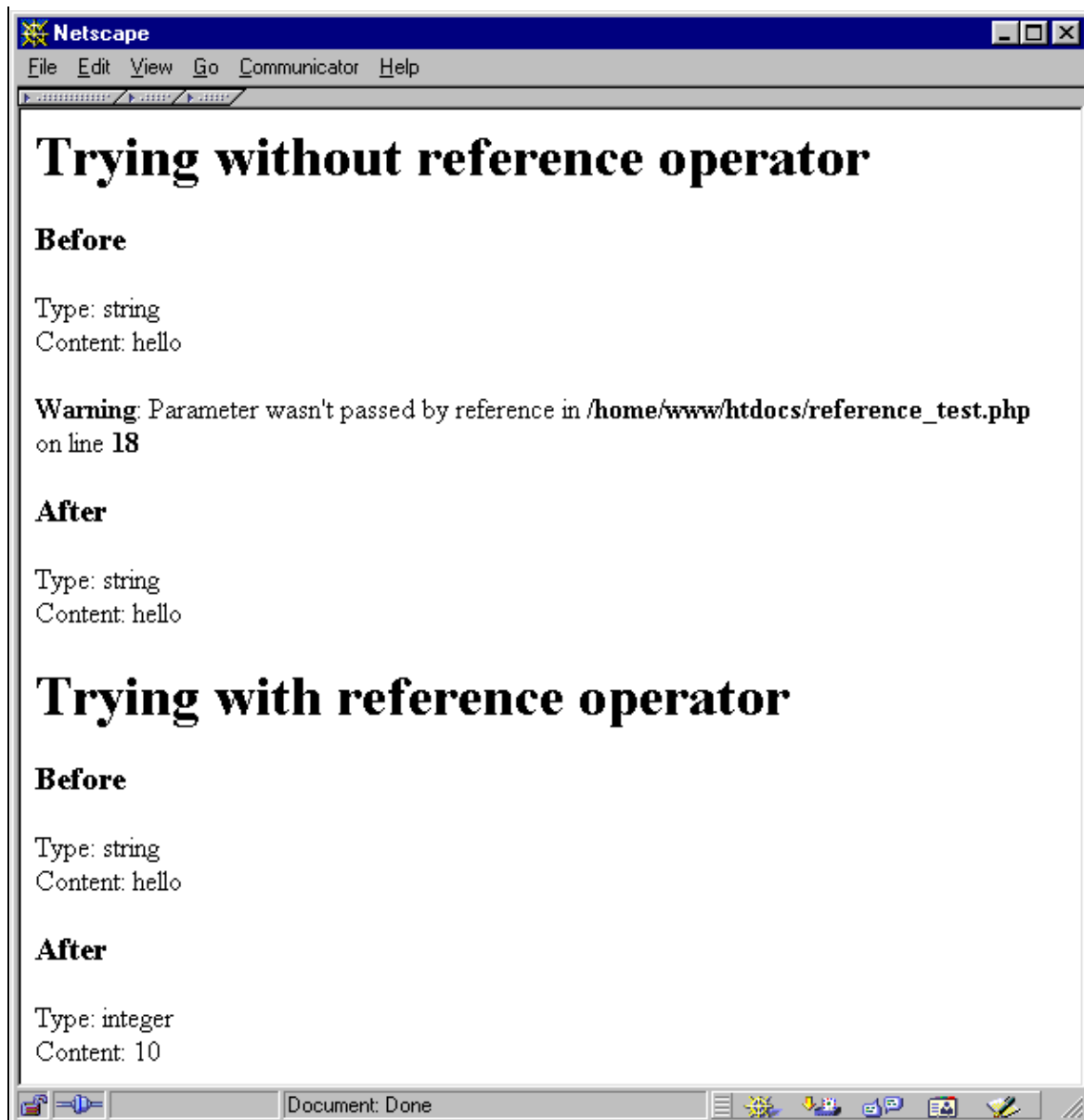
Test des paramètres passés par référence.

```
zval *parameter;

if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "z", &parameter) == FAILURE)
    return;

/* vérifie si le paramètre est passé par référence. */
if (!PZVAL_IS_REF(*parameter)) {
{
    zend_error(E_WARNING, "Le paramètre a été passé par référence");
    RETURN_NULL();
}

/* convertit le paramètre */
ZVAL_LONG(*parameter, 10);
```



8.9.7 Assurer les autres paramètres contre l'écriture

Vous pouvez vous trouver dans une situation où vous devez avoir accès à un paramètre qui a été lu par `zend_get_parameters_ex` mais pas passé par référence. Dans ce cas, vous pouvez utiliser la macro `SEPARATE_ZVAL`, qui effectue une séparation zval sur l'enveloppe fournie. La nouvelle enveloppe générée est détachée de sa représentation interne, et n'est valide que localement, ce qui signifie qu'elle peut être modifiée ou détruite dans l'implication au niveau global.

```
zval **parameter;

/* Lit les paramètres */
zend_get_parameters_ex(1, &parameter);

/* A ce moment, <parameter> est toujours connecté */
/* aux buffers internes de Zend */

/* Rend <parameter> accessible en écriture */
SEPARATE_ZVAL(parameter);

/* Maintenant, nous pouvons modifier sans problème <parameter> */
/* sans faire de modifications globales */
```

`SEPARATE_ZVAL` utilise `emalloc` pour allouer une nouvelle enveloppe `zval`, ce qui signifie que même si vous ne désallouer par la mémoire vous-même, elle sera nettoyé dynamiquement lors de la terminaison du script. Cependant, appeler souvent cette macro sans libérer votre RAM va fragmenter votre mémoire vive.

Note: vous pouvez contourner facilement le manque d'accès dans la l'API traditionnelle (avec la fonction `zend_get_parameters`, etc...), mais cette API est obsolète et n'est pas détaillée dans ce chapitre.

8.10 Créer des variables

Lorsque vous échangez des données entre votre extension et PHP, un des points les plus importants est la création de variables. Cette section vous montre comment traiter les types de variables que PHP supte.

8.10.1 Présentation

Pour créer de nouvelles variables qui seront accessibles hors de vos fonctions, et utilisées par le script, vous devez allouer une enveloppe `zval`, lui assigner une valeur significative, et introduire la variable dans les tables internes de symboles de Zend. Ce processus est commun à toutes les créations de variables :

```
zval *new_variable;

/* Alloue et initialise la nouvelle enveloppe */
MAKE_STD_ZVAL(new_variable);

/* Choisit le type et le contenu de la variable : reportez vous aux sections suivantes */

/* Introduit cette variable sous le nom de "new_variable_name" dans la table des symboles */
ZEND_SET_SYMBOL(EG(active_symbol_table), "new_variable_name", new_variable);

/* La variable est désormais accessible au script, sous le nom de $new_variable_name */
```

La macro `MAKE_STD_ZVAL` alloue une nouvelle enveloppe `zval` en utilisant la macro `ALLOC_ZVAL` puis, initialise cette variable avec la macro `INIT_ZVAL`. Comme implémenté dans Zend au moment de l'écriture de ce chapitre, **initialisation** signifie configurer le compteur de références à ¹ et remettre à zéro l'option `is_ref`, mais ce processus pourra être amélioré plus tard. C'est pour cela que c'est une bonne idée que d'utiliser la macro `MAKE_STD_ZVAL` plutôt que `ALLOC_ZVAL`. Si vous voulez optimiser la vitesse de votre extension (et que vous n'avez pas à initialiser explicitement votre enveloppe `zval` ici), vous pouvez utiliser `ALLOC_ZVAL`, mais ce n'est pas recommandé, car cette méthode n'assure pas l'intégrité de vos données.

`ZEND_SET_SYMBOL` prend en charge l'introduction de votre variable dans la table des symboles de Zend. Cette macro vérifie si une variable de même nom n'existe pas déjà, puis convertit la nouvelle variable en une référence si c'est le cas (avec désallocation automatique de l'ancienne enveloppe `zval`). C'est la meilleure méthode si la vitesse n'est pas cruciale, et que vous souhaitez garder un niveau d'utilisation de la mémoire bas.

Notez que `ZEND_SET_SYMBOL` fait usage des variables globales d'exécutions via la macro `EG`. En spécifiant `EG(active_symbol_table)`, vous gagnez l'accès à la table des symboles actives, qui gère l'environnement local. L'environnement local peut varier de fonction en fonction.

Si vous devez optimiser la vitesse de votre extension, et que vous ne vous souciez pas de l'utilisation optimale de la mémoire, vous pouvez omettre la vérification d'existence d'une variable ayant la même valeur, et forcer l'introduction du symbole dans la table, en utilisant la fonction

zend_hash_update :

```
zval *new_variable;

/* Alloue et initialise la nouvelle enveloppe */
MAKE_STD_ZVAL(new_variable);

/* Choisit le type et le contenu de la variable : reportez vous aux sections suivantes */

/* Introduit cette variable sous le nom de "new_variable_name" dans la table des symboles */
zend_hash_update(
    EG(active_symbol_table),
    "new_variable_name",
    strlen("new_variable_name") + 1,
    &new_variable,
    sizeof(zval *),
    NULL
);
```

C'est la méthode qui est utilisée dans la plus part des modules.

Les variables générées avec le code ci-dessus seront placées dans l'environnement d'exécution local : elles résident dans l'environnement de la fonction qui les a appelées. Pour créer des variables dans l'environnement global, utilisez la même méthode, mais faites références à une autre table de symboles :

```
zval *new_variable;

/* Alloue et initialise la nouvelle enveloppe */
MAKE_STD_ZVAL(new_variable);

/* Choisit le type et le contenu de la variable : reportez vous aux sections suivantes */

/* Introduit cette variable sous le nom de "new_variable_name" dans la table des symboles globale */
ZEND_SET_SYMBOL(&EG(symbol_table), "new_variable_name", new_variable);
```

La macro `ZEND_SET_SYMBOL` est ici appelée avec une référence sur la table des symboles globales, en utilisant la macro `EG(symbol_table)`.

Note : La variable `active_symbol_table` est un pointeur, mais `symbol_table` n'en est pas un. C'est pour cela que vous devez utiliser `EG(active_symbol_table)` et `&EG(symbol_table)` comme paramètre de `ZEND_SET_SYMBOL` : elle requiert un pointeur.

Similairement, pour améliorer l'efficacité de votre code, vous pouvez coder en dur la modification de la table des symboles globale :

```
zval *new_variable;

/* Alloue et initialise la nouvelle enveloppe */
MAKE_STD_ZVAL(new_variable);

/* Choisit le type et le contenu de la variable : reportez vous aux sections suivantes */

/* Introduit cette variable sous le nom de "new_variable_name" dans la table des symboles globale */
zend_hash_update(
    &EG(symbol_table),
    "new_variable_name",
    strlen("new_variable_name") + 1,
    &new_variable,
    sizeof(zval *),
    NULL
);
```

Création de variables dans différents environnements montre un exemple de sources qui crée deux variables : `local_variable` dans l'environnement local, et `global_variable` dans l'environnement global (voir Figure 9.7).

Note: Vous pouvez remarquer que la variable globale n'est pas réellement accessible depuis la fonction. C'est parcequ'elle n'est pas importée dans l'environnement local avec `global $global_variable;`, dans le source PHP.

Création de variables dans différents environnements

```

ZEND_FUNCTION(variable_creation)
{
    zval *new_var1, *new_var2;

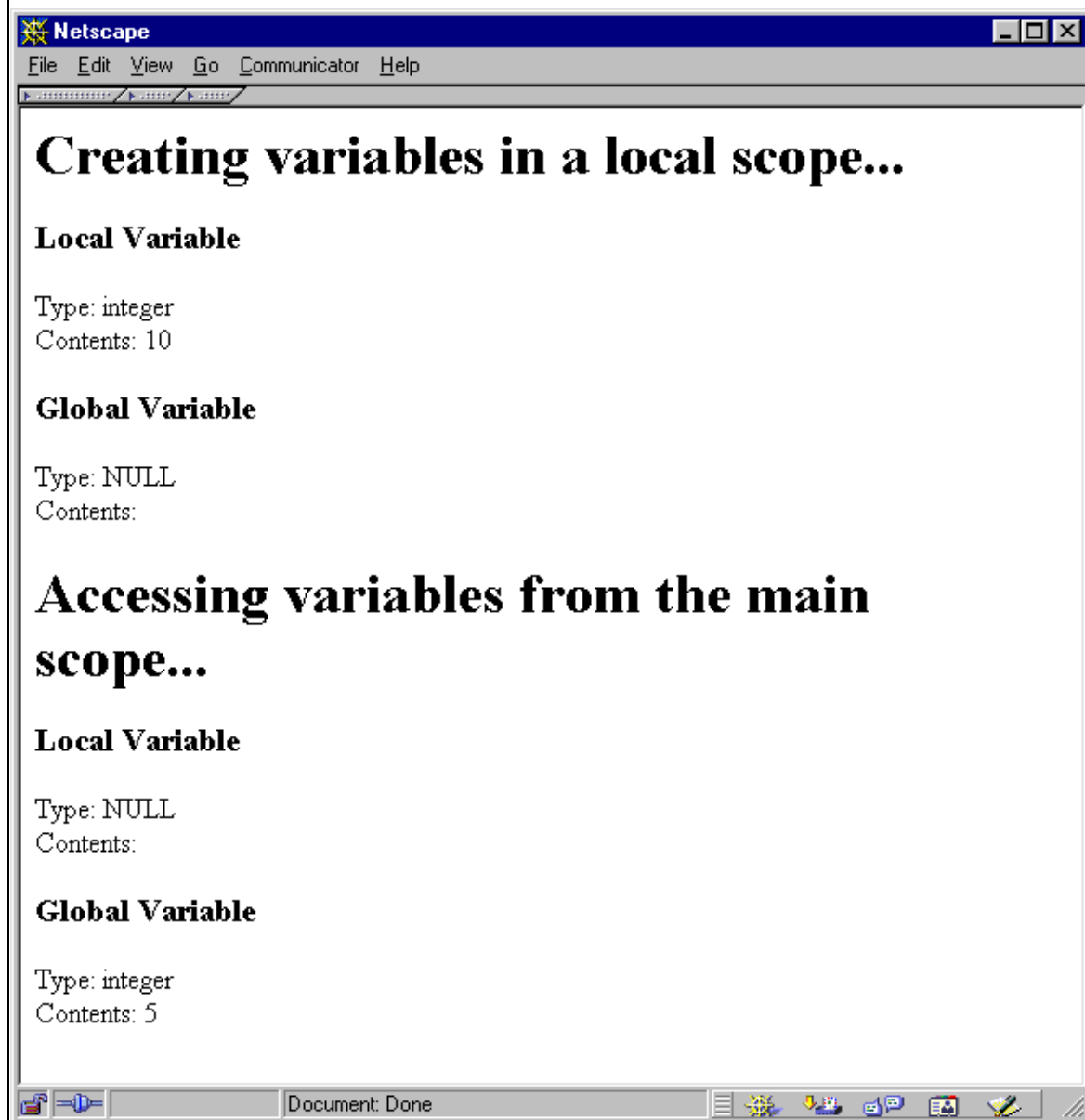
    MAKE_STD_ZVAL(new_var1);
    MAKE_STD_ZVAL(new_var2);

    ZVAL_LONG(new_var1, 10);
    ZVAL_LONG(new_var2, 5);

    ZEND_SET_SYMBOL(EG(active_symbol_table), "local_variable", new_var1);
    ZEND_SET_SYMBOL(&EG(symbol_table), "global_variable", new_var2);

    RETURN_NULL();
}

```



8.10.2 Entiers (Longs)

Passons maintenant à l'assignation de données aux variables, en commençant par les entiers. Les entiers (appelés aussi les longs, en anglais), sont très simples à stocker. En observant la structure des enveloppes `zval`, vous trouverez l'élément `zval.value`, présenté plus tôt dans ce chapitre. Le membre de `zval.value` qui contient cette valeur est `lval`. Le type correspondant à la valeur est `IS_LONG` (voir [Creation of a long.](#)).

Creation of a long.

```
zval *new_long;

MAKE_STD_ZVAL(new_long);

new_long->type = IS_LONG;
new_long->value.lval = 10;
```

Alternativement, vous pouvez utiliser la macro `ZVAL_LONG` :

```
zval *new_long;

MAKE_STD_ZVAL(new_long);
ZVAL_LONG(new_long, 10);
```

8.10.3 Les nombres à virgules flottantes (Doubles, Floats)

Les nombres à virgules flottantes (appelés aussi les doubles et les floats) sont aussi simples à manipuler que les entiers, car leur valeur est directement contenue dans l'union. Le membre de `zval.value` qui contient cette valeur est `dval`. Le type correspondant est `IS_DOUBLE`.

```
zval *new_double;

MAKE_STD_ZVAL(new_double);

new_double->type = IS_DOUBLE;
new_double->value.dval = 3.45;
```

Alternativement, vous pouvez utiliser la macro `ZVAL_DOUBLE` :

```
zval *new_double;

MAKE_STD_ZVAL(new_double);
ZVAL_DOUBLE(new_double, 3.45);
```

8.10.4 Chaînes de caractères (strings)

Les chaînes de caractères (appelées aussi des strings) demandent quelques efforts supplémentaires. Comme mentionné auparavant, toutes les chaînes de caractères seront associées avec une structure de données interne à Zend qui a besoin d'être allouée avec les fonctions de gestion de mémoire de Zend. Le référencement de données statiques, et celui de données référencées avec des routines standards ne sont pas autorisés. Pour assigner une chaîne, vous devez utiliser le membre `str` dans le champs `zval.value`. Le type correspondant est `IS_STRING`.

```
zval *new_string;
char *string_contents = "Ceci est une nouvelle chaîne";
```

```
MAKE_STD_ZVAL(new_string);

new_string->type = IS_STRING;
new_string->value.str.len = strlen(string_contents);
new_string->value.str.val = estrdup(string_contents);
```

Notez bien l'utilisation `estrdup`. Bien sur, vous pouvez aussi utiliser la macro prédéfinie `ZVAL_STRING` :

```
zval *new_string;
char *string_contents = "Ceci est une nouvelle chaîne";

MAKE_STD_ZVAL(new_string);
ZVAL_STRING(new_string, string_contents, 1);
```

`ZVAL_STRING` accepte un troisième argument qui indique si la chaîne fournie doit être dupliquée (en utilisant `estrdup`). En donnant la valeur de ¹ à ce paramètre, la chaîne sera dupliquée. ⁰ fera simplement usage du pointeur fourni pour référer le contenu. C'est très pratique si vous souhaitez créer une nouvelle variable qui fait référence à une chaîne qui existe déjà en mémoire Zend.

Si vous voulez tronquer la chaîne à une taille, ou que vous connaissez déjà la taille de la chaîne, utilisez donc la macro `ZVAL_STRINGL(zval, string, length, duplicate)`, qui accepte une taille de chaîne explicite. Cette macro est bien plus rapide que `ZVAL_STRING` tout en supportant les chaînes binaires.

Pour créer une chaîne vide, donner à votre chaîne la taille de ⁰ et utilisez `empty_string` comme contenu :

```
new_string->type = IS_STRING;
new_string->value.str.len = 0;
new_string->value.str.val = empty_string;
```

Bien sur, il y a aussi une macro pour cela : `ZVAL_EMPTY_STRING`).

```
MAKE_STD_ZVAL(new_string);
ZVAL_EMPTY_STRING(new_string);
```

8.10.5 Booléens

Les booléens sont créés comme des entiers, mais ils ont le type `IS_BOOL`. Les valeurs autorisées dans `lval` sont ⁰ et ¹ :

```
zval *new_bool;

MAKE_STD_ZVAL(new_bool);

new_bool->type = IS_BOOL;
new_bool->value.lval = 1;
```

Les macros correspondantes pour ces types sont `ZVAL_BOOL` (en spécifiant la valeur), ainsi que `ZVAL_TRUE` et `ZVAL_FALSE` (qui donne explicitement la valeur de `TRUE` et `FALSE`, respectivement).

8.10.6 Tableaux

Les tableaux sont stockés dans les tables de hashage internes de Zend, qui sont accessibles via les fonctions `zend_hash_*`. Pour chaque tableau que vous voulez créer, vous devez créer une nouvelle table de hashage, qui sera stockée dans le membre `ht` du champs `zval.value`.

Il y a tout un jeu de fonctions pour simplement créer des tableaux, qui se révèle très pratique. Pour créer un nouveau tableau, appelez la fonction `array_init`.

```
zval *new_array;

MAKE_STD_ZVAL(new_array);

if(array_init(new_array) != SUCCESS)
{
    // gestion des erreurs
}
```

Si `array_init` échoue à créer un tableau, elle retournera `FAILURE`.

Pour ajouter un nouvel élément au tableau, vous pouvez utiliser l'une des nombreuses fonctions conçues pour ce faire, suivant la manière que vous voulez le faire. Les tableaux [API Zend pour les tableaux associatifs](#), [API Zend pour les tableaux indexés \(Partie 1\)](#) et [API Zend pour les tableaux indexés \(Partie 2\)](#) vous décrivent toutes ces fonctions. Elles retournent toutes `FAILURE` en cas d'échec et `SUCCESS` en cas de succès.

Fonction	Description
<code>add_assoc_long(zval *array, char *key, long n);</code>	Ajoute un élément de type entier.
<code>add_assoc_unset(zval *array, char *key);</code>	Ajoute un élément indéfini.
<code>add_assoc_bool(zval *array, char *key, int b);</code>	Ajoute un élément de type booléen.
<code>add_assoc_resource(zval *array, char *key, int r);</code>	Ajoute un élément de type ressource.
<code>add_assoc_double(zval *array, char *key, double d);</code>	Ajoute un élément de type nombre à virgule flottante.
<code>add_assoc_string(zval *array, char *key, char *str, int duplicate);</code>	Ajoute un élément de type chaîne de caractères. L'option <code>duplicate</code> indique si le contenu de la chaîne doit être copié en mémoire interne Zend ou pas.
<code>add_assoc_stringl(zval *array, char *key, char *str, uint length, int duplicate);</code>	Ajoute une chaîne, tout en en précisant la taille. Sinon, se comporte comme <code>add_assoc_string</code> .

Fonction	Description
<code>add_index_long(zval *array, uint idx, long n);</code>	Ajoute un élément de type entier.
<code>add_index_unset(zval *array, uint idx);</code>	Ajoute un élément de type indéfini.
<code>add_index_bool(zval *array, uint idx, int b);</code>	Ajoute un élément de type booléen.
<code>add_index_resource(zval *array, uint idx, int r);</code>	Ajoute un élément de type ressource.
<code>add_index_double(zval *array, uint idx, double d);</code>	Ajoute un élément de type nombre à virgule flottante.
<code>add_index_string(zval *array, uint idx, char *str, int duplicate);</code>	Ajoute une chaîne au tableau. L'option <code>duplicate</code> indique si le contenu de la chaîne doit être copié en mémoire interne Zend ou pas.
<code>add_index_stringl(zval *array, uint idx, char *str, uint length, int duplicate);</code>	Ajoute une chaîne, tout en en précisant la taille. Sinon, se comporte comme <code>add_index_string</code> . Cette fonction est plus rapide, et compatible avec les données binaires.

Fonction	Description
<code>add_next_index_long(zval *array, long n);</code>	Ajoute un élément de type entier.
<code>add_next_index_unset(zval *array);</code>	Ajoute un élément de type indéfini.
<code>add_next_index_bool(zval *array, int b);</code>	Ajoute un élément de type booléen.
<code>add_next_index_resource(zval *array, int r);</code>	Ajoute un élément de type ressource.
<code>add_next_index_double(zval *array, double d);</code>	Ajoute un élément de type nombre à virgule flottante.
<code>add_next_index_string(zval *array, char *str, int duplicate);</code>	Ajoute une chaîne au tableau. L'option <code>duplicate</code> indique si le contenu de la chaîne doit être copié en mémoire interne Zend ou pas.
<code>add_next_index_stringl(zval *array, char *str, uint length, int duplicate);</code>	Ajoute une chaîne, tout en en précisant la taille. Sinon, se comporte comme <code>add_index_string()</code> . Cette fonction est plus rapide, et compatible avec les données binaires.

Toutes ces fonctions fournissent une couche d'abstraction des tables de hashage Zend. Bien sur, vous pouvez aussi utiliser directement les fonctions de hash : par exemple, si vous avez déjà une enveloppe `zval` allouée, que vous souhaitez insérer dans un tableau. Cela se fait avec la fonction `zend_hash_update()` pour les tableaux associatifs (voir [Ajouter un élément dans un tableau associatif](#)) et `zend_hash_index_update` pour les tableaux indexés (voir [Ajouter un élément dans un tableau indexé](#)) :

Ajouter un élément dans un tableau associatif
<pre>zval *new_array, *new_element; char *key = "element_key"; MAKE_STD_ZVAL(new_array); MAKE_STD_ZVAL(new_element);</pre>

```

if(array_init(new_array) == FAILURE)
{
    // gérer les erreurs ici
}

ZVAL_LONG(new_element, 10);

if(zend_hash_update(new_array->value.ht, key, strlen(key) + 1, (void *)&new_element, sizeof(zval *), NULL) == FAILURE)
{
    // gérer les erreurs ici
}

```

Ajouter un élément dans un tableau indexé

```

zval *new_array, *new_element;
int key = 2;

MAKE_STD_ZVAL(new_array);
MAKE_STD_ZVAL(new_element);

if(array_init(new_array) == FAILURE)
{
    // gérer les erreurs ici
}

ZVAL_LONG(new_element, 10);

if(zend_hash_index_update(new_array->value.ht, key, (void *)&new_element, sizeof(zval *), NULL) == FAILURE)
{
    // gérer les erreurs ici
}

```

Pour émuler la fonctionnalité de [add_next_index *](#), vous pouvez utiliser ceci :

```
zend_hash_next_index_insert(ht, zval **new_element, sizeof(zval *), NULL)
```

Note: pour retourner un tableau depuis une fonction, utilisez la fonction [array_init](#) et toutes les fonctions nécessaires sur la variable prédéfinie `return_value` (qui est fournie comme argument de vos fonctions exportées; reportez vous aux discussions sur le retour de valeurs). Vous n'avez pas à utiliser `MAKE_STD_ZVAL` avec cette variable.

Tip: Pour éviter d'avoir à écrire `new_array->value.ht` à chaque fois, vous pouvez utiliser la macro `HASH_OF(new_array)`, qui est aussi recommandée pour des raisons de compatibilité et de style.

8.10.7 Objets

Comme les objets peuvent être convertis en tableaux (et vice versa), vous pouvez imaginer que les deux auront beaucoup de similitudes. Les objets sont aussi gérés dans des tables de hashage internes à Zend, mais c'est un autre jeu de fonction qui les crée.

Pour initialiser un objet, vous pouvez utiliser la fonction [object_init](#) :

```

zval *new_object;

MAKE_STD_ZVAL(new_object);

if(object_init(new_object) != SUCCESS)
{
    // gérer les erreurs ici
}

```

Vous pouvez utiliser les fonctions décrites dans [API Zend pour la création d'objets](#) pour ajouter des membres à votre objet.

Fonction	Description

<code>add_property_long(zval *object, char *key, long l);</code>	Ajoute un membre de type entier.
<code>add_property_unset(zval *object, char *key);</code>	Ajoute un membre de type indéfini.
<code>add_property_bool(zval *object, char *key, int b);</code>	Ajoute un membre de type booléen.
<code>add_property_resource(zval *object, char *key, long r);</code>	Ajoute un membre de type ressource.
<code>add_property_double(zval *object, char *key, double d);</code>	Ajoute un membre de type nombre à virgule flottante.
<code>add_property_string(zval *object, char *key, char *str, int duplicate);</code>	Ajoute un membre de type chaîne.
<code>add_property_stringl(zval *object, char *key, char *str, uint length, int duplicate);</code>	Ajoute une chaîne de taille spécifiée. Cette fonction est plus rapide que <code>add_property_string</code> et elle gère les valeurs binaires.
<code>add_property_zval(zval *object, char *key, zval *container);</code>	Ajoute une enveloppe zval à l'objet. Cette fonction est pratique pour ajouter des membres qui sont des types complexes, comme des tableaux ou d'autres objets.

8.10.8 Ressources

Les ressources sont des types de données spéciaux en PHP. Le terme de **ressources** ne fait pas référence à un type spécial de données, mais représente une méthode d'abstraction, pour gérer n'importe quelle type d'information. Les ressources sont gérées dans une liste spéciale de Zend. Chaque entrée de cette liste dispose d'une définition de type spéciale. Zend gère en interne les références de toutes ces ressources. L'accès aux ressources n'est pas possible directement : il faut utiliser l'API fournie. Aussitôt que toutes les références à une ressources sont perdues, une fonction de destruction est appelée.

Par exemple, des ressources sont utilisées pour stocker les connexions aux bases de données, ou des pointeurs de fichiers. Le standard **de facto** d'implémentation peut être trouvée dans le module MySQL, mais d'autres modules comme celui d'Oracle utilisent aussi des ressources.

Note
En fait, une ressource peut être un pointeur sur n'importe quoi dont vous auriez besoin pour gérer vos fonctions (ce peut être un pointeur sur une structure), et l'utilisateur a simplement à passer cette variable de ressource à toutes vos fonctions.

Pour créer une nouvelle ressource, vous devez enregistrer un destructeur de ressource. Comme vous allez stocker n'importe quel type de données dans votre ressource, Zend doit savoir comment les détruire lorsque la ressource devra être libérée. Cela se fait en enregistrant un gestionnaire de ressource, qui sera appelé par Zend dès que votre ressource doit être libérée (manuellement ou automatiquement). Cet enregistrement auprès de Zend vous donne le **gestionnaire de ressource** de cette ressource. Ce gestionnaire est nécessaire à chaque fois que vous aurez à accéder à une ressource de ce type, et la plus part du temps, il est stocké dans une variable globale statique de votre extension. Il n'y a pas de problème de sécurité avec les threads, car ce gestionnaire est créé une fois, durant l'initialisation du module.

La fonction d'enregistrement de votre gestionnaire de ressource est la suivante :

```
ZEND_API int zend_register_list_destructors_ex(rsrc_dtor_func_t ld, rsrc_dtor_func_t pld, char *type_name, int module_number);
```

Il y a deux types de gestionnaire de ressources que vous pouvez passer à cette fonction : le destructeur de ressources normales, et le destructeur de ressources persistantes. Les ressources sont utilisées dans le cas des connexions aux bases de données. Lorsque vous enregistrez une ressource, l'un de ces deux gestionnaires doit être fourni. Si vous n'en utilisez qu'un seul, passez la valeur de `NULL` pour l'autre.

`zend_register_list_destructors_ex` accepte les paramètres suivants :

ld	Destructeur de ressource normale.
pld	Destructeur de ressource persistante.
type_name	Une chaîne spécifiant le nom de votre ressource. C'est toujours une bonne chose que de spécifier un nom unique dans PHP pour le type de ressource. Lorsque l'utilisateur appellera <code>var_dump</code> , il saura aussi quel type de ressource est utilisé.

module_number	La valeur de module_number est automatiquement disponible dans la macro PHP_MINIT_FUNCTION et donc, vous pouvez l'ignorer.

La valeur retournée par la fonction est un identifiant unique de votre **type de ressource** .

Les destructeurs de ressources (permanentes ou normales) ont le prototype suivant :

```
void resource_destruction_handler(zend_rsrc_list_entry *rsrc TSRMLS_DC);
```

Le pointeur ^{rsrc} référence une structure comme celle-ci :

```
typedef struct _zend_rsrc_list_entry {
    void *ptr;
    int type;
    int refcount;
} zend_rsrc_list_entry;
```

Le membre `void *ptr` est le pointeur réel de votre ressource.

Maintenant que nous savons comment démarrer, définissons la ressource que nous souhaitons dans Zend. C'est simplement une structure avec deux entiers membres :

```
typedef struct {
    int resource_link;
    int resource_type;
} my_resource;
```

Notre destructeur de ressource va probablement ressembler à ceci :

```
void my_destruction_handler(zend_rsrc_list_entry *rsrc TSRMLS_DC) {
    // Généralement, on transpasse le pointeur void vers une structure
    my_resource *my_rsrc = (my_resource *) rsrc->ptr;

    // Maintenant, faites ce que vous souhaitez avec votre ressource.
    // Fermer des fichiers, des sockets, libérez de la mémoire, etc...
    // De plus, n'oubliez pas de libérer la mémoire de votre ressource elle-même.

    do_whatever_needs_to_be_done_with_the_resource(my_rsrc);
}
```

Note

Une chose importante à citer : si votre ressource est une structure plutôt complexe, qui contient des pointeurs vers de la mémoire allouée durant l'exécution, il faut aussi les libérer mais **avant** de libérer votre ressource.

Maintenant que nous avons défini

1. Ce que notre ressource est
2. notre destructeur de ressource

nous pouvons nous consacrer aux autres étapes :

1. créer une variable globale dans l'extension pour stocker l'identifiant de ressource, pour qu'il soit accessible à toutes les fonctions qui en auront besoin
2. Définir un nom de ressource
3. Ecrire le destructeur de ressource
4. et finalement, enregistrer le gestionnaire

```
// Quelque part dans votre extension, définissez la variable contenant vos ressources enregistrées
// Si vous vous demandez ce que "le" signifie, c'est 'list entry' (élément de liste).
static int le_myresource;

// C'est une bonne chose que de donner un nom à votre ressource
#define le_myresource_name "My type of resource"

[...]

// Maintenant, codons réellement votre destructeur de ressources
void my_destruction_handler(zend_rsrc_list_entry *rsrc TSRMLS_DC) {

    my_resource *my_rsrc = (my_resource *) rsrc->ptr;
    do_whatever_needs_to_be_done_with_the_resource(my_rsrc);
}

[...]

PHP_MINIT_FUNCTION(my_extension) {

    // Notez qu8e "module_number" est déjà fournie grâce à la définition
    // de PHP_MINIT_FUNCTION().

    le_myresource = zend_register_resource_destructors_ex(my_destruction_handler, NULL, le_myresource_name, module_number);

    // Vous pouvez enregistrer d'autres ressources, initialiser vos
    // variables globales, vos constantes, etc...
}
```

Pour enregistrer réellement votre nouvelle ressource, vous pouvez soit utiliser la fonction `zend_register_resource` ou bien la macro `zend_register_resource`, qui sont toutes les deux définies dans le fichier d'entête `zend_list.h`. Bien que les arguments des deux soient exactement les mêmes, c'est une bonne idée que d'utiliser la macro pour assurer une meilleure compatibilité :

```
int ZEND_REGISTER_RESOURCE(zval *rsrc_result, void *rsrc_pointer, int rsrc_type);
```

<code>rsrc_result</code>	Une enveloppe <code>zval *</code> déjà allouée.
<code>rsrc_pointer</code>	Votre pointeur de ressource, que vous souhaitez stocker.
<code>rsrc_type</code>	Le type que vous avez reçu lorsque vous avez enregistré votre destructeur de ressources. Si vous avez suivi la convention de nommage, c'est <code>le_myresource</code> .

La valeur retournée est un identifiant de ressource unique.

Ce qui se passe en réalité lorsque vous enregistrez une nouvelle ressource, c'est que vos valeurs sont insérées dans une liste interne de Zend, et que le résultat est stocké dans l'enveloppe `zval *` fournie :

```
rsrc_id = zend_list_insert(rsrc_pointer, rsrc_type);

if (rsrc_result) {
    rsrc_result->value.lval = rsrc_id;
    rsrc_result->type = IS_RESOURCE;
}

return rsrc_id;
```

La valeur retournée `rsrc_id` identifie de manière unique la nouvelle ressource. Vous pouvez utiliser la macro `RETURN_RESOURCE` pour la retourner à l'utilisateur.

```
RETURN_RESOURCE(rsrc_id)
```

Note
C'est une pratique courante si vous souhaitez retourner immédiatement une ressource à votre utilisateur que de spécifier <code>return_value</code> comme enveloppe <code>zval *</code> .

Zend prend en charge la ressource. Aussitôt que toutes les références à cette ressources seront perdues, le destructeur que vous avez enregistré sera appelé. La magie de ce concept est que vous n'avez pas à vous soucier de fuites de mémoire, introduites par votre module : notez bien toutes les allocations que votre ressource demande. Aussitôt que cette mémoire doit être libérée, Zend vous appellera, et vous demandera de les libérer. keeps track of all references to this resource. As soon

as

Maintenant que l'utilisateur a sa ressource, et il la passe à chacune de vos fonctions qui en a besoin. Le champs `value.lval` dans l'enveloppe `zval *` contient une clé vers votre ressource, et sert donc à la récupérer, avec la macro suivante : `ZEND_FETCH_RESOURCE` :

```
ZEND_FETCH_RESOURCE(rsrc, rsrc_type, rsrc_id, default_rsrc_id, resource_type_name, resource_type)
```

<code>rsrc</code>	Votre pointeur, qui va pointer sur votre ressource enregistrée précédemment.
<code>rsrc_type</code>	C'est le type de votre pointeur, i.e. <code>myresource *</code> .
<code>rsrc_id</code>	C'est l'adresse de votre enveloppe <code>zval *</code> , que l'utilisateur a passé à votre fonction, i.e. <code>&z_resource</code> si <code>zval *z_resource</code> est fourni.
<code>default_rsrc_id</code>	Cet entier spécifie l'identifiant de ressource par défaut, si aucune ressource n'a pu être lue, ou bien <code>-1</code> .
<code>resource_type_name</code>	C'est le nom de la ressource demandée. C'est une chaîne de caractères, et elle est utilisée lorsque la ressource n'a pu être trouvée, ou qu'un message d'erreur intelligible n'a pu être constitué.
<code>resource_type</code>	Le type de ressource <code>resource_type</code> que vous avez obtenu lorsque de l'enregistrement de votre destructeur de ressource. Dans notre exemple, c'était le <code>myresource</code> .

Cette macro n'a pas de valeur de retour. C'est fait pour le confort des développeurs, et elle prend en charge les arguments de TSRMLS, ainsi que la vérification de la validité de la ressource lue. Au besoin, elle émet une alerte, et termine la fonction PHP courante avec la valeur `NULL`, si un problème de lecture de la ressource été rencontré.

Pour forcer la suppression d'une ressource de la liste, utilisez la fonction `zend_list_delete`. Vous pouvez aussi former le compteur de référence à 0 si vous savez que vous allez créer un autre référence pour une valeur précédemment allouée (par exemple, si vous réutilisez une connexion à une base de données par défaut). Pour rechercher les ressources déjà allouées, utilisez la fonction `zend_list_find`. L'API complète est disponible dans le fichier `zend_list.h`.

8.10.9 Macros de création automatique de variables globales

In addition to the macros discussed earlier, a few macros allow easy creation of simple global variables. These are nice to know in case you want to introduce global flags, for example. This is somewhat bad practice, but Table [Macros de créations de variables globales](#) describes macros that do exactly this task. They don't need any `zval` allocation; you simply have to supply a variable name and value.

Macro	Description
<code>SET_VAR_STRING(name, value)</code>	Crée une nouvelle chaîne.
<code>SET_VAR_STRINGL(name, value, length)</code>	Crée une nouvelle chaîne en spécifiant la taille. Cette macro est plus rapide <code>SET_VAR_STRING</code> et elle est compatible avec les données binaires.
<code>SET_VAR_LONG(name, value)</code>	Crée un nouvel entier.
<code>SET_VAR_DOUBLE(name, value)</code>	Crée un nouveau nombre à virgule flottante.

8.10.10 Créations de constantes

Zend supporte la création de vraies constantes (le contraire des variables habituelles). Les constantes sont accessibles sans le signe dollar (`$`) et sont disponibles dans tous les environnements. Parmi les exemples classiques, `TRUE` et `FALSE`.

Pour créer vos propres constantes, vous pouvez utiliser les macros dans la table [Macros de création des constantes](#). Toutes ces macros créent des constantes avec un nom et une valeur spécifique.

Vous pouvez aussi spécifier les options suivantes pour chaque constante :

- `CONST_CS` – Le nom de la constante doit être traité de manière insensible à la casse.

- `CONST_PERSISTENT` – Cette constante est persistante, et ne sera pas oubliée lorsque le processus actuel sera terminé.

Pour utiliser les options, combinez les par l'opérateur OR binaire :

```
// Enregistre une nouvelle constante de type "long"
REGISTER_LONG_CONSTANT("NEW_MEANINGFUL_CONSTANT", 324, CONST_CS |
CONST_PERSISTENT);
```

Il y a deux types de macros : `REGISTER_*_CONSTANT` et `REGISTER_MAIN_*_CONSTANT`. Le premier type crée des constantes liées au module. Ces constantes sont supprimées de la table des symboles aussitôt que le module est déchargé. Le second type crée des constantes qui sont en permanence dans la table des symboles, indépendamment du module.

Macro	Description
<code>REGISTER_LONG_CONSTANT(name, value, flags)</code> <code>REGISTER_MAIN_LONG_CONSTANT(name, value, flags)</code>	Crée une nouvelle constante de type long.
<code>REGISTER_DOUBLE_CONSTANT(name, value, flags)</code> <code>REGISTER_MAIN_DOUBLE_CONSTANT(name, value, flags)</code>	Crée une nouvelle constante de type double.
<code>REGISTER_STRING_CONSTANT(name, value, flags)</code> <code>REGISTER_MAIN_STRING_CONSTANT(name, value, flags)</code>	Crée une nouvelle constante de type chaîne de caractères (string). La chaîne spécifiée doit résider dans la mémoire interne de Zend.
<code>REGISTER_STRINGL_CONSTANT(name, value, length, flags)</code> <code>REGISTER_MAIN_STRINGL_CONSTANT(name, value, length, flags)</code>	Crée une nouvelle constante de type chaîne de caractères (string). La taille de la chaîne est explicitement configurée à <code>length</code> . La chaîne spécifiée ne doit pas exister en mémoire interne de Zend.

8.11 Duplique le contenu d'une variable : le bâtisseur de copie.

A un moment ou à un autre, vous aurez besoin d'assigner le contenu d'une enveloppe `zval` à une autre. C'est plus facile à dire qu'à faire, car une enveloppe `zval` ne contient pas que des données, mais aussi des références dans les rouages internes du moteur Zend. Par exemple, suivant leur taille, les tableaux et objets pourraient être imbriqués de manière complexe. En assignant une enveloppe `zval` à une autre, vous évitez la duplication de la table d'entrées, en utilisant simplement une référence.

Pour copier ce type de données complexes, utilisez le **bâtisseur de copie**. Les bâtisseurs de copie sont typiquement définis dans les langages qui supportent le surchargement d'opérateur, avec l'objectif précis de copier les objets complexes. Si vous définissez un objet dans un tel langage, vous avez la possibilité de surcharger l'opérateur "=", qui est généralement responsable de l'assignation de l'expression (résultat d'une évaluation, à gauche de l'opérateur) à la rvalue (la même chose, de l'autre côté).

Surcharger signifie assigner un nouveau sens à cet opérateur, et il sert généralement à impliquer un appel de fonction lors de l'assignation. À chaque fois que cet opérateur est utilisé sur un tel objet dans un programme, cette fonction sera appelée avec la lvalue et la rvalue comme paramètre. Connaissant ces informations, cette fonction peut réaliser l'opération convenue avec un "=" (C'est généralement une copie complexe).

C'est la même forme de copie complexe qui est nécessaire pour les enveloppes `zval` de PHP. Encore une fois, dans le cas d'un tableau, la copie complexe impliquerait la re-création de toutes les tables de hashage liées à ce tableau. Pour les chaînes de caractères, l'allocation de mémoire doit être assurée, etc...

Zend dispose d'une telle fonction appelée `zend_copy_ctor` (son équivalent précédent en PHP était la fonction `zval_copy_constructor`).

Une démonstration plus pratique est la fonction qui accepte un type complexe, le modifie et le retourne comme argument :

```
zval *parameter;

if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "z", &parameter) == FAILURE)
    return;
}

// faire quelques modifications au paramètre ici

// maintenant, nous voulons retourner une copie de l'enveloppe
*return_value == *parameter;
zval_copy_ctor(return_value);
```

La première partie de la fonction est un cas typique de récupération d'arguments. Après les modifications (laissés à votre bon plaisir), la partie intéressante commence. L'enveloppe `parameter` est assignée à l'enveloppe prédéfinie `return_value`. Maintenant, afin de pouvoir réellement dupliquer son contenu, le bâtisseur de copie est appelé. Il travaille directement avec l'argument fourni, et la valeur de retour standard de cette fonction est `FAILURE` en cas d'échec, et `SUCCESS` en cas de succès.

Si vous omettez l'appel au bâtisseur de copie dans cet exemple, les deux enveloppes `parameter` et `return_value` pointeront sur les mêmes données internes, ce qui signifie que l'enveloppe `return_value` serait une référence illégale vers les mêmes données. A chaque fois que des modifications auraient lieu dans les données référencées par `parameter`, `return_value` serait aussi affecté. Ainsi, pour créer des copies séparées, le bâtisseur de copie doit être appelé.

Le complémentaire du bâtisseur de copie de l'API Zend est le destructeur de valeur, `zval_dtor`, qui fait le contraire du bâtisseur.

8.12 Retourner des valeurs

Retourner des valeurs depuis vos fonctions vers PHP a été décrit rapidement dans une section précédente : en voici les détails. Les valeurs retournées sont passées via la variable `return_value`, qui est passée à vos fonctions comme argument. L'argument `return_value` est constitué d'une enveloppe `zval` (voir la discussion précédente sur l'API), que vous pouvez librement modifier. L'enveloppe elle-même est déjà allouée, ce qui vous évite d'avoir à utiliser `MAKE_STD_ZVAL`. A la place, vous pouvez accéder directement à ses membres.

Pour faciliter le renvoi de valeur des fonctions, et pour éviter les tracas d'accès aux structures internes, un jeu de macros est disponible (comme d'habitude). Ces macros configure le type du retour, et sa valeur, comme décrit dans les tables Macros prédéfinies pour retourner des valeurs d'une fonction et Macros prédéfinies pour configurer la valeur de retour d'une fonction.

Note
Ces macros de <u>Macros prédéfinies pour retourner des valeurs d'une fonction</u> termine termine automatiquement votre fonction. Celles de la table <u>Macros prédéfinies pour configurer la valeur de retour d'une fonction</u> ne font que préparer la valeur, sans terminer votre fonction.

Macro	Description
RETURN_RESOURCE(resource)	Retourne une ressource.
RETURN_BOOL(bool)	Retourne un booléen.
RETURN_NULL()	Retourne rien (valeur NULL).
RETURN_LONG(long)	Retourne un long.
RETURN_DOUBLE(double)	Retourne un double.
RETURN_STRING(string, duplicate)	Retourne une chaîne de caractères. L'option duplicate indique si la chaîne doit être dupliquée ou non, avec la fonction <code>estrdup</code> .
RETURN_STRINGL(string, length, duplicate)	Retourne une chaîne de caractères de taille donnée; sinon, elle se comporte exactement comme RETURN_STRING. Cette macro est bien plus rapide, et compatible avec les données binaires.

RETURN_EMPTY_STRING()	Retourne une chaîne de caractères vides.
RETURN_FALSE	Retourne un booléen FALSE .
RETURN_TRUE	Retourne un booléen TRUE .

Macro	Description
RETVAL_RESOURCE(resource)	Configure la valeur de retour avec la ressource fournie.
RETVAL_BOOL(bool)	Configure la valeur de retour avec le booléen fourni.
RETVAL_NULL	Configure la valeur de retour avec la valeur NULL .
RETVAL_LONG(long)	Configure la valeur de retour avec le long fourni.
RETVAL_DOUBLE(double)	Configure la valeur de retour avec le double fourni.
RETVAL_STRING(string, duplicate)	Configure la valeur de retour avec le chaîne de caractères fournie, et la duplique dans Zend si souhaité (voir aussi RETURN_STRING).
RETVAL_STRINGL(string, length, duplicate)	Configure la valeur de retour avec le chaîne de caractères fournie, et force la taille de cette chaîne à la valeur de length (voir aussi RETVAL_STRING). Cette macro est plus rapide, et compatible avec les données binaires. Elle doit être utilisée à chaque fois que la longueur de la chaîne est connue.
RETVAL_EMPTY_STRING	Configure la valeur de retour avec le chaîne de caractères vide.
RETVAL_FALSE	Configure la valeur de retour avec le booléen FALSE .
RETVAL_TRUE	Configure la valeur de retour avec le booléen TRUE .

Les types complexes comme les objets et les tableaux sont retournés en utilisant les fonctions `array_init` et `object_init` , ainsi que la fonction de hash `return_value` . Comme ces types ne peuvent pas être construit de manière triviale, il n'y a pas de macros prédéfinie pour eux.

8.13 Afficher des informations

Souvent, il est nécessaire d'afficher des messages depuis votre extension, tout comme avec la fonction `print` dans un script. PHP fournit des fonctions pour les tâches les plus courantes, comme par exemple l'affichage de messages d'alerte, le `phpinfo` , etc... La section suivante vous fournit plus d'information à ce sujet.

8.13.1 zend_printf

`zend_printf` fonctionne comme la fonction classique `printf` , hormis le fait qu'elle affiche le résultat dans la sortie standard de Zend.

8.13.2 zend_error

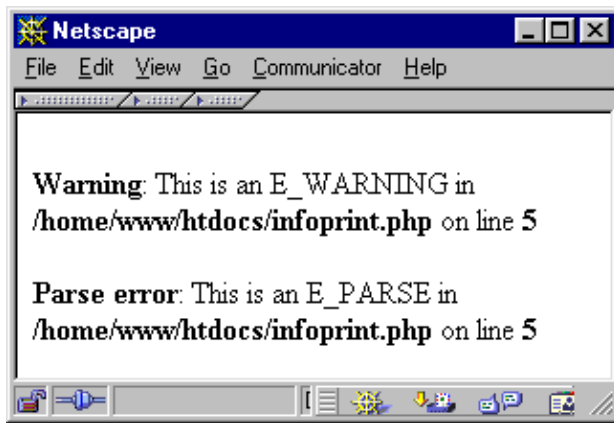
`zend_error` sert à générer des messages d'erreur. Cette fonction accepte deux arguments : le premier est le type d'erreur (voir `zend_errors.h`), et le second est le message d'erreur.

```
zend_error(E_WARNING, "This function has been called with empty arguments");
```

Messages d'erreur prédéfinis dans Zend affiche une liste des valeurs possibles (reportez vous à Affichage de messages d'erreurs dans le navigateur.). Ces valeurs sont aussi listées dans le fichier `php.ini` . Suivant le type d'erreur choisi, les messages seront enregistrés dans les historiques ou non.

Erreur	Description
E_ERROR	Signale une erreur et termine immédiatement l'exécution du script.
E_WARNING	Signale une alerte générique. L'exécution du script continue.
E_PARSE	Signale une erreur d'analyse. L'exécution du script continue.
E_NOTICE	Signale une note. L'exécution du script continue. Notez que par défaut, l'affichage de ce niveau d'erreur est inactivé dans le <code>php.ini</code> .
E_CORE_ERROR	Signale une erreur interne du coeur; ne doit pas être utilisé dans les modules externes.
E_COMPILE_ERROR	Signale une erreur interne du compilateur; ne doit pas être utilisé dans les modules externes.
E_COMPILE_WARNING	Signale une alerte interne du compilateur; ne doit pas être utilisé dans les modules externes.

Affichage de messages d'erreurs dans le navigateur.



8.13.3 Inclure du contenu dans phpinfo

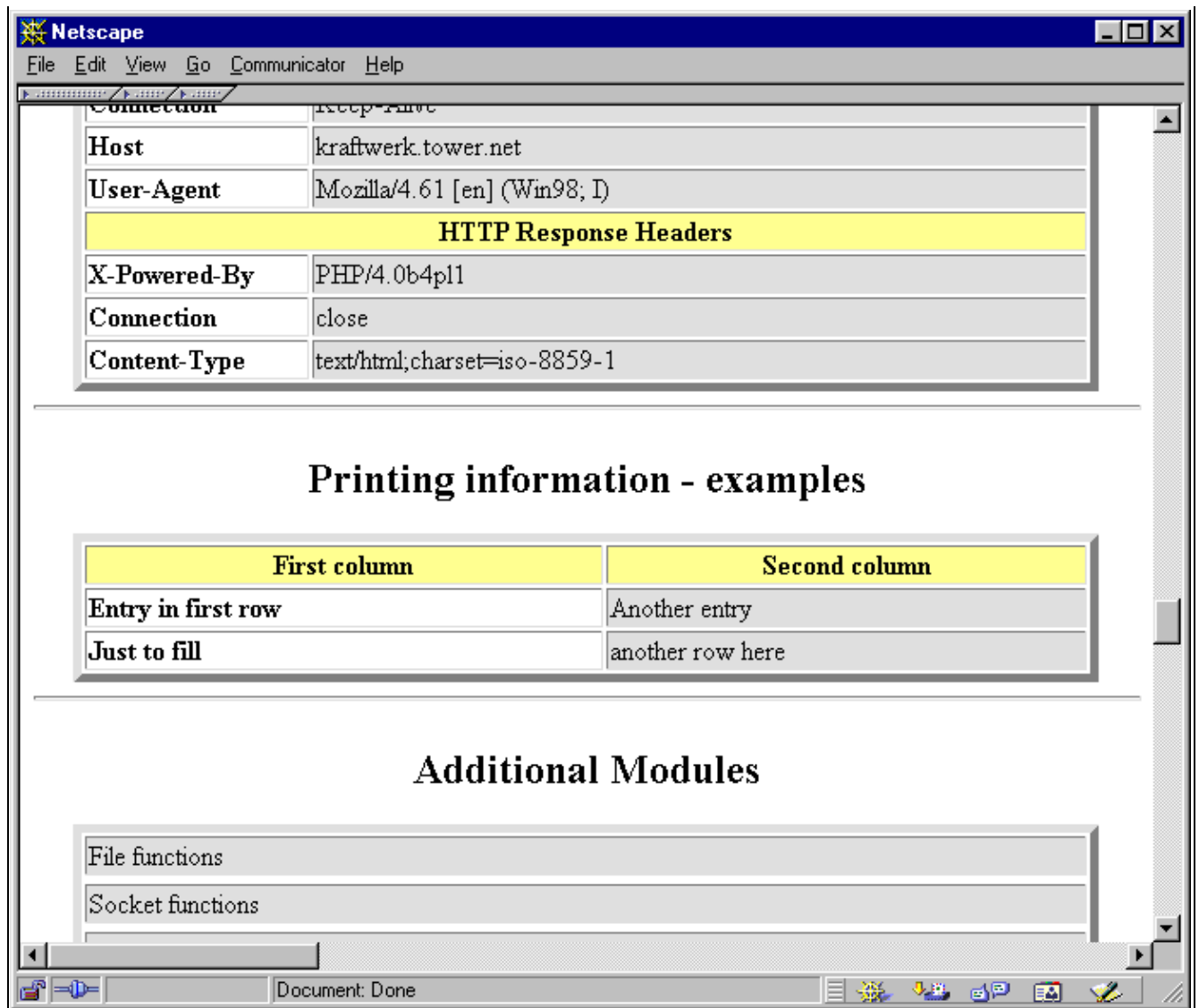
Après avoir créé un vrai module PHP, vous voudrez certainement ajouter des informations à son propos dans le résultat de la fonction [phpinfo](#) (en plus du nom du module, qui apparaît automatiquement dans la liste des modules). PHP vous permet de créer votre propre section de [phpinfo](#) avec la fonction `ZEND_MINFO()`. Cette fonction doit être placée dans le bloc descripteur de module (présenté plus tôt), et il est automatiquement appelé lorsqu'un script fait appel à [phpinfo](#).

PHP affiche automatiquement une section du [phpinfo](#) si vous spécifiez la fonction `ZEND_MINFO`, incluant le nom dans l'entête. Le reste doit être formaté et affiché par vos soins.

Typiquement, vous pouvez afficher un entête de table HTML en utilisant la fonction [php_info_print_table_start](#) puis utiliser les fonctions [php_info_print_table_header](#) et [php_info_print_table_row](#). Comme arguments, ces deux fonctions prennent un nombre de colonnes (sous forme d'entier) et le contenu de ces colonnes (sous forme de chaînes de caractères). [Code source et résultat dans phpinfo](#) montre un exemple de source et son résultat. Pour afficher le pied de table, utilisez la fonction [php_info_print_table_end](#).

Code source et résultat dans [phpinfo](#).

```
php_info_print_table_start();
php_info_print_table_header(2, "First column", "Second column");
php_info_print_table_row(2, "Entry in first row", "Another entry");
php_info_print_table_row(2, "Just to fill", "another row here");
php_info_print_table_end();
```



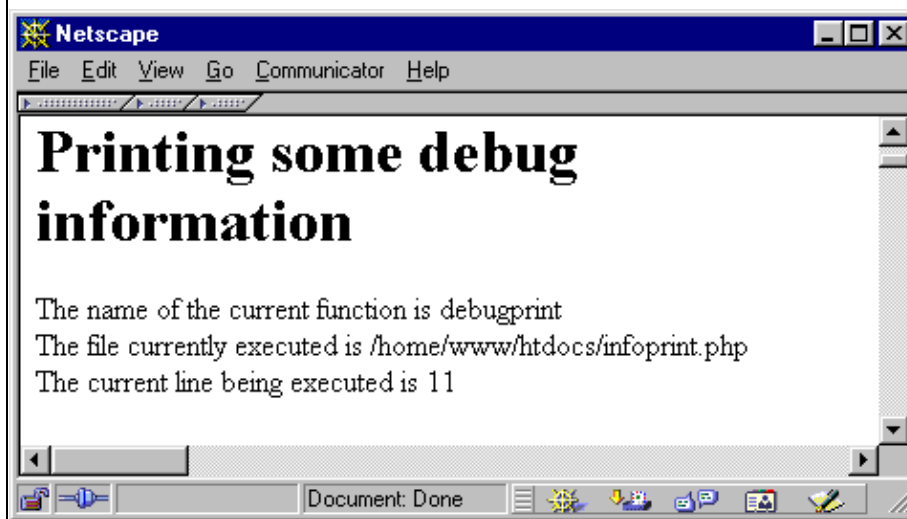
8.13.4 Informations d'exécution

Vous pouvez aussi afficher des informations d'exécution, comme le nom du fichier en cours d'exécution, ou la fonction en cours. Ce nom de la fonction actuellement utilisée est accessible via `get_active_function_name`. Cette fonction retourne un pointeur sur une fonction, et n'accepte aucun argument. Pour lire le nom du fichier en cours d'exécution, utilisez `zend_get_executed_filename`. Cette fonction accède aux variables globales, qui lui sont passées avec la macro `TSRMLS_C`. Les variables globales d'exécution sont automatiquement disponibles pour chaque fonction qui est appelée directement par Zend (ils font partie des `INTERNAL_FUNCTION_PARAMETERS` décrit plus tôt dans ce chapitre). Si vous voulez accéder aux variables globales d'exécution dans une autre fonction qui n'a pas été invoquée par Zend, appelez la macro `TSRMLS_FETCH()` une fois rendu dans la fonction. Cela vous l'introduira dans l'environnement local.

Finalement, le numéro de ligne actuellement exécuté est accessible via la fonction `zend_get_executed_lineno`. Cette fonction requiert aussi les variables globales d'exécution comme arguments. Pour une illustration de ces fonctions, voyez [Afficher des informations d'exécution](#).

Afficher des informations d'exécution.

```
zend_printf("Le nom de la fonction exécutée est %s<br>", get_active_function_name(TSRMLS_C));
zend_printf("Le nom du fichier exécuté est %s<br>", zend_get_executed_filename(TSRMLS_C));
zend_printf("Le numéro de la ligne exécutée est %i<br>", zend_get_executed_lineno(TSRMLS_C));
```



8.14 Fonctions de démarrage et d'extinction

Les fonctions de démarrage et d'extinction servent à réaliser des initialisations et des terminaisons de vos modules. Comme présenté plus tôt dans ce chapitre (reportez vous à la description du bloc Zend), il y a des événements de démarrage au niveau global, du module et de la requête.

Les fonctions de démarrage globales sont appelées lorsque PHP est démarré. De même, la fonction d'extinction est appelée lorsque PHP est éteint. Notez bien qu'elles ne sont réellement appelées **qu'une fois**, et non pas à chaque création de processus Apache.

Les fonctions de démarrage et d'extinction sont appelées lorsque le module est chargé et qu'il requiert des initialisations, ou déchargé et qu'il requiert de l'entretien. Ces fonctions de démarrage et d'extinction de module sont appelées à chaque fois qu'une requête est traitée (à chaque exécution de script).

Pour les extensions dynamiques, les événements de démarrage et d'extinction de niveau module et requêtes interviennent au même moment.

La déclaration et l'implémentation de ces fonctions peut être fait avec des macros; voyez la section ci-dessus sur la déclaration du bloc Zend.

8.15 Appeler des fonctions utilisateurs

Vous pouvez appeler des fonctions définies par l'utilisateur dans son script PHP (dites fonctions utilisateurs). C'est très pratique lorsque vous implémentez des fonctions de callback; par exemple, dans les cas de fonctions qui traitent les tableaux.

Une fonction utilisateur peut être appelée avec la fonction `call_user_function_ex`. Cette fonction nécessite une valeur de hash pour la table de fonctions que vous voulez utiliser, un pointeur vers un objet (si vous voulez en appeler une méthode), le nom de la fonction, une valeur de retour, un nombre d'arguments, un tableau d'arguments, et une option qui indique si vous souhaitez faire une séparation zval.

```
ZEND_API int call_user_function_ex(HashTable *function_table, zval *object,
zval *function_name, zval **retval_ptr_ptr,
int param_count, zval **params[],
int no_separation);
```

Notez que vous n'avez qu'à spécifier l'un des deux arguments `function_table` et `object` ; l'un ou l'autre feront l'affaire. Si vous voulez appeler une méthode, vous devez fournir un objet qui contient cette méthode, auquel cas, `call_user_function` va automatiquement utiliser la table de fonctions de l'objet. Sinon, vous n'avez qu'à spécifier `function_table`. Dans ce cas, passez la valeur `NULL` au paramètre `object`.

Généralement, la table de fonctions est la table principale (dite "root"), qui contient la liste de toutes les fonctions définies. Cette table de fonctions fait partie des variables globales du compilateur, et peut être utilisée via la macro `CG`. Pour accéder aux variables globales du compilateur, appelez la macro `TSRMLS_FETCH` une fois.

Le nom de la fonction est spécifié grâce à une enveloppe `zval`. Cela peut être un peu surprenant au premier abord, mais c'est une étape logique car la plus part du temps, vous acceptez des noms de fonctions comme paramètre, issue du script appelant. Ainsi, comme vous recevez les noms de fonctions sous forme d'enveloppe `zval`, il vous suffit de le passer directement en argument à cette fonction. Le type de cette enveloppe `zval` doit être `IS_STRING`.

L'argument suivant est constitué d'un pointeur sur la valeur de retour. Vous n'avez pas à allouer la mémoire pour cette enveloppe : la fonction le fera pour vous. Cependant, vous devrez détruire cette enveloppe (avec la fonction `zval_dtor`) après l'exécution!

Le paramètre suivant est le nombre de paramètre (un entier), puis le tableau contenant les paramètres nécessaires. Le dernier argument spécifie si la fonction doit effectuer une séparation `zval` : il est recommandé d'utiliser la valeur `0`. Si vous utilisez la valeur de `1`, la fonction va consommer moins de mémoire, mais échouera si l'un des paramètres a besoin d'une séparation `zval`.

Appeler des fonctions définies par l'utilisateur. illustre l'appel d'une fonction utilisateur. Le code appelle une fonction dont le nom est fournie par l'argument, et retourne le résultat de cette fonction directement. Notez que l'utilisation du constructeur et du destructeur à la fin : ce n'est pas nécessairement la solution ici (car les valeurs devraient être séparées, et l'assignation est blindée), mais c'est une solution techniquement sûre.

Appeler des fonctions définies par l'utilisateur.

```
zval **function_name;
zval *retval;

if((ZEND_NUM_ARGS() != 1) || (zend_get_parameters_ex(1, &function_name) != SUCCESS))
{
    WRONG_PARAM_COUNT;
}

if((*function_name)->type != IS_STRING)
{
    zend_error(E_ERROR, "Function requires string argument");
}

TSRMLS_FETCH();

if(call_user_function_ex(CG(function_table), NULL, *function_name, &retval, 0, NULL, 0) != SUCCESS)
{
    zend_error(E_ERROR, "Function call failed");
}

zend_printf("Nous avons %i de type<br>", retval->type);

*return_value = *retval;
zval_copy_ctor(return_value);
zval_ptr_dtor(&retval);
```

```

<?php
dl("call_userland.so");

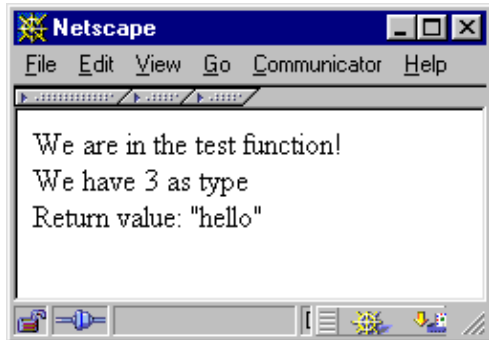
function test_function()
{
    print("Nous sommes dans la fonction de test!<br>");

    return("bonjour");
}

$return_value = call_userland("test_function");

print("Valeur retournée : \"\$return_value\"<br>");
?>

```



8.16 Support du fichier d'initialisation File Support

PHP 4 fournit un nouveau système pour le support du fichier d'initialisation. Il est possible de spécifier des valeurs par défaut d'option de configuration directement dans votre code, lire et modifier ces valeurs lors de l'exécution, et créer des gestionnaires pour être prévenu lors de modifications de ces options.

Pour créer une section .ini dans votre propre module, utilisez les macros `PHP_INI_BEGIN()` pour marquer le début de cette section, et `PHP_INI_END()` pour en marquer la fin. Entre les deux, vous pouvez utiliser `PHP_INI_ENTRY()` pour créer des entrées.

```

PHP_INI_BEGIN()
PHP_INI_ENTRY("premiere_entree_ini", "une chaîne de caractères", PHP_INI_ALL, NULL)
PHP_INI_ENTRY("deuxieme_entree_ini", "2", PHP_INI_SYSTEM, OnChangeSecond)
PHP_INI_ENTRY("troisieme_entree_ini", "xyz", PHP_INI_USER, NULL)
PHP_INI_END()

```

La macro `PHP_INI_ENTRY()` accepte quatre paramètres : le nom de l'option, la valeur, les droits de modification et un pointeur sur un gestionnaire de modification. Le nom et la valeur doivent être spécifiés sous forme de chaîne de caractères, quelque soit leur type réel.

Les droits sont regroupés en trois sections : `PHP_INI_SYSTEM` permet de modifier l'option uniquement dans le fichier `php.ini` ; `PHP_INI_USER` permet les modifications au moment de l'exécution, avec un fichier complémentaire de configuration qui remplace la configuration du `php.ini`, comme par exemple `.htaccess` ; et `PHP_INI_ALL` qui permet la modification de l'option sans aucune restriction. Il y a aussi un quatrième niveau de droits, `PHP_INI_PERDIR`, mais il n'est pas en service actuellement.

Le quatrième paramètre est constitué d'un pointeur sur un gestionnaire de modifications. A chaque fois que ces options de configurations sont modifiées, ce gestionnaire est appelé. Un tel gestionnaire doit être déclaré en utilisant la macro `PHP_INI_MH` :

```

PHP_INI_MH(OnChangeSecond); // gestionnaire pour "deuxieme_entree_ini"

```



```
// spécifiez les entrées ici

PHP_INI_MH(OnChangeSecond)
{
    zend_printf("Message lu, notre option a été modifiée en %s<br>", new_value);

    return(SUCCESS);
}
```

La nouvelle valeur est fournie comme paramètre du gestionnaire, sous forme de chaîne de caractères, dans la variable `new_value`. Lorsque vous regardez la définition de `PHP_INI_MH`, vous trouvez plusieurs paramètres :

```
#define PHP_INI_MH(name) int name/php_ini_entry *entry, char *new_value,
                        uint new_value_length, void *mh_arg1,
                        void *mh_arg2, void *mh_arg3)
```

Toutes ces définitions sont accessibles dans le fichier `php_ini.h`. Votre gestionnaire de modification aura accès à la structure qui contient l'option complète, sa nouvelle valeur, sa longueur, et trois arguments optionnels. Ces arguments optionnels peuvent être spécifiés avec les macros `PHP_INI_ENTRY1` (qui permet un argument supplémentaire), `PHP_INI_ENTRY2` (qui permet deux arguments supplémentaires), et `PHP_INI_ENTRY3` (qui permet trois arguments supplémentaires).

Le gestionnaire de modification doit être utilisé pour mettre en cache les valeurs initiales localement, afin d'accélérer les accès ultérieurs, ou bien pour effectuer certaines tâches complémentaires. Par exemple, si une connexion permanente est nécessaire à un module, et que l'utilisateur en modifie le nom de serveur hôte, vous pourriez terminer automatiquement l'ancienne connexion et en ouvrir une autre.

L'accès aux options d'initialisation peut aussi être réalisé avec les macros suivantes :

Macro	Description
<code>INI_INT(name)</code>	Retourne la valeur courante de name sous forme d'entier (long).
<code>INI_FLT(name)</code>	Retourne la valeur courante de name sous forme de nombre à virgule flottante (double).
<code>INI_STR(name)</code>	Retourne la valeur courante de name sous forme de chaîne de caractères. Note: Cette chaîne n'est pas dupliquée mais pointe sur des données internes. Pour y accéder, il faudra la dupliquer en mémoire locale.
<code>INI_BOOL(name)</code>	Retourne la valeur courante de name sous forme booléen (définis comme <code>zend_bool</code> , ce qui signifie actuellement <code>unsigned char</code>).
<code>INI_ORIG_INT(name)</code>	Retourne la valeur courante de name sous forme d'entier (long).
<code>INI_ORIG_FLT(name)</code>	Retourne la valeur courante de name sous forme de nombre à virgule flottante (double).
<code>INI_ORIG_STR(name)</code>	Retourne la valeur courante de name sous forme de chaîne de caractères. Note: Cette chaîne n'est pas dupliquée mais pointe sur des données internes. Pour y accéder, il faudra la dupliquer en mémoire locale.
<code>INI_ORIG_BOOL(name)</code>	Retourne la valeur courante de name sous forme booléen (définis comme <code>zend_bool</code> , ce qui signifie actuellement <code>unsigned char</code>).

Finalement, vous devez présenter vos options à PHP. Cela peut être fait dans lors du démarrage et d'extinction du module, en utilisant les macros `REGISTER_INI_ENTRIES()` et `UNREGISTER_INI_ENTRIES()` :

```
ZEND_MINIT_FUNCTION(mymodule)
{
    REGISTER_INI_ENTRIES();
}

ZEND_MSHUTDOWN_FUNCTION(mymodule)
{
    UNREGISTER_INI_ENTRIES();
}
```

8.17 Par où continuer?

Vous en avez appris beaucoup sur PHP. Vous savez désormais comment créer des modules dynamiques et statiques. Vous savez comment PHP et Zend gèrent les ressources internes, et vous savez comment créer et accéder à ces variables. Vous connaissez tout un jeu de macros pour réaliser des tâches routinières comme l'affichage de messages informatifs, la récupération des paramètres, etc...

Même si ce chapitre a souvent eu un caractère de référence, nous espérons qu'il vous a donné une bonne introduction sur la méthode à suivre pour écrire vos propres extensions. Dans un souci de concision, nous en avons laissé beaucoup de côté. Nous vous suggérons de prendre le temps d'étudier les fichiers d'entêtes, et certains modules (notamment ceux qui sont installés dans le dossier `ext/standard` et le module MySQL, qui implémente les fonctionnalités les plus standards. Cela vous donnera des idées sur comment d'autres développeurs que vous ont utilisé ces API, notamment celles qui n'ont pas été introduites dans ce chapitre.

8.18 Référence : quelques macros de configuration

8.18.1 config.m4

Le fichier `config.m4` est traité par `buildconf` et doit contenir toutes les instructions nécessaires à la configuration. Par exemple, il peut inclure des tests de présence de fichiers externes, de bibliothèques, etc... PHP définit un jeu de macros qui peuvent être utilisées dans ce but. Les plus utiles sont décrites dans la table Macros M4 pour config.m4.

Macro	Description
AC_MSG_CHECKING(message)	Affiche le message "checking <message>" durant l'exécution de configure.
AC_MSG_RESULT(value)	Affiche le résultat de AC_MSG_CHECKING ; Le résultat doit prendre les valeurs de yes ou no comme value.
AC_MSG_ERROR(message)	Affiche message comme message d'erreur durant l'exécution du script configure et interrompt le script.
AC_DEFINE(name, value, description)	Ajoute un #define dans php_config.h avec la valeur de value et le commentaire description (c'est pratique pour les compilations conditionnelles de votre module).
AC_ADD_INCLUDE(path)	Ajoute un chemin d'inclusion pour le compilateur. Par exemple, c'est utilisé pour les modules qui doivent rechercher les fichiers d'entête dans des dossiers séparés.
AC_ADD_LIBRARY_WITH_PATH(libraryname, librarypath)	Spécifie un lien vers une bibliothèque supplémentaire.
AC_ARG_WITH(modulename, description, unconditionaltest, conditionaltest)	Une macro très puissante, qui ajoute le module et sa description dans le résultat de configure --help. PHP vérifie si l'option --with-<modulename> est fournie au script configure. Si c'est le cas, il exécute le script unconditionaltest (par exemple, --with-mysql=yes), auquel cas la valeur de l'option est contenue dans la variable \$withval. Sinon, il exécute conditionaltest.
PHP_EXTENSION(modulename, [shared])	Cette macro est un impératif , pour que PHP configure votre extension. Vous pouvez lui fournir un second argument en plus de votre nom de module, qui indique si vous envisagez la compilation sous forme de module partagé. Cela donnera une définition pour la compilation de votre source comme <code>COMPILE_DL_<modulename></code> .

8.19 API Macros

Un jeu de macros a été introduit dans l'API Zend pour simplifier l'accès aux enveloppes `zval` (voir Macros API d'accès aux enveloppes zval).

Macro	Représente
Z_LVAL(zval)	(zval).value.lval

Z_DVAL(zval)	(zval).value.dval
Z_STRVAL(zval)	(zval).value.str.val
Z_STRLEN(zval)	(zval).value.str.len
Z_ARRVAL(zval)	(zval).value.ht
Z_LVAL_P(zval)	(*zval).value.lval
Z_DVAL_P(zval)	(*zval).value.dval
Z_STRVAL_P(zval_p)	(*zval).value.str.val
Z_STRLEN_P(zval_p)	(*zval).value.str.len
Z_ARRVAL_P(zval_p)	(*zval).value.ht
Z_LVAL_PP(zval_pp)	(**zval).value.lval
Z_DVAL_PP(zval_pp)	(**zval).value.dval
Z_STRVAL_PP(zval_pp)	(**zval).value.str.val
Z_STRLEN_PP(zval_pp)	(**zval).value.str.len
Z_ARRVAL_PP(zval_pp)	(**zval).value.ht

9 PHP API: Interface pour les extensions

9.1 API de flôts les auteurs d'extensions

9.1.1 Présentation

Les flôts de PHP proposent une approche unifiée à la gestion des fichiers et sockets, depuis une extension PHP. En utilisant l'API standard, les flôts vous permettent d'accéder à des fichiers, des sockets, des URL, de la mémoire ou encore des objets définis dans des scripts. Les flôts (streams en anglais), sont une API flexible et dynamique, dynamiquement chargée dans les modules (et les scripts!).

Le but des flôts 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 dépayés.

L'API des flôts 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.

Les flôts peuvent être converti en pilotes pour fichiers, de manière à être utilisé par des librairies tierce-partie avec beaucoup de facilité. Cela autorise ces librairies à accéder aux données directement depuis des sources représentées par des URL. Si votre système supporte la fonction [fopencookie](#), vous pouvez même passer n'importe quel flôt PHP à une autre librairie qui utilise les entrée/sortie standard ANSI!

9.1.2 Eléments de bases

Utiliser les flôts ressemble énormément à l'utilisation des fonctions d'entrée/sortie standard ANSI streams. La principale différence est que vous devez créer le pointeur de flôt 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 flôt qui affiche la page de 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 flôts équivalentes aux fonctions ANSI. Sauf indication contraire, les fonctionnalités sont les mêmes.

Fonction ANSI	Fonction Flôt PHP	Notes
fopen	php_stream_open_wrapper	Les flôts incluent des paramètres supplémentaires
fclose	php_stream_close	
fgets	php_stream_gets	
fread	php_stream_read	Le paramètre nmemb est supposé avoir la valeur 1 : le prototype ressemble plutôt à read(2)
fwrite	php_stream_write	Le paramètre nmemb est supposé avoir la valeur 1 : le prototype ressemble plutôt à write(2)
fseek	php_stream_seek	
ftell	php_stream_tell	
rewind	php_stream_rewind	
feof	php_stream_eof	
fgetc	php_stream_getc	
fputc	php_stream_putc	
fflush	php_stream_flush	
puts	php_stream_puts	Identique à puts, et non pas à fputs
fstat	php_stream_stat	Les informations des flôts sont plus riches

9.1.3 Les flôts qui sont des ressources

Tous les flôts 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 flôts. L'API des flôts rend ce processus très convivial.

Comment accepter un flôt comme paramètre d'entrée
<pre> 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 flôt. Cependant, vous ne "possédez" pas le flôt, mais le script le possède. Cela signifie que vous DEVEZ PAS fermer le flôt, car sinon, PHP va crasher! */ php_stream_write(stream, "hello\n"); RETURN_TRUE(); } </pre>

Comment retourner un flôt depuis une fonction
<pre> 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 flôt appartient au script. Si vous le fermez ici, vous allez faire crasher PHP! */ } </pre>

Comme les flôts 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 flôt 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 flôts conservent des verrous sur les ressources systèmes durant un temps plus long, ce qui fait que laisser le verrou 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 flôt inclut un peu de magie pour que vous puissiez garder votre code propre : si un flôt 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 (<code>--enable-debug</code> 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 flôt 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 flôt 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](#).

9.1.4 Référence des API communes de flôts

9.1.4.1 `php_stream_stat_path()` : Lit l'état du fichier ou de l'URL

```
int php_stream_stat_path ( char * path , php_stream_statbuf * ssb )
```

[php_stream_stat_path](#) examine le fichier ou l'URL spécifié par ^{path} et retourne des informations telles que la taille, les droits d'accès, la date de création, etc... La valeur retournée est 0 en cas de succès, et -1 en cas d'échec. Pour plus d'informations, voyez [php_stream_statbuf](#).

9.1.4.2 `php_stream_stat()` : Lit le statut du cache associé au flôt

```
int php_stream_stat ( php_stream * stream , php_stream_statbuf * ssb )
```

[php_stream_stat](#) examine le système de stockage du flôt ^{stream}, et retourne des informations telles que la taille de fichier, les droits d'accès, la date de création, etc... La valeur retournée est 0 en cas de succès, et -1 en cas d'échec. Pour plus d'informations sur les informations retournées, voyez [php_stream_statbuf](#).

9.1.4.3 `php_stream_open_wrapper()` : Ouvre un flôt sur un fichier ou une URL

```
php_stream * php_stream_open_wrapper ( char * path , char * mode , int options , char ** opened )
```

[php_stream_open_wrapper](#) ouvre un fichier, une URL ou n'importe quelle ressource gérée, spécifiée par ^{path}. Suivant la valeur de ^{mode}, le flôt peut être ouvert en lecture, écriture, ajout ou une combinaison des trois. Voyez la table ci-dessous pour connaître les modes utilisées. De plus vous pouvez aussi intégrer le caractère 'b' comme deuxième ou dernière option. Sa présence indiquera que le flôt fonctionnera en mode binaire.

L'option 'b' est ignorée sur les systèmes POSIX, qui traitent les fichiers textes et binaires de la même façon. C'est une bonne idée que de spécifier 'b' à chaque fois que votre flôt accède à des données binaires, pour que votre code fonctionne lorsqu'il est compilé sur un système où 'b' serait important.

Tous les fichiers locaux créés par l'API des flôts recevront les droits par défaut du système : sous

Unix, cela signifie que le masque utilisateur du processus sera utilisé. Sous Windows, le fichier sera donné au propriétaire du processus. Tous les fichiers distants seront créés suivant le protocole utilisé, et les droits utilisés seront ceux du serveur distant.

<code>r</code>	Ouvre le fichier texte en mode lecture. Le pointeur du flôt est placé au début du fichier.
<code>r+</code>	Ouvre le fichier texte en lecture et écriture. Le pointeur du flôt est placé au début du fichier.
<code>w</code>	Tronque le fichier à une taille de 0 ou bien crée le fichier, puis l'ouvre en mode écriture. Le pointeur du flôt est placé au début du fichier.
<code>w+</code>	Tronque le fichier à une taille de 0 ou bien crée le fichier, puis l'ouvre en mode écriture et lecture. Le pointeur du flôt est placé au début du fichier.
<code>a</code>	Ouvre le fichier en mode d'écriture. Le fichier est créé si il n'existe pas. Le pointeur de flôt est positionné à la fin du fichier.
<code>a+</code>	Ouvre le fichier en mode d'écriture et lecture. Le fichier est créé si il n'existe pas. Le pointeur de flôt est positionné à la fin du fichier.

`options` affecte l'interprétation du path ou de l'URL, les vérifications du `safe_mode` et les actions à prendre lorsqu'il y a une erreur. Voyez [Stream open options](#) pour plus de détails sur ces options.

Si `opened` est non NULL, il sera assigné avec le nom du fichier ou de la ressource qui aura été réellement ouverte. C'est important lorsque l'option inclus `USE_PATH`, qui permet de fouiller l'`include_path` pour rechercher le fichier. Vous, l'appelant, êtes responsable de l'appel de `efree` sur le nom de fichier retourné par ce paramètre.

Note
Si vous avez spécifié l'option <code>STREAM_MUST_SEEK</code> dans <code>options</code> , le chemin retourné dans <code>opened</code> peut ne pas être le même que celui qui vous a été réellement retourné. Toutefois, ce sera le nom de la ressource originale à partir de laquelle un flôt à pointeur variable a été créé.

9.1.4.4 `php_stream_read()` : Lit des octets depuis un flôt dans un buffer

```
size_t php_stream_read ( php_stream * stream, char * buf, size_t count )
```

`php_stream_read` lit jusqu'à `count` octets dans le flôt `stream` puis les copie dans le buffer `buf`.

`php_stream_read` retourne le nombre d'octets lus. Il n'y a pas de distinction entre une lecture qui échoue et la fin du fichier : utilisez `php_stream_eof` pour tester la fin du fichier avec `EOF`.

Le pointeur de position dans le flôt est avancé d'autant d'octets que le nombre qui en a été lu. Par conséquent, la prochaine lecture reprendra là où la lecture précédente s'est arrêtée.

Si moins de `count` octets ont été lus, cet appel bloquera l'application jusqu'à ce que le nombre d'octets nécessaire soit atteint, en fonction de la configuration du flôt (bloquant, non bloquant). Par défaut, un flôt est ouvert en mode bloquant. Lorsque vous lisez dans des fichiers classiques, le mode bloquant ne vous gênera pas. Lorsque le flôt atteint `EOF` `php_stream_read` retournera un nombre d'octets lus inférieur à `count`, puis il retournera 0 aux prochaines lectures.

9.1.4.5 php_stream_write() : écrit des octets depuis un buffer vers un flôt

```
size_t php_stream_write ( php_stream * stream , const char * buf , size_t count )
```

php_stream_write écrit ^{count} octets du buffer ^{buf} dans le flôt ^{stream} .

php_stream_write retourne le nombre d'octets qui ont été écrit. Si il y a eu une erreur, le nombre d'octets écrit sera inférieur au paramètre ^{count} .

Le pointeur de position dans le flôt est avancé d'autant d'octets que le nombre qui en a été écrit. Par conséquent, la prochaine écriture reprendra là ou la lecture précédente s'est arrêté.

9.1.4.6 php_stream_eof() : Vérifie que le pointeur n'est pas au bout du flôt

```
int php_stream_eof ( php_stream * stream )
```

php_stream_eof vérifie que le pointeur n'est pas au bout du flôt ^{stream} .

php_stream_read retourne 1 pour indiquer la fin du flôt (^{EOF}), 0 si il n'y a pas de fin de flôt, et -1 pour indiquer une erreur.

9.1.4.7 php_stream_getc() : Lit un octet dans un flôt

```
int php_stream_getc ( php_stream * stream )
```

php_stream_getc lit un octet dans le flôt ^{stream} puis le retourne de type unsigned char transtypé en int, ou ^{EOF} si la fin du flôt est atteinte, ou bien si une erreur est survenue.

php_stream_getc peut bloquer l'application de la même façon que php_stream_read bloque.

Le pointeur de position est avancé d'une position si la lecture réussit.

9.1.4.8 php_stream_gets() : Lit une ligne de données d'un flôt dans un buffer

```
char * php_stream_gets ( php_stream * stream , char * buf , size_t maxlen )
```

php_stream_gets lit jusqu'à ^{count} -1 octets de données dans le flôt ^{stream} puis les copie dans le buffer ^{buf} . La lecture s'arrête après la rencontre de la fin du fichier (^{EOF}) ou une nouvelle ligne. Si la nouvelle ligne est lue, elle est stockée dans ^{buf} dans les données retournées. Un caractère NUL de fin de chaîne est stocké comme dernier caractère dans le buffer.

php_stream_read retourne ^{buf} lorsque cela fonctionne, et NULL sinon.

Le pointeur de position est avancé d'une position si la lecture réussit.

php_stream_gets peut bloquer l'application de la même façon que php_stream_read bloque.

9.1.4.9 php_stream_close() : Ferme un flôt

```
int php_stream_close ( php_stream * stream )
```

php_stream_close ferme correctement le flôt `stream` et libère les ressources qui lui sont associées. Après la fermeture de `stream`, il prend une valeur indéfinie, et ne devrait plus être utilisé.

php_stream_close retourne 0 si le flôt a été fermé, ou bien `EOF` pour indiquer une erreur. Indépendamment du succès de l'opération, `stream` prend une valeur indéfinie, et ne devrait plus être utilisé.

9.1.4.10 php_stream_flush() : Vide les buffers dans le flôt

```
int php_stream_flush ( php_stream * stream )
```

php_stream_flush vide toutes les données contenues dans les buffers d'écriture du flôt `stream` et force leur stockage.

php_stream_flush retourne 0 si les buffers ont été vidés, ou si les buffers n'ont pas besoin d'être vidés, mais retourne `EOF` pour indiquer une erreur.

9.1.4.11 php_stream_seek() : Modifie la position du pointeur de flôt

```
int php_stream_seek ( php_stream * stream , off_t offset , int whence )
```

php_stream_seek modifie la position du pointeur de position du flôt `stream`. La nouvelle position est déterminée en ajoutant `offset` à la position indiquée par `whence`. Si `whence` vaut `SEEK_SET`, `SEEK_CUR` ou `SEEK_END` l'offset sera relatif au début du flôt, à la position courante ou à la fin du flôt, respectivement.

php_stream_seek retourne 0 en cas de succès, et -1 en cas d'erreur.

Note

Tous les flôts ne supportent pas la modification de la position du pointeur, même si l'API va émuler un tel pointeur si `whence` vaut `SEEK_CUR` et `offset` est positif, en appelant php_stream_read pour lire (et ignorer) `offset` octets.

L'émulation est uniquement appliquée si l'implémentation du gestionnaire de flôt supporte le pointeur de position. Si le flôt est (par exemple) basé sur des fichiers qui travaille avec des flôt de pipes, l'API de flôt ne pourra pas appliquer une telle émulation. Le positionnement du pointeur de flôt échouera et une erreur sera retournée.

9.1.4.12 php_stream_tell() : Détermine la position du pointeur dans un flôt

```
off_t php_stream_tell ( php_stream * stream )
```

php_stream_tell retourne la position du pointeur du flôt `stream`, relativement au début du flôt. Si une erreur survient, retourne -1.

9.1.4.13 php_stream_copy_to_stream() : Copie des données d'un flôt à l'autre

```
size_t php_stream_copy_to_stream ( php_stream * src , php_stream * dest , size_t maxlen )
```

`php_stream_copy_to_stream` essaie de lire un maximum de `maxlen` octets dans le flôt `src` puis de les écrire dans le flôt `dest`. La fonction retourne le nombre d'octets copiés.

Si vous souhaitez copier tous les octets restants dans le flôt `src`, passez la constante `PHP_STREAM_COPY_ALL` comme valeur de `maxlen`.

Note

Cette fonction va essayer d'effectuer la copie de la manière la plus efficace, en utilise dans buffer mémoire lorsque c'est possible.

9.1.4.14 `php_stream_copy_to_mem()` : Copie des données depuis un flôt dans un buffer

```
size_t php_stream_copy_to_mem ( php_stream * src, char ** buf, size_t maxlen, int persistent )
```

`php_stream_copy_to_mem` alloue un buffer de taille `maxlen + 1` en utilisant la fonction `pcalloc` (en passant `persistent`). Puis, elle lit `maxlen` octets depuis le flôt `src` and stores them in the allocated buffer.

The allocated buffer is returned in `buf`, and the number of bytes successfully read. You, the caller, are responsible for freeing the buffer by passing it and `persistent` to `pcfree`.

If you want to copy all remaining data from the `src` stream, pass the constant `PHP_STREAM_COPY_ALL` as the value of `maxlen`.

Note

Cette fonction va essayer d'effectuer la copie de la manière la plus efficace, en utilise dans buffer mémoire lorsque c'est possible.

9.1.4.15 `php_stream_make_seekable()` : Convertit un flôt en flôt avec un pointeur de position

```
int php_stream_make_seekable ( php_stream * origstream, php_stream ** newstream, int flags )
```

`php_stream_make_seekable` vérifie si `origstream` dispose d'un pointeur de position. Si ce n'est pas le cas, il va copier les données du flôt dans un autre flôt temporaire, et disposant d'un pointeur de position. Si cela fonctionne, `newstream` sera un flôt valide même si le flôt original avait déjà un pointeur de position.

`flags` vous permet de spécifier votre préférence pour le flôt retourné : utilisez la constante `PHP_STREAM_NO_PREFERENCE` pour utiliser le flôt à pointeur de position par défaut (qui utilise la mémoire, mais passe automatiquement sur un fichier si le fichier devient trop grand), ou bien utilisez `PHP_STREAM_PREFER_STDIO` pour utiliser un fichier temporaire classique.

Valeur	Signification
<code>PHP_STREAM_UNCHANGED</code>	Le flôt original avait déjà son pointeur de position. <code>newstream</code> vaut <code>origstream</code> .
<code>PHP_STREAM_RELEASED</code>	Le flôt originl n'avait pas pointeur de position, et il a été détruit. Vous n'avez plus accès à <code>origstream</code> .
<code>PHP_STREAM_FAILED</code>	Une erreur est survenue lors de la conversion. <code>newstream</code> vaut <code>NULL</code> ; <code>origstream</code> est toujours valide.
<code>PHP_STREAM_CRITICAL</code>	Une erreur est survenue lors de la conversion de <code>origstream</code> , qui est maintenant dans un état indéterminé. <code>newstream</code> vaut <code>NULL</code> et il est très recommandé que vous vous fermiez <code>origstream</code> .

Note

Si vous devez déplacer le pointeur et écrire dans le flôt, n'utilisez pas cette fonction, car le flôt retourné est un flôt mis en cache. Il n'est pas garanti qu'il soit lié à la ressource originale.

Note

Si vous avez uniquement de déplacer le pointeur de position vers l'avant, il n'y a pas besoin d'appeler cette fonction, car l'API va automatiquement émuler ce fonctionnement avec le paramètre

whence à la valeur `SEEK_CUR`.

Note

Si `origstream` repose sur le réseau, cette fonction va se bloquer jusqu'à ce que tout le contenu soit téléchargé.

Note

Ne JAMAIS appeler cette fonction avec un flôt `origstream` qui est une référence à un pointeur de fichier dans un script PHP. Cette fonction pourrait alors fermer le fichier, ce qui conduira à un crash au prochain accès au pointeur.

Note

Dans de nombreux cas, cette fonction ne peut réussir que si `origstream` vient d'être tout juste ouvert, et qu'aucune données n'ont été mises dans les caches de la couche de flôt. Pour cette raison, et parceque cette fonction est compliquée à utiliser correctement, il est recommandé que vous utilisiez la fonction `php_stream_open_wrapper` en lui passant la constante `PHP_STREAM_MUST_SEEK` dans vos options, au lieu d'appeler cette fonction directement.

9.1.4.16 `php_stream_cast()` : Convertit un flôt en une autre forme, comme un FILE* ou une socket

```
int php_stream_cast ( php_stream * stream , int castas , void ** ret , int flags )
```

`php_stream_cast` tente de convertir le flôt `stream` en un autre ressource indiqué par `castas`. Si `ret` est NULL, le flôt est interrogé pour savoir une telle conversoin est possible, sans effectuer réellement la conversion (mais certains état interne pourrait être changés dans ce cas). Si `flags` vaut `REPORT_ERRORS`, un message d'erreur apparaîtra si une erreur survient durant la conversion.

Note

Cette fonction retourne `SUCCESS` en cas de réussite et `FAILURE` en cas d'échec. Soyez prévenu que vous devez faire une comparaison explicite avec `SUCCESS` ou `FAILURE` à cause des valeurs sous jacentes de ces constantes. Une simple comparaison d'expression booléenne ne sera pas interprétée comme vous le souhaitez.

Valeur	Signification
<code>PHP_STREAM_AS_STDIO</code>	Un pointeur ANSI FILE* qui représente le flôt
<code>PHP_STREAM_AS_FD</code>	Un pointeur de fichier de type POSIX qui représente le flôt
<code>PHP_STREAM_AS_SOCKETD</code>	Un pointeur de socket réseau qui représente le flôt

En plus des types de ressources basiques ci-dessus, le processus de conversion peut être modifié en utilisant les paramètres ci-dessous, ou bien en utilisant l'opérateur OR pour combiner le type de ressource avec une ou plusieurs des valeurs suivantes :

Valeur	Signification
<code>PHP_STREAM_CAST_TRY_HARD</code>	Essaie aussi fort que possible, même si c'est au prix de quelques ressources supplémentaires, que la conversion réussisse
<code>PHP_STREAM_CAST_RELEASE</code>	Informe l'API de flôt que d'autre code (possible une librairie tierce partie) sera responsable de la gestion des ressources de bas niveau. Cela fait que le flôt stream sera fermé de telle façon à ce que le gestionnaire soit préservé et retourné dans le paramètre <code>ret</code> . Si la fonction réussit, le paramètre stream doit être considéré comme fermé, et inutilisable en l'état.

Note

Si votre système supporte la fonction `fopencookie` (les systèmes utilisant la librairie glibc 2 ou plus récent), l'API de flôt sera toujours capable de créer des pointeurs de fichiers de type ANSI FILE* pour tous les flôts. Même si c'est extrêmement utile pour passer des flôts PHP à des librairie tierce partie, un tel système n'est pas portable. Il est demandé que vous considériez la portabilité e vos fonctions avant de diffuser votre extension. Si l'utilisation de `fopencookie()` n'est pas désirable, il est

recommandé d'interroger le flût pour voir si il supporte naturellement la conversion en FILE* en utilisant la fonction [php_stream_is](#)

Note

Si vous demandez une socket réseau à la place d'un FILE*, l'API de flûts va utiliser la fonction [fdopen](#) pour la créer pour vous. Soyez prévenu que faire cela risque de perdre les données qui ont été mises en buffer dans le flût, si vous mélanger des appels de l'API de flût et les appels standards ANSI.

Voir aussi [php_stream_is](#) et [php_stream_can_cast](#).

9.1.4.17 php_stream_can_cast() : Détermine si un flût peut être converti en une autre forme, comme FILE* ou socket

```
int php_stream_can_cast ( php_stream * stream , int castas )
```

[php_stream_can_cast](#) est l'équivalent de la fonction [php_stream_cast](#) avec le paramètre `ret` mis à NULL, et le paramètre `flags` mis à 0. Elle retourne `SUCCESS` si le flût peut être converti dans la forme demandée, ou bien `FAILURE` si la conversion ne peut se faire.

Note

Bien que cette fonction n'effectue aucune conversion, il est possible que certains état interne du flût soient modifiés.

Note

Vous devez explicitement comparer la valeur de retour avec une des constantes, comme décrit dans [php_stream_cast](#).

Voir aussi [php_stream_cast](#) et [php_stream_is](#).

9.1.4.18 php_stream_is_persistent() : Détermine si un flût est un flût persistant

```
int php_stream_is_persistent ( php_stream * stream )
```

[php_stream_is_persistent](#) retourne 1 si le flût `stream` est un flût persistant, et 0 sinon.

9.1.4.19 php_stream_is() : Détermine si un flût est d'un type particulier

```
int php_stream_is ( php_stream * stream , int istype )
```

[php_stream_is](#) retourne 1 si `stream` est du type spécifié par `istype`, et 0 sinon.

Valeur	Signification
PHP_STREAM_IS_STDIO	Le flût est implémenté en utilisant l'implémentation standard
PHP_STREAM_IS_SOCKET	Le flût est implémenté en utilisant l'implémentation des sockets réseau
PHP_STREAM_IS_USERSPACE	Le flût est implémenté en utilisant l'implémentation de l'objet utilisateur
PHP_STREAM_IS_MEMORY	Le flût est implémenté en utilisant l'implémentation mémoire à taille automatique

Note

Les constantes PHP_STREAM_IS_XXX sont en fait définies comme des pointeurs sur la structure des opérations de base. Si vous extension (ou une autre extension définit des flûts additionnels), il faut qu'elle déclare aussi les constantes PHP_STREAM_IS_XXX dans ses en-têtes, pour que vous puissiez les utiliser comme base de comparaison.

Note

Cette fonction est implémentée comme une comparaison de pointeurs simple (et rapide), et ne modifie pas l'état du flût.

Voir aussi [php_stream_cast](#) et [php_stream_can_cast](#).

9.1.4.20 `php_stream_passthru()` : Affiche toutes les données restantes d'un flût

```
size_t php_stream_passthru ( php_stream * stream )
```

[php_stream_passthru](#) affiche tous les octets restant dans le flût `stream` vers le buffer de sortie. Les données sont écrites directement dans le buffer de sortie, qui sera le navigateur dans le cas de PHP sur un serveur web, ou la sortie standard dans le cas de PHP en mode de ligne de commande. Cette fonction essaie d'utiliser des fichiers en mémoire pour accélérer l'exécution.

9.1.4.21 `php_register_url_stream_wrapper()` : Enregistre un gestionnaire pour l'API de flûts

```
int php_register_url_stream_wrapper ( char * protocol , php_stream_wrapper * wrapper , TSRMLS_DC )
```

[php_register_url_stream_wrapper](#) enregistre le gestionnaire `wrapper` pour le protocole `protocol`.

Note

Si vous appelez cette fonction dans un module dynamiquement chargé, vous DEVEZ appeler la fonction [php_unregister_url_stream_wrapper](#) dans la fonction de terminaison de votre module, sinon PHP va crasher.

9.1.4.22 `php_unregister_url_stream_wrapper()` : Désenregistre un gestionnaire pour l'API de flûts

```
int php_unregister_url_stream_wrapper ( char * protocol , TSRMLS_DC )
```

[php_unregister_url_stream_wrapper](#) désenregistre le gestionnaire associé avec le protocole `protocol`.

9.1.4.23 `php_stream_open_wrapper_ex()` : Ouvre un flût avec un fichier ou une URL, en spécifiant le contexte

```
php_stream * php_stream_open_wrapper_ex ( char * path , char * mode , int options , char ** opened ,  
php_stream_context * context )
```

[php_stream_open_wrapper_ex](#) est exactement comme [php_stream_open_wrapper](#), mais vous permet de spécifier un objet de type `php_stream_context` avec le paramètre `context`. Pour en savoir plus sur les contextes, voyez ... une section qui viendra plus tard.

9.1.4.24 `php_stream_open_wrapper_as_file()` : Ouvre un flût avec un fichier ou une URL, et le convertit en pointeur FILE*

```
FILE * php_stream_open_wrapper_as_file ( char * path , char * mode , int options , char ** opened )
```

[php_stream_open_wrapper_as_file](#) est exactement identique à la fonction [php_stream_open_wrapper](#), mais convertit le flût en un pointeur ANSI FILE* puis le retourne à la place du flût. C'est un raccourci commode pour

les extensions qui passent des pointeurs FILE* aux librairies tierce partie.

9.1.4.25 `php_stream_filter_register_factory()` : Enregistre une usine à filtres avec l'API des flôts

```
int php_stream_filter_register_factory ( const char * filterpattern , php_stream_filter_factory * factory )
```

Utilisez cette fonction pour enregistrer une usine à filtres, dont le nom est donné par `filterpattern`. `filterpattern` peut être un nom classique, sous forme de chaîne, (i.e. `myfilter`) ou un modèle général (i.e. `myfilterclass.*`) pour permettre à un filtre d'effectuer différentes opérations en fonction du nom du filtre invoqué (i.e. `myfilterclass.foo`, `myfilterclass.bar`, etc...)

Note

Les filtres enregistrés par une extension dynamiquement chargée doivent s'assurer d'appeler `php_stream_filter_unregister_factory()` durant MSHUTDOWN.

9.1.4.26 `php_stream_filter_unregister_factory()` : Supprime une usine à filtres des API de flôts

```
int php_stream_filter_unregister_factory ( const char * filterpattern )
```

Supprime l'usine à filtres `filterfactory` spécifié par le paramètre `filterpattern`, la rendant inutilisable.

Note

Les filtres enregistrés par une extension dynamiquement chargée doivent s'assurer d'appeler `php_stream_filter_unregister_factory()` durant MSHUTDOWN.

9.1.5 Streams Dir API Reference

Les fonctions ci-dessous fonctionnent sur les dossiers locaux, mais aussi sur les dossier distants (en supposant que le gestionnaire sache le faire!).

9.1.5.1 `php_stream_opendir()` : Ouvre un dossier pour le lister

```
php_stream * php_stream_opendir ( char * path , php_stream_context * context )
```

`php_stream_opendir` retourne un flôt qui peut être utilisé pour lister les fichiers qui sont contenu dans le dossier spécifié par `path`. Cette fonction est l'équivalent de `opendir` sous POSIX. Bien que cette fonction retourne un objet `php_stream`, il n'est pas recommandé d'essayer d'utiliser les fonctions de fichiers sur ce flôt.

9.1.5.2 `php_stream_readdir()` : Lit le fichier suivant dans un dossier

```
php_stream_dirent * php_stream_readdir ( php_stream * dirstream , php_stream_dirent * ent )
```

`php_stream_readdir` lit le nom du fichier suivant dans le dossier `dirstream` et le stocke dans le paramètre `ent`. Si la fonction réussit, la valeur retournée est `ent`. Si la fonction échoue, la valeur retournée est NULL.

9.1.5.3 `php_stream_rewinddir()` : Replace le flôt de dossier au début

```
int php_stream_rewinddir ( php_stream * dirstream )
```

`php_stream_rewinddir` replace le pointeur de dossier au début du flôt. Retourne 0 en cas de succes, et -1 en cas d'échec.

9.1.5.4 `php_stream_closedir()` : Ferme un flôt de dossier, et libère les ressources

```
int php_stream_closedir ( php_stream * dirstream )
```

`php_stream_closedir` ferme un flôt le dossier `dirstream`, et libère les ressources. Retourne 0 en cas de succes, et -1 en cas d'échec.

9.1.6 API des flôts pour les fichiers

9.1.6.1 `php_stream_fopen_from_file()` : Convertit un pointeur ANSI FILE* en un flôt

```
php_stream * php_stream_fopen_from_file ( FILE * file , char * mode )
```

`php_stream_fopen_from_file` retourne un flôt basé sur le fichier `file`. `mode` doit être le même mode que celui utilisé pour ouvrir `file`, sinon d'étranges erreurs surviendront lors de l'écriture lorsque le mode du flôt sera différent de celui du fichier.

9.1.6.2 `php_stream_fopen_tmpfile()` : Ouvre un pointeur FILE* avec la fonction `tmpfile()` et la convertit en flôt

```
php_stream * php_stream_fopen_tmpfile ( void )
```

`php_stream_fopen_tmpfile` retourne un flôt basé sur un fichier temporaire, ouvert en mode "w+b". Le fichier temporaire sera supprimé automatiquement lorsque le flôt sera fermé ou que le processus sera terminé.

9.1.6.3 `php_stream_fopen_temporary_file()` : Génère un fichier temporaire et ouvre un flôt avec

```
php_stream * php_stream_fopen_temporary_file ( const char * dir , const char * pfx , char ** opened )
```

`php_stream_fopen_temporary_file` génère un fichier temporaire dans le dossier `dir` avec le préfixe `pfx`. Le nom du fichier ainsi généré est retourné dans la variable `opened`, dont vous êtes responsable de l'effacement avec `efree`. Un flôt est ouvert sur ce fichier généré, en mode "w+b". Le fichier N'EST PAS automatiquement effacé. Vous êtes responsables de son effacement lorsque vous aurez fini avec.

9.1.7 Streams Socket API Reference

9.1.7.1 php_stream_sock_open_from_socket() : Convertit un pointeur de socket en flût

```
php_stream * php_stream_sock_open_from_socket ( int socket , int persistent )
```

php_stream_sock_open_from_socket retourne un flût basé sur la socket `socket`. `persistent` est une option qui contrôle si le flût doit être ouvert comme un flût persistant. Généralement, ce paramètre vaudra 0.

9.1.7.2 php_stream_sock_open_host() : Ouvre une connexion avec un serveur, et retourne un flût

```
php_stream * php_stream_sock_open_host ( const char * host , unsigned short port , int socktype , struct timeval * timeout , int persistent )
```

php_stream_sock_open_host établit une connexion à l'hôte `host` et sur le port `port`. `socktype` spécifie les options de sockets qui devront être appliquées. `socktype` est dépendant du système, mais inclura notamment `SOCK_STREAM` pour les connexions full-duplex, séquentielle et fiables et `SOCK_DGRAM` pour les connexions non fiables, sans connexion, avec une taille maximale fixée (UDP).

`persistent` est une option qui contrôle si le flût doit être ouvert comme un flût persistant. Généralement, ce paramètre vaudra 0.

Si non-NULL, `timeout` spécifie la durée maximale de connexion. Si la connexion prend plus de temps que cette valeur, la tentative de connexion sera abandonnée et NULL sera retourné pour indiquer que le flût n'a pu être ouvert.

Note

Le paramètre `timeout` n'inclut pas le temps de résolution DNS. La raison à cela est qu'il n'y a pas de méthode portable pour effectuer des résolutions DNS.

La durée d'expiration s'applique à la phase de connexion; si vous avez besoin de durées d'expiration pour les opérations de lecture et d'écritures, il faudra utiliser la fonction php_stream_sock_set_timeout pour configurer la durée d'expiration de votre flût une fois qu'il sera ouvert.

L'API n'impose aucune restriction sur les valeurs de `socktype`, mais il est recommandé de considérer les problèmes de portabilité lorsque vous faites ce choix.

9.1.7.3 php_stream_sock_open_unix() : Ouvre une connexion avec une socket UNIX et la convertit en flût

```
php_stream * php_stream_sock_open_unix ( const char * path , int pathlen , int persistent , struct timeval * timeout )
```

php_stream_sock_open_unix tente d'ouvrir une connexion avec une socket UNIX, spécifiée par `path`. `pathlen` spécifie la taille de `path`. Si non-NULL, `timeout` spécifie la durée maximale de connexion. Si la connexion prend plus de temps que cette valeur, la tentative de connexion sera abandonnée et NULL sera retourné pour indiquer que le flût n'a pu être ouvert. `persistent` est une option qui contrôle si le flût doit être ouvert comme un flût persistant. Généralement, ce paramètre vaudra 0.

Note

Cette fonction n'est pas utilisable sous Windows, qui n'implémente pas les sockets de domaines. Une exception possible à cette règle sont les exécutables PHP compilé avec cygwin. Vous êtes encouragés à considérer cet aspect de la portabilité de votre extension avant sa publication.

Note

Cette fonction traite ^{path} en tenant compte des données binaires, ce qui est bon pour les systèmes qui utilisent un espace de nom abstrait (comme par exemple Linux), où le premier caractère est le caractère NUL.

9.1.8 Streams Structures

9.1.8.1 struct php_stream_statbuf() : Conserve des information sur le fichier ou l'URL

```
php_stream_statbuf struct stat sb
```

^{sb} est une structure standard, définie par le système de statistiques fichier.

9.1.8.2 struct php_stream_dirent() : Conserve des informations sur un fichier durant une analyse de dossier

```
php_stream_dirent char d_name[MAXPATHLEN]
```

^{d_name} contient le nom du fichier, relativement au dossier actuellement analysé.

9.1.8.3 struct php_stream_ops() : Conserve les fonctions membres d'une implémentation de flôt

```
typedef struct _php_stream_ops {
    /* Tous les flôts DOIVENT implémenter ces opérations */
    size_t (*write)(php_stream *stream, const char *buf, size_t count TSRMLS_DC);
    size_t (*read)(php_stream *stream, char *buf, size_t count TSRMLS_DC);
    int (*close)(php_stream *stream, int close_handle TSRMLS_DC);
    int (*flush)(php_stream *stream TSRMLS_DC);

    const char *label; /* name describing this class of stream */

    /* Ces opérations sont optionnelles, et peuvent prendre la valeur de
    NULL si le flôt ne supporte pas cette opération */
    int (*seek)(php_stream *stream, off_t offset, int whence TSRMLS_DC);
    char *(*gets)(php_stream *stream, char *buf, size_t size TSRMLS_DC);
    int (*cast)(php_stream *stream, int castas, void **ret TSRMLS_DC);
    int (*stat)(php_stream *stream, php_stream_statbuf *ssb TSRMLS_DC);
} php_stream_ops;
```

9.1.8.4 struct php_stream_wrapper() : Conserve les propriétés et pointeur d'un gestionnaire de flôt

```
struct _php_stream_wrapper {
    php_stream_wrapper_ops *wops; /* operations the wrapper can perform */
    void *abstract; /* context for the wrapper */
    int is_url; /* so that PG(allow_url_fopen) can be respected */

    /* support for wrappers to return (multiple) error messages to the stream opener */
    int err_count;
    char **err_stack;
} php_stream_wrapper;
```

9.1.8.5 struct php_stream_wrapper_ops() : Conserve les fonctions d'un gestionnaire de flôt

```
typedef struct _php_stream_wrapper_ops {
    /* open/create a wrapped stream */
    php_stream *(*stream_opener)(php_stream_wrapper *wrapper, char *filename, char *mode,
```

```

        int options, char **opened_path, php_stream_context *context STREAMS_DC TSRMLS_DC);
/* close/destroy a wrapped stream */
int (*stream_closer)(php_stream_wrapper *wrapper, php_stream *stream TSRMLS_DC);
/* stat a wrapped stream */
int (*stream_stat)(php_stream_wrapper *wrapper, php_stream *stream, php_stream_statbuf *ssb TSRMLS$
/* stat a URL */
int (*url_stat)(php_stream_wrapper *wrapper, char *url, php_stream_statbuf *ssb TSRMLS_DC);
/* open a "directory" stream */
php_stream *(*dir_opener)(php_stream_wrapper *wrapper, char *filename, char *mode,
        int options, char **opened_path, php_stream_context *context STREAMS_DC TSRMLS_DC);

const char *label;

/* Delete/Unlink a file */
int (*unlink)(php_stream_wrapper *wrapper, char *url, int options, php_stream_context *context TSRMLS_DC);
} php_stream_wrapper_ops;

```

9.1.8.6 struct php_stream_filter() : Conserve les propriétés et le pointeur de filtre

```

struct _php_stream_filter {
    php_stream_filter_ops *fops;
    void *abstract; /* for use by filter implementation */
    php_stream_filter *next;
    php_stream_filter *prev;
    int is_persistent;

    /* link into stream and chain */
    php_stream_filter_chain *chain;

    /* buffered buckets */
    php_stream_bucket_brigade buffer;
} php_stream_filter;

```

9.1.8.7 struct php_stream_filter_ops() : Conserver les fonctions membres pour un filtre de flût

```

typedef struct _php_stream_filter_ops {
    php_stream_filter_status_t (*filter)(
        php_stream *stream,
        php_stream_filter *thisfilter,
        php_stream_bucket_brigade *buckets_in,
        php_stream_bucket_brigade *buckets_out,
        size_t *bytes_consumed,
        int flags
        TSRMLS_DC);

    void (*dtor)(php_stream_filter *thisfilter TSRMLS_DC);

    const char *label;
} php_stream_filter_ops;

```

9.1.9 Constantes de flûts

9.1.9.1 options d'ouverture de flût() : Affectent les fonctions de flût

Plusieurs valeurs peuvent être combinées avec l'opérateur OR.

IGNORE_PATH

C'est l'option par défaut pour les flûts. Elle indique que l'include_path ne sera pas fouillé pour trouver le fichier demandé.

USE_PATH

Requiert que l'include_path soit fouillé à la recherche du fichier demandé.

IGNORE_URL

Requiert que les gestionnaires d'URL soient ignorés pour traiter ce flût. Les gestionnaires qui ne traitent pas d'URL seront utilisés lors du décodage du chemin. Il n'y a pas de contraire à cette option : l'API des flûts utilisera tous les gestionnaires

par défaut.

IGNORE_URL_WIN

Sous Windows, c'est l'équivalent de IGNORE_URL. Sur tous les autres systèmes, cette option n'a aucun effet.

ENFORCE_SAFE_MODE

Requiert que le gestionnaire de flôt effectue les vérifications liées au safe_mode sur le fichier à ouvrir avant de le faire. En omettant cette option, le safe_mode sera ignoré et n'importe quel fichier sera ouvert par PHP, du moment qu'il a les droits pour le faire.

REPORT_ERRORS

Si cette option est utilisée, et qu'il y a une erreur durant l'ouverture du fichier ou de l'URL, l'API va appeler automatiquement la fonction php_error pour vous. C'est pratique car le chemin peut contenir des informations le login et mot de passe, que vous souhaitez ou pas afficher (les afficher serait un trou de sécurité). Lorsque les API des flôts génèrent une erreur, elles commencent par supprimer les informations de login et mot de passe, afin que l'affichage des données soit sans souci.

STREAM_MUST_SEEK

Cette option est pratique lorsque votre extension doit réellement être capable déplacer le pointeur de fichier dans le flôt. Certains flots ne disposent pas d'un tel pointeur. Cette option s'assure que le flôt va supporter ce pointeur. Si il ne le fait pas, le flôt sera copié dans un fichier temporaire (qui peut être un fichier ou un bloc de mémoire), qui supportera le pointeur. Notez bien que cette option est inutile lorsque vous voulez déplacer le pointeur de fichier et y écrire, car le flôt que vous allez utiliser n'est peut être pas votre ressource initiale, mais un fichier en cache.

Note

Si la ressource demandée est sur le réseau, cette option va bloquer la fonction d'ouverture jusqu'à ce que le contenu complet du fichier soit téléchargé.

STREAM_WILL_CAST

Si votre extension utilise une librairie tierce qui s'attend à recevoir un pointeur FILE* ou un pointeur de fichier, vous pouvez utiliser cette option pour demander à l'API d'ouvrir la ressource, mais d'éviter d'utiliser les buffers. Vous pouvez alors utiliser [php_stream_cast](#) pour obtenir un FILE* ou un pointeur de fichier, que vous transmettez à la librairie.

Ceci est particulièrement pratique lorsque vous accédez à des URL via HTTP, qui vont diffuser des données, mais dont le corps ne sera accessible qu'après une quantité indéterminée d'en-têtes.

Comme cette option désactive les buffers, vous pouvez rencontrer des performances moins bonnes qu'avec les API habituelles. C'est parfaitement compréhensible, car vous avez indiqué aux flôts que vous alliez utiliser les fonctions de bases de l'implémentation. Utilisez donc cette option uniquement si vous en avez besoin.

10 FAQ : Foire Aux Questions

10.1 General Information

This section holds the most general questions about PHP: what it is and what it does.

10.1.1 What is PHP?

From the [preface of the manual](#) :

PHP is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

A nice introduction to PHP by Stig Sæther Bakken can be found [here](#) on the Zend website. Also, much of the [PHP Conference Material](#) is freely available.

10.1.2 What does PHP stand for?

PHP stands for **PHP: Hypertext Preprocessor** . This confuses many people because the first word of the acronym is the acronym. This type of acronym is called a recursive acronym. The curious can visit [Free On-Line Dictionary of Computing](#) for more information on recursive acronyms.

10.1.3 What is the relation between the versions?

PHP/FI 2.0 is an early and no longer supported version of PHP. PHP 3 is the successor to PHP/FI 2.0 and is a lot nicer. PHP 4 is the current generation of PHP, which uses the [Zend engine](#) under the hood. PHP 5 uses [Zend engine 2](#) which, among other things, offers many additional [OOP](#) features. PHP 5 is experimental.

10.1.4 Can I run several versions of PHP at the same time?

Yes. See the ^{INSTALL} file that is included in the PHP 4 source distribution. Also, read the related [appendix](#) .

10.1.5 What are the differences between PHP 3 and PHP 4?

There are [a couple of articles](#) written on this by the authors of PHP 4. Here's a list of some of the more important new features:

- Extended API module
- Generalized build process under UNIX
- Generic web server interface that also supports multi-threaded web servers
- Improved syntax highlighter
- Native HTTP session support
- Output buffering support

- More powerful configuration system
- Reference counting

Please see the [What's new in PHP 4 overview](#) for a detailed explanation of these features and more. If you're migrating from PHP 3 to PHP 4, also read the related [appendix](#) .

10.1.6 I think I found a bug! Who should I tell?

You should go to the PHP Bug Database and make sure the bug isn't a known bug. If you don't see it in the database, use the reporting form to report the bug. It is important to use the bug database instead of just sending an email to one of the mailing lists because the bug will have a tracking number assigned and it will then be possible for you to go back later and check on the status of the bug. The bug database can be found at <http://bugs.php.net/> .

10.2 Mailing lists

This section holds questions about how to get in touch with the PHP community. The best way is the mailing lists.

10.2.1 Are there any PHP mailing lists?

Of course! There are many mailing lists for several subjects. A whole list of mailing lists can be found on our [Support](#) page.

The most general mailing list is `php-general` . To subscribe, send mail to php-general-subscribe@lists.php.net . You don't need to include anything special in the subject or body of the message. To unsubscribe, send mail to php-general-unsubscribe@lists.php.net .

You can also subscribe and unsubscribe using the web interface on our [Support](#) page.

10.2.2 Are there any other communities?

There are countless of them around the world. We have links for example to some IRC servers and foreign language mailing lists on our [Support](#) page.

10.2.3 Help! I can't seem to subscribe/unsubscribe to/from one of the mailing lists!

If you have problems subscribing to or unsubscribing from the `php-general` mailing list, it may be because the mailing list software can't figure out the correct mailing address to use. If your email address was `joeblow@example.com` , you can send your subscription request to php-general-subscribe-joeblow@example.com@lists.php.net , or your unsubscription request to php-general-unsubscribe-joeblow@example.com@lists.php.net . Use similar addresses for the other mailing lists.

10.2.4 Is there an archive of the mailing lists anywhere?

Yes, you will find a list of archive sites on the [Support](#) page. The mailing list articles are also archived as news messages. You can access the news server at <news://news.php.net/> with a news client. There is also an experimental web interface for the news server at <http://news.php.net/>

10.2.5 What can I ask the mailing list?

Since PHP is growing more and more popular by the day the traffic has increased on the php-general mailing list and as of now the list gets about 150 to 200 posts a day. Because of this it is in everyones interest that you use the list as a last resort when you have looked everywhere else.

Before you post to the list please have a look in this FAQ and the manual to see if you can find the help there. If there is nothing to be found there try out the mailing list archives (see above). If you're having problem with installing or configuring PHP please read through all included documentation and README's. If you still can't find any information that helps you out you're more than welcome to use the mailing list.

Before asking questions, you may want to read the paper on [How To Ask Questions The Smart Way](#) as this is a good idea for everyone.

10.2.6 What information should I include when posting to the mailing list?

Posts like "I can't get PHP up and running! Help me! What is wrong?" are of absolutely no use to anyone. If you're having problems getting PHP up and running you must include what operating system you are running on, what version of PHP you're trying to set up, how you got it (pre-compiled, CVS, RPMs and so on), what you have done so far, where you got stuck and the exact error message.

This goes for any other problem as well. You have to include information on what you have done, where you got stuck, what you're trying to do and, if applicable, exact error messages. If you're having problems with your source code you need to include the part of the code that isn't working. Do not include more code than necessary though! It makes the post hard to read and a lot of people might just skip it all together because of this. If you're unsure about how much information to include in the mail it's better that you include to much than to little.

Another important thing to remember is to summarize your problem on the subject line. A subject like "HELP MEEEE!!!" or "What is the problem here?" will be ignored by the majority of the readers.

And lastly, you're encouraged to read the paper on [How To Ask Questions The Smart Way](#) as this will be a great help for everyone, especially yourself.

10.3 Obtenir PHP

Cette section couvre les téléchargements 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 [Téléchargements](#), pour les binaires Linux, visitez le site de votre distribution.

10.3.3 Où puis-je obtenir les librairies dont j'ai besoin pour compiler les extensions optionelles de PHP ?

Note

Celles marquées avec un * ne sont pas des librairies 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* \(Unix\)](#) .
- [LDAP \(Unix/Win\)](#) : Netscape Directory (LDAP) SDK 1.1.
- [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 \(libtiff\)](#) :
- [ZLib \(Unix/Win32\)](#) .
- [expat XML parser \(Unix/Win32\)](#) .
- [PDFLib](#) .
- [mcrypt](#) .
- [mhash](#) .
- [t1lib](#) .
- [dmalloc](#) .
- [aspell](#) .
- [readline](#) .

10.3.4 Comment faire fonctionner ces librairies ?

Vous devrez suivre les instructions fournies avec les librairies. Quelques une d'entre-elles sont détectées automatiquement lorsque vous exécutez le script 'configure' de PHP (commela librairie 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 sur mon 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 compiler 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 Database issues

This section holds common questions about relation between PHP and databases. Yes, PHP can access virtually any database available today.

10.4.1 I heard it's possible to access Microsoft SQL Server from PHP. How?

On Windows machines, you can simply use the included ODBC support and the correct ODBC driver.

On Unix machines, you can use the Sybase-CT driver to access Microsoft SQL Servers because they are (at least mostly) protocol-compatible. Sybase has made a free version of the necessary libraries for Linux systems. For other Unix operating systems, you need to contact Sybase for the correct libraries. Also see the answer to the next question.

10.4.2 Can I access Microsoft Access databases?

Yes. You already have all the tools you need if you are running entirely under Windows 9x/Me, or NT/2000, where you can use ODBC and Microsoft's ODBC drivers for Microsoft Access databases.

If you are running PHP on a Unix box and want to talk to MS Access on a Windows box you will need Unix ODBC drivers. OpenLink Software has Unix-based ODBC drivers that can do this. There is a free pilot program where you can download an evaluation copy that doesn't expire and prices start at \$675 for the commercial supported version.

Another alternative is to use an SQL server that has Windows ODBC drivers and use that to store the data, which you can then access from Microsoft Access (using ODBC) and PHP (using the built in drivers), or to use an intermediary file format that Access and PHP both understand, such as flat files or dBase databases. On this point Tim Hayes from OpenLink software writes:

Using another database as an intermediary is not a good idea, when you can use ODBC from PHP straight to your database - i.e. with OpenLink's drivers. If you do need to use an intermediary file format, OpenLink have now released Virtuoso (a virtual database engine) for NT, Linux and other unix platforms. Please visit our website for a free download.

One option that has proven successful is to use MySQL and its MyODBC drivers on Windows and synchronizing the databases. Steve Lawrence writes:

- Install MySQL on your platform according to instructions with MySQL. Latest available from www.mysql.com (get it from your mirror!). No special configuration required except when you set up a database, and configure the user account, you should put % in the host field, or the host name of the Windows computer you wish to access MySQL with. Make a note of your server name, username, and password.
- Download the MyODBC for Windows driver from the MySQL site. Latest release is myodbc-2_50_19-win95.zip (NT available too, as well as source code). Install it on your Windows machine. You can test the operation with the utilities included with this program.
- Create a user or system dsn in your ODBC administrator, located in the control panel. Make up a dsn name, enter your hostname, user name, password, port, etc for you MySQL database configured in step 1.
- Install Access with a full install, this makes sure you get the proper add-ins.. at the least you will need ODBC support and the linked table manager.
- Now the fun part! Create a new access database. In the table window right click and select Link Tables, or under the file menu option, select Get External Data and then Link Tables. When the file browser box comes up, select files of type: ODBC. Select System dsn and the name of your dsn created in step 3. Select the table to link, press OK, and presto! You can now open the table and add/delete/edit data on your MySQL server! You can also build queries, import/export tables to MySQL, build forms and reports, etc.

Tips and Tricks:

- You can construct your tables in Access and export them to MySQL, then link them back in. That makes table creation quick.
- When creating tables in Access, you must have a primary key defined in order to have write access to the table in access. Make sure you create a primary key in MySQL before linking in access
- If you change a table in MySQL, you have to re-link it in Access. Go to tools>add-ins>linked table manager, cruise to your ODBC DSN, and select the table to re-link from there. you can also move your dsn source around there, just hit the always prompt for new location checkbox before pressing OK.

10.4.3 I upgraded to PHP 4, and now mysql keeps telling me "Warning: MySQL: Unable to save result set in ...". What's up?

Most likely what has happened is, PHP 4 was compiled with the '`--with-mysql`' option, without specifying the path to MySQL. This means PHP is using its built-in MySQL client library. If your system is running applications, such as PHP 3 as a concurrent Apache module, or `auth-mysql`, that use other versions of MySQL clients, then there is a conflict between the two differing versions of those clients.

Recompiling PHP 4, and adding the path to MySQL to the flag, '`--with-mysql=/your/path/to/mysql`' usually solves the problem.

10.4.4 PHP 5 no longer bundles MySQL client libraries, what does this mean to me? Can I still use MySQL with PHP? I try to use MySQL and get "function undefined" errors, what gives?

Yes. There will always be MySQL support in PHP of one kind or another. The only change in PHP 5 is that we are no longer bundling the client library itself. Some reasons in no particular order:

- Most systems these days already have the client library installed.
- Given the above, having multiple versions of the library can get messy. For example, if you link `mod_auth_mysql` against one version and PHP against another, and then enable both in Apache, you get a nice fat crash. Also, the bundled library didn't always play well with the installed server version. The most obvious symptom of this being disagreement over where to find the `mysql.sock` unix domain socket file.
- Maintenance was somewhat lax and it was falling further and further behind the released version.
- Future versions of the library are under the GPL and thus we don't have an upgrade path since we cannot bundle a GPL'ed library in a BSD/Apache-style licensed project. A clean break in PHP 5 seemed like the best option.

This won't actually affect that many people. UNIX users, at least the ones who know what they are doing, tend to always build PHP against their system's `libmysqlclient` library simply by doing `--with-mysql=/usr` when building PHP. Windows users may enable the extension `php_mysql.dll` inside `php.ini`. Also, copy `libmysql.dll` into the appropriate `%SYSTEMROOT%` directory, just like you do with every other bundled DLL from the `dll` directory.

10.4.5 After installing shared MySQL support, Apache dumps core as soon as `libphp4.so` is loaded. Can this be fixed?

If your MySQL libs are linked against pthreads this will happen. Check using `ldd`. If they are, grab the MySQL tarball and compile from source, or recompile from the source rpm and remove the switch in the spec file that turns on the threaded client code. Either of these suggestions will fix this. Then recompile PHP with the new MySQL libs.

10.4.6 Why do I get an error that looks something like this: "Warning: 0 is not a MySQL result index in <file> on line <x>" or "Warning: Supplied argument is not a valid MySQL result resource in <file> on line <x>"?

You are trying to use a result identifier that is 0. The 0 indicates that your query failed for some reason. You need to check for errors after submitting a query and before you attempt to use the returned result identifier. The proper way to do this is with code similar to the following:

```
$result = mysql_query("SELECT * FROM tables_priv");
if (!$result) {
    echo mysql_error();
    exit;
}
```

or

```
$result = mysql_query("SELECT * FROM tables_priv")
or die("Bad query: ".mysql_error());
```

10.5 Installation

This section holds common questions about the way to install PHP. PHP is available for almost any OS (except maybe for MacOS before OSX), and almost any web server.

To install PHP, follow the instructions in the INSTALL file located in the distribution. Windows users should also read the install.txt file. There are also some helpful hints for Windows users here .

10.5.1 Unix/Windows: Where should my

By default on UNIX it should be in `/usr/local/lib` which is `<install-path>/lib` . Most people will want to change this at compile-time with the `--with-config-file-path` flag. You would, for example, set it with something like:

```
--with-config-file-path=/etc
```

And then you would copy `php.ini-dist` from the distribution to `/etc/php.ini` and edit it to make any local changes you want.

```
--with-config-file-scan-dir=PATH
```

On Windows the default path for the `php.ini` file is the Windows directory. If you're using the Apache webserver, `php.ini` is first searched in the Apaches install directory, e.g. `c:\program files\apache group\apache` . This way you can have different `php.ini` files for different versions of Apache on the same machine.

See also the chapter about the configuration file .

10.5.2 Unix: I installed PHP, but every time I load a document, I get the message 'Document Contains No Data!' What's going on here?

This probably means that PHP is having some sort of problem and is core-dumping. Look in your server error log to see if this is the case, and then try to reproduce the problem with a small test case. If you know how to use 'gdb', it is very helpful when you can provide a backtrace with your bug report to help the developers pinpoint the problem. If you are using PHP as an Apache module try something like:

- Stop your httpd processes
- gdb httpd
- Stop your httpd processes
- > run -X -f /path/to/httpd.conf
- Then fetch the URL causing the problem with your browser
- > run -X -f /path/to/httpd.conf
- If you are getting a core dump, gdb should inform you of this now
- type: bt
- You should include your backtrace in your bug report. This should be submitted to <http://bugs.php.net/>

If your script uses the regular expression functions (`ereg` and friends), you should make sure that you compiled PHP and Apache with the same regular expression package. This should happen automatically with PHP and Apache 1.3.x

10.5.3 Unix: I installed PHP using RPMS, but Apache isn't processing the PHP pages! What's going on here?

Assuming you installed both Apache and PHP from RPM packages, you need to uncomment or add some or all of the following lines in your `httpd.conf` file:

```
# 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
```

And add:

```
AddType application/x-httpd-php3 .php3 /* for PHP 3 */
AddType application/x-httpd-php .php /* for PHP 4 */
```

... to the global properties, or to the properties of the VirtualDomain you want to have PHP support added to.

10.5.4 Unix: I installed PHP 3 using RPMS, but it doesn't compile with the database support I need! What's going on here?

Due to the way PHP 3 built, it is not easy to build a complete flexible PHP RPM. This issue is addressed in PHP 4. For PHP 3, we currently suggest you use the mechanism described in the `INSTALL.REDHAT` file in the PHP distribution. If you insist on using an RPM version of PHP 3, read on...

The RPM packagers are setting up the RPMS to install without database support to simplify installations **and** because RPMS use `/usr/` instead of the standard `/usr/local/` directory for files. You need to tell the RPM spec file which databases to support and the location of the top-level of your database server.

This example will explain the process of adding support for the popular MySQL database server, using the mod installation for Apache.

Of course all of this information can be adjusted for any database server that PHP supports. We will assume you installed MySQL and Apache completely with RPMS for this example as well.

- First remove `mod_php3` :

```
rpm -e mod_php3
```

- Then get the source rpm and `INSTALL` it, NOT `--rebuild`

```
rpm -Uvh mod_php3-3.0.5-2.src.rpm
```

- Then edit the `/usr/src/redhat/SPECS/mod_php3.spec` file

In the `%build` section add the database support you want, and the path.

For MySQL you would add

```
--with-mysql=/usr \
```

The %build section will look something like this:

```
./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
```

- Once this modification is made then build the binary rpm as follows:

```
rpm -bb /usr/src/redhat/SPECS/mod_php3.spec
```

- Then install the rpm

```
rpm -ivh /usr/src/redhat/RPMS/i386/mod_php3-3.0.5-2.i386.rpm
```

Make sure you restart Apache, and you now have PHP 3 with MySQL support using RPM's. Note that it is probably much easier to just build from the distribution tarball of PHP 3 and follow the instructions in `INSTALL.REDHAT` found in that distribution.

10.5.5 Unix: I patched Apache with the FrontPage extensions patch, and suddenly PHP stopped working. Is PHP incompatible with the Apache FrontPage extensions?

No, PHP works fine with the FrontPage extensions. The problem is that the FrontPage patch modifies several Apache structures, that PHP relies on. Recompiling PHP (using 'make clean ; make') after the FP patch is applied would solve the problem.

10.5.6 Unix/Windows: I have installed PHP, but when I try to access a PHP script file via my browser, I get a blank screen.

Do a 'view source' in the web browser and you will probably find that you can see the source code of your PHP script. This means that the web server did not send the script to PHP for interpretation. Something is wrong with the server configuration – double check the server configuration against the PHP installation instructions.

10.5.7 Unix/Windows: I have installed PHP, but when try to access a PHP script file via my browser, I get a server 500 error.

Something went wrong when the server tried to run PHP. To get to see a sensible error message, from the command line, change to the directory containing the PHP executable (`php.exe` on Windows) and run `php -i`. If PHP has any problems running, then a suitable error message will be displayed which will give you a clue as to what needs to be done next. If you get a screen full of html codes (the output of the `phpinfo` function) then PHP is working, and your problem may be related to your

server configuration which you should double check.

10.5.8 Some operating systems: I have installed PHP without errors, but when I try to start apache I get undefined symbol errors:

This has actually nothing to do with PHP, but with the MySQL client libraries. Some need `--with-zlib`, others do not. This is also covered in the MySQL FAQ.

10.5.9 Windows: I have installed PHP, but when I to access a PHP script file via my browser, I get the error:

This error message means that PHP failed to output anything at all. To get to see a sensible error message, from the command line, change to the directory containing the PHP executable (`php.exe` on Windows) and run `php -i`. If PHP has any problems running, then a suitable error message will be displayed which will give you a clue as to what needs to be done next. If you get a screen full of html codes (the output of the `phpinfo` function) then PHP is working.

Once PHP is working at the command line, try accessing the script via the browser again. If it still fails then it could be one of the following:

- File permissions on your PHP script, `php.exe`, `php4ts.dll`, `php.ini` or any PHP extensions you are trying to load are such that the anonymous internet user `ISUR_<machinename>` cannot access them.
- The script file does not exist (or possibly isn't where you think it is relative to your web root directory). Note that for IIS you can trap this error by ticking the 'check file exists' box when setting up the script mappings in the Internet Services Manager. If a script file does not exist then the server will return a 404 error instead. There is also the additional benefit that IIS will do any authentication required for you based on the NT LanMan permissions on your script file.

10.5.10 Windows: I've followed all the instructions, but still can't get PHP and IIS to work together!

Make sure any user who needs to run a PHP script has the rights to run `php.exe` ! IIS uses an anonymous user which is added at the time IIS is installed. This user needs rights to `php.exe`. Also, any authenticated user will also need rights to execute `php.exe`. And for IIS4 you need to tell it that PHP is a script engine. Also, you will want to read [this faq](#).

10.5.11 When running PHP as CGI with IIS, PWS, OmniHTTPD or Xitami, I get the following error:

You must set the `cgi.force_redirect` directive to ⁰. It defaults to ¹ so be sure the directive isn't commented out (with a `;`). Like all directives, this is set in `php.ini`.

Because the default is ¹, it's critical that you're 100% sure that the correct `php.ini` file is being read. Read [this faq](#) for details.

10.5.12 How do I know if my

To be sure your `php.ini` is being read by PHP, make a call to `phpinfo` and near the top will be a listing called `Configuration File (php.ini)`. This will tell you where PHP is looking for `php.ini` and whether or not it's being read. If just a directory `PATH` exists than it's not being read and you should put your `php.ini` in that directory. If `php.ini` is included within the `PATH` than it is being read.

If `php.ini` is being read and you're running PHP as a module, then be sure to restart your web server after making changes to `php.ini`.

10.6 Build Problems

This section gathers most common errors that occur at build time.

10.6.1 I got the latest version of PHP using the anonymous CVS service, but there's no configure script!

You have to have the GNU autoconf package installed so you can generate the configure script from `configure.in`. Just run `./buildconf` in the top-level directory after getting the sources from the CVS server. (Also, unless you run configure with the `--enable-maintainer-mode` option, the configure script will not automatically get rebuilt when the `configure.in` file is updated, so you should make sure to do that manually when you notice `configure.in` has changed. One symptom of this is finding things like `@VARIABLE@` in your Makefile after configure or `config.status` is run.)

10.6.2 I'm having problems configuring PHP to work with Apache. It says it can't find

You need to tell the configure/setup script the location of the top-level of your Apache source tree. This means that you want to specify `--with-apache=/path/to/apache` and **not** `--with-apache=/path/to/apache/src`.

10.6.3 While configuring PHP (

Be sure to read the [installation](#) instructions carefully and note that you need both flex and bison installed to compile PHP. Depending on your setup you will install bison and flex from either source or a package, such as a RPM.

10.6.4 When I try to start Apache, I get the the following message:

This error usually comes up when one compiles the Apache core program as a DSO library for shared usage. Try to reconfigure apache, making sure to use at least the following flags:

```
--enable-shared=max --enable-rule=SHARED_CORE
```

For more information, read the top-level Apache `INSTALL` file or the Apache [DSO manual page](#).

10.6.5 When I run configure, it says that it can't find the include files or library for GD, gdbm, or some other package!

You can make the configure script look for header files and libraries in non-standard locations by specifying additional flags to pass to the C preprocessor and linker, such as:

```
CPPFLAGS=-I/path/to/include LDFLAGS=-L/path/to/library ./configure
```

If you're using a csh-variant for your login shell (why?), it would be:

```
env CPPFLAGS=-I/path/to/include LDFLAGS=-L/path/to/library ./configure
```

10.6.6 When it is compiling the file

You need to update your version of Bison. You can find the latest version at <http://www.gnu.org/software/bison/bison.html>.

10.6.7 When I run

Some old versions of make that don't correctly put the compiled versions of the files in the functions directory into that same directory. Try running `cp *.o functions` and then re-running `make` to see if that helps. If it does, you should really upgrade to a recent version of GNU make.

10.6.8 When linking PHP, it complains about a number of undefined references.

Take a look at the link line and make sure that all of the appropriate libraries are being included at the end. Common ones that you might have missed are '-ldl' and any libraries required for any database support you included.

If you're linking with Apache 1.2.x, did you remember to add the appropriate information to the EXTRA_LIBS line of the Configuration file and re-run Apache's Configure script? See the INSTALL file that comes with the distribution for more information.

Some people have also reported that they had to add '-ldl' immediately following `libphp4.a` when linking with Apache.

10.6.9 I can't figure out how to build PHP with Apache 1.3.

This is actually quite easy. Follow these steps carefully:

- Grab the latest Apache 1.3 distribution from <http://www.apache.org/dist/httpd/>.
- Ungzip and untar it somewhere, for example `/usr/local/src/apache-1.3`.
- Compile PHP by first running `./configure --with-apache=/<path>/apache-1.3` (substitute <path> for the actual path to your apache-1.3 directory).
- Type `make` followed by `make install` to build PHP and copy the necessary files to the Apache distribution tree.

- Change directories into to your `<path>/apache-1.3/src` directory and edit the `Configuration` file. Add to the file: `AddModule modules/php4/libphp4.a`.
- Type: `./configure` followed by `make`.
- You should now have a PHP-enabled httpd binary!

Note: You can also use the new Apache `./configure` script. See the instructions in the `README.configure` file which is part of your Apache distribution. Also have a look at the `INSTALL` file in the PHP distribution.

10.6.10 I have followed all the steps to install the Apache module version on UNIX, and my PHP scripts show up in my browser or I am being asked to save the file.

This means that the PHP module is not getting invoked for some reason. Three things to check before asking for further help:

- Make sure that the httpd binary you are running is the actual new httpd binary you just built. To do this, try running: `/path/to/binary/httpd -l`

If you don't see `mod_php4.c` listed then you are not running the right binary. Find and install the correct binary.

- Make sure you have added the correct Mime Type to one of your `Apache .conf` files. It should be: `AddType application/x-httpd-php3 .php3` (for PHP 3)

or `AddType application/x-httpd-php .php` (for PHP 4)

Also make sure that this `AddType` line is not hidden away inside a `<Virtualhost>` or `<Directory>` block which would prevent it from applying to the location of your test script.

- Finally, the default location of the Apache configuration files changed between Apache 1.2 and Apache 1.3. You should check to make sure that the configuration file you are adding the `AddType` line to is actually being read. You can put an obvious syntax error into your `httpd.conf` file or some other obvious change that will tell you if the file is being read correctly.

10.6.11 It says to use:

Note that the `libphp4.a` file is not supposed to exist. The apache process will create it!

10.6.12 When I try to build Apache with PHP as a static module using

This is a misleading error message from Apache that has been fixed in more recent versions.

10.6.13 When I try to build PHP using

There are three things to check here. First, for some reason when Apache builds the `apxs` Perl script, it sometimes ends up getting built without the proper compiler and flags variables. Find your `apxs` script (try the command `which apxs`), it's sometimes found in `/usr/local/apache/bin/apxs` or `/usr/sbin/apxs`. Open it and check for lines similar to these:

```
my $CFG_CFLAGS_SHLIB = ' ';      # substituted via Makefile.tmpl
my $CFG_LD_SHLIB     = ' ';      # substituted via Makefile.tmpl
my $CFG_LDFLAGS_SHLIB = ' ';     # substituted via Makefile.tmpl
```

If this is what you see, you have found your problem. They may contain just spaces or other incorrect values, such as 'q()'. Change these lines to say:

```
my $CFG_CFLAGS_SHLIB = '-fpic -DSHARED_MODULE'; # substituted via Makefile.tpl
my $CFG_LD_SHLIB     = 'gcc';                  # substituted via Makefile.tpl
my $CFG_LDFLAGS_SHLIB = q(-shared);           # substituted via Makefile.tpl
```

The second possible problem should only be an issue on Red Hat 6.1 and 6.2. The apxs script Red Hat ships is broken. Look for this line:

```
my $CFG_LIBEXECDIR = 'modules';               # substituted via APACI install
```

If you see the above line, change it to this:

```
my $CFG_LIBEXECDIR = '/usr/lib/apache'; # substituted via APACI install
```

Last, if you reconfigure/reinstall Apache, add a `make clean` to the process after `./configure` and before `make`.

10.6.14 During

During the `make` portion of installation, if you encounter problems that look similar to this:

```
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
```

Your system is broken. You need to fix your `/usr/include` files by installing a `glibc-devel` package that matches your `glibc`. This has absolutely nothing to do with PHP. To prove this to yourself, try this simple test:

```
$ cat >test.c <<X
#include <sys/resource.h>
X
$ gcc -E test.c >/dev/null
```

If that spews out errors, you know your include files are messed up.

10.6.15 When compiling PHP with MySQL, configure runs fine but during

First, it's important to realize that this is a `Warning` and not a fatal error. Because this is often the last output seen during `make`, it may seem like a fatal error but it's not. Of course, if you set your compiler to die on Warnings, it will. Also keep in mind that MySQL support is enabled by default.

Note

As of PHP 4.3.2, you'll also see the following text after the build (make) completes:

```
Build complete.
(It is safe to ignore warnings about tempnam and tmpnam).
```

10.6.16 I want to upgrade my PHP. Where can I find the

Either you look at config.nice file, in the source tree of your current PHP installation or, if this is not available, you simply run a

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

script. On top of the output the `./configure` line, that was used to build this PHP installation is shown.

10.6.17 When building PHP with the GD library it either gives strange compile errors or segfaults on execution.

Make sure your GD library and PHP are linked against the same depending libraries (e.g. libpng).

10.6.18 When compiling PHP I seemingly get random errors, like it hangs. I'm using Solaris if that matters.

Using non-GNU utilities while compiling PHP may cause problems. Be sure to use GNU tools in order to be certain that compiling PHP will work. For example, on Solaris, using either the SunOS BSD-compatible or Solaris versions of `sed` will not work, but using the GNU or Sun POSIX (xpg4) versions of `sed` will work. Links: [GNU sed](#) , [GNU flex](#) , and [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 fournit plusieurs variables prédéfinies , 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;
    }
}

print "<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;
}
?>

```

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](#)

10.7.2 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 entêtes 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 entêtes. 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'authentification avec IIS j'obtiens 'No Input file specified'.

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'authentifier. Utilisez alors une balise META pour rediriger vers la page ^{PHP}, ou encore proposez un lien vers celle-ci. ^{PHP} reconnaîtra alors l'authentification 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/support/kb/articles/q160/4/22.asp> et la section du manuel concernant l'[authentification HTTP](#).

10.7.9 Mon script

Netscape est plus stricte concernant les balises HTML (comme les tables) que IE. Faire valider votre HTML généré par un validateur HTML, comme validator.w3.org, peut se révéler utile. Par exemple, une `</table>` maquante 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.

10.7.10 Comment mélanger XML et

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`.

10.7.11 Comment utiliser

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.

10.7.12 Où puis-je trouver une liste complète des variables prédéfinies que je peux utiliser dans mes scripts

Lisez la page du manuel qui concerne les variables prédéfinies vu qu'elle présente une liste partielle des variables prédéfinies 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 <u>superglobals</u>.

10.7.13 Comment puis-je générer des fichiers PDF sans utiliser les librairies payantes

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 .

10.7.14 J'essaye d'accéder à une des variables standard CGI (comme \$DOCUMENT_ROOT ou \$HTTP_REFERER) dans une fonction écrite par moi-même, et il semblerait qu'elle ne soit pas définie. Que se passe-t-il ?

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 parceque elle est comme toute autre variable et requière 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 superglobaux

10.8 PHP and HTML

PHP and HTML interact a lot: PHP can generate HTML, and HTML can pass information to PHP. Before reading these faqs, it's important you learn how to retrieve variables from outside of PHP . The manual page on this topic includes many examples as well. Pay close attention to what `register_globals` means to you too.

10.8.1 What encoding/decoding do I need when I pass a value through a form/URL?

There are several stages for which encoding is important. Assuming that you have a `string $data` , which contains the string you want to pass on in a non-encoded way, these are the relevant stages:

- HTML interpretation. In order to specify a random string, you **must** include it in double quotes, and htmlspecialchars the whole value.
- URL: A URL consists of several parts. If you want your data to be interpreted as one item, you **must** encode it with urlencode .

A hidden HTML form element

```
<?php
echo "<input type='hidden' value='" . htmlspecialchars($data) . "'>\n";
?>
```

Note

It is wrong to urlencode `$data` , because it's the browsers responsibility to urlencode the data. All popular browsers do that correctly. Note that this will happen regardless of the method (i.e., GET or POST). You'll only notice this in case of GET request though, because POST requests are usually hidden.

Data to be edited by the user

```
<?php
echo "<textarea name='mydata'>\n";
echo htmlspecialchars($data)."\n";
echo "</textarea>";
?>
```

Note

The data is shown in the browser as intended, because the browser will interpret the HTML escaped symbols.

Upon submitting, either via GET or POST, the data will be urlencoded by the browser for transferring, and directly urldecoded by PHP. So in the end, you don't need to do any urlencoding/urldecoding yourself, everything is handled automagically.

In an URL

```
<?php
echo "<a href='" . htmlspecialchars("/nextpage.php?stage=23&data=" .
urlencode($data)) . "'>\n";
?>
```

Note

In fact you are faking a HTML GET request, therefore it's necessary to manually urlencode the data.

Note

You need to htmlspecialchars the whole URL, because the URL occurs as value of an HTML-attribute. In this case, the browser will first un-htmlspecialchars the value, and then pass the URL on. PHP will understand the URL correctly, because you urlencode the data.

You'll notice that the & in the URL is replaced by &. Although most browsers will recover if you forget this, this isn't always possible. So even if your URL is not dynamic, you **need** to htmlspecialchars the URL.

10.8.2 I'm trying to use an <input type="image"> tag, but the

When submitting a form, it is possible to use an image instead of the standard submit button with a tag like:

```
<input type="image" src="image.gif" name="foo">
```

When the user clicks somewhere on the image, the accompanying form will be transmitted to the server with two additional variables: `foo.x` and `foo.y`.

Because `foo.x` and `foo.y` would make invalid variable names in PHP, they are automagically converted to `foo_x` and `foo_y`. That is, the periods are replaced with underscores. So, you'd access these variables like any other described within the section on retrieving variables from outside of PHP. For example, `$_GET['foo_x']`.

10.8.3 How do I create arrays in a HTML <form>?

To get your <form> result sent as an array to your PHP script you name the <input>, <select> or <textarea> elements like this:


```
<input name="MyArray[]">
<input name="MyArray[]">
<input name="MyArray[]">
<input name="MyArray[]">
```

Notice the square brackets after the variable name, that's what makes it an array. You can group the elements into different arrays by assigning the same name to different elements:

```
<input name="MyArray[]">
<input name="MyArray[]">
<input name="MyOtherArray[]">
<input name="MyOtherArray[]">
```

This produces two arrays, MyArray and MyOtherArray, that gets sent to the PHP script. It's also possible to assign specific keys to your arrays:

```
<input name="AnotherArray[]">
<input name="AnotherArray[]">
<input name="AnotherArray[email]">
<input name="AnotherArray[phone]">
```

The AnotherArray array will now contain the keys 0, 1, email and phone.

Note
Specifying an arrays key is optional in HTML. If you do not specify the keys, the array gets filled in the order the elements appear in the form. Our first example will contain keys 0, 1, 2 and 3.

See also [Array Functions](#) and [Variables from outside PHP](#) .

10.8.4 How do I get all the results from a select multiple HTML tag?

The select multiple tag in an HTML construct allows users to select multiple items from a list. These items are then passed to the action handler for the form. The problem is that they are all passed with the same widget name. ie.

```
<select name="var" multiple="yes">
```

Each selected option will arrive at the action handler as:

```
var=option1
var=option2
var=option3
```

Each option will overwrite the contents of the previous `$var` variable. The solution is to use PHP's "array from form element" feature. The following should be used:

```
<select name="var[]" multiple="yes">
```

This tells PHP to treat `$var` as an array and each assignment of a value to `var[]` adds an item to the array. The first item becomes `$var[0]` , the next `$var[1]` , etc. The `count` function can be used to determine how many options were selected, and the `sort` function can be used to sort the option array if necessary.

Note that if you are using JavaScript the `[]` on the element name might cause you problems when you try to refer to the element by name. Use it's numerical form element ID instead, or enclose the variable name in single quotes and use that as the index to the elements array, for example:

```
variable = documents.forms[0].elements['var[]'];
```

10.8.5 How can I pass a variable from Javascript to PHP?

Since Javascript is (usually) a client-side technology, and PHP is (usually) a server-side technology, and since HTTP is a "stateless" protocol, the two languages cannot directly share variables.

It is, however, possible to pass variables between the two. One way of accomplishing this is to generate Javascript code with PHP, and have the browser refresh itself, passing specific variables back to the PHP script. The example below shows precisely how to do this -- it allows PHP code to capture screen height and width, something that is normally only possible on the client side.

```
<?php
if (isset($_GET['width']) AND isset($_GET['height'])) {
    // output the geometry variables
    echo "Screen width is: " . $_GET['width'] . "<br />\n";
    echo "Screen height is: " . $_GET['height'] . "<br />\n";
} else {
    // pass the geometry variables
    // (preserve the original query string
    // -- post variables will need to handled differently)

    echo "<script language='javascript'>\n";
    echo "  location.href=\"${_SERVER['SCRIPT_NAME']}?${_SERVER['QUERY_STRING']}\"
        . "&width=\" + screen.width + \"&height=\" + screen.height;\n";
    echo "</script>\n";
    exit();
}
?>
```

10.9 PHP and COM

PHP can be used to access COM and DCOM objects on Win32 platforms.

10.9.1 I have built a DLL to calculate something. Is there any way to run this DLL under PHP ?

If this is a simple DLL there is no way yet to run it from PHP. If the DLL contains a COM server you may be able to access it if it implements the IDispatch interface.

10.9.2 What does 'Unsupported variant type: xxxx (0xxxxx)' mean ?

There are dozens of VARIANT types and combinations of them. Most of them are already supported but a few still have to be implemented. Arrays are not completely supported. Only single dimensional indexed only arrays can be passed between PHP and COM. If you find other types that aren't supported, please report them as a bug (if not already reported) and provide as much information as available.

10.9.3 Is it possible manipulate visual objects in PHP ?

Generally it is, but as PHP is mostly used as a web scripting language it runs in the web servers context, thus visual objects will never appear on the servers desktop. If you use PHP for application scripting e.g. in conjunction with PHP-GTK there is no limitation in accessing and manipulating visual objects through COM.

10.9.4 Can I store a COM object in a session ?

No, you can't. COM instances are treated as resources and therefore they are only available in a single script's context.

10.9.5 How can I trap COM errors ?

Currently it's not possible to trap COM errors beside the ways provided by PHP itself (@, track_errors, ..), but we are thinking of a way to implement this.

10.9.6 Can I generate DLL files from PHP scripts like i can in Perl ?

No, unfortunately there is no such tool available for PHP.

10.9.7 What does 'Unable to obtain IDispatch interface for CLSID {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx}' mean ?

This error can have multiple reasons:

- the CLSID is wrong
- the requested DLL is missing
- the requested component doesn't implement the IDispatch interface

10.9.8 How can I run COM object from remote server ?

Exactly like you run local objects. You only have to pass the IP of the remote machine as second parameter to the COM constructor.

Make sure that you have set `com.allow_dcom=true` in your `php.ini`.

10.9.9 I get 'DCOM is disabled in C:\path...\scriptname.php on line 6', what can I do ?

Edit your `php.ini` and set `com.allow_dcom=true`.

10.9.10 Is it possible to load/manipulate an ActiveX object in a page with PHP ?

This has nothing to do with PHP. ActiveX objects are loaded on client side if they are requested by the HTML document. There is no relation to the PHP script and therefore there is no direct server side interaction possible.

10.9.11 Is it possible to get a running instance of a component ?

This is possible with the help of monikers. If you want to get multiple references to the same word instance you can create that instance like shown:

```
$word = new COM("C:\docs\word.doc");
```

This will create a new instance if there is no running instance available or it will return a handle to the running instance, if available.

10.9.12 Is there a way to handle an event sent from COM object ?

Starting in PHP 4.3.0, you can define an event sink and bind it as shown in the example below. You can use `com_print_typeinfo` to have PHP generate a skeleton for the event sink class.

COM event sink example
<pre><?php class IEEEventSink { var \$terminated = false; function ProgressChange(\$progress, \$progressmax) { echo "Download progress: \$progress / \$progressmax\n"; } function DocumentComplete(&\$dom, \$url) { echo "Document \$url complete\n"; } function OnQuit() { echo "Quit!\n"; \$this->terminated = true; } } \$ie = new COM("InternetExplorer.Application"); \$sink =& new IEEEventSink(); com_event_sink(\$ie, \$sink, "DWebBrowserEvents2"); \$ie->Visible = true; \$ie->Navigate("http://www.php.net"); while(!\$sink->terminated) { com_message_pump(4000); } \$ie = null; ?></pre>

10.9.13 I'm having problems when trying to invoke a method of a COM object which exposes more than one interface. What can I do ?

The answer is as simple as unsatisfying. I don't know exactly but i think you can do nothing. If someone has specific information about this, please let me know :)

10.9.14 So PHP works with COM, how about COM+ ?

COM+ extends COM by a framework for managing components through MTS and MSMQ but there is nothing special that PHP has to support to use such components.

10.9.15 If PHP can manipulate COM objects, can we imagine to use MTS to manage components resources, in conjunction with PHP ?

PHP itself doesn't handle transactions yet. Thus if an error occurs no rollback is initiated. If you use components that support transactions you will have to implement the transaction management yourself.

10.10 PHP and other languages

PHP is the best language for web programming, but what about other languages?

10.10.1 PHP vs. ASP?

ASP is not really a language in itself, it's an acronym for Active Server Pages, the actual language used to program ASP with is Visual Basic Script or JScript. The biggest drawback of ASP is that it's a proprietary system that is natively used only on Microsoft Internet Information Server (IIS). This limits it's availability to Win32 based servers. There are a couple of projects in the works that allows ASP to run in other environments and webserver: InstantASP from Halcyon (commercial), Chili!Soft ASP from Chili!Soft (commercial). ASP is said to be a slower and more cumbersome language than PHP, less stable as well. Some of the pros of ASP is that since it primarily uses VBScript it's relatively easy to pick up the language if you're already know how to program in Visual Basic. ASP support is also enabled by default in the IIS server making it easy to get up and running. The components built in ASP are really limited, so if you need to use "advanced" features like interacting with FTP servers, you need to buy additional components.

10.10.2 Is there an ASP to PHP converter?

Yes, the server-side asp2php is the one most often referred to as well as this client-side option.

10.10.3 PHP vs. Cold Fusion?

PHP is commonly said to be faster and more efficient for complex programming tasks and trying out new ideas. PHP is generally referred to as more stable and less resource intensive as well. Cold Fusion has better error handling, database abstraction and date parsing although database abstraction is addressed in PHP 4. Another thing that is listed as one of Cold Fusion's strengths is its excellent search engine, but it has been mentioned that a search engine is not something that should be included in a web scripting language. PHP runs on almost every platform there is; Cold Fusion is only available on Win32, Solaris, Linux and HP/UX. Cold Fusion has a good IDE and is generally easier to get started with, whereas PHP initially requires more programming knowledge. Cold Fusion is designed with non-programmers in mind, while PHP is focused on programmers.

A great summary by Michael J Sheldon on this topic has been posted to the PHP mailing list. A

copy can be found [here](#) .

10.10.4 PHP vs. Perl?

The biggest advantage of PHP over Perl is that PHP was designed for scripting for the web where Perl was designed to do a lot more and can because of this get very complicated. The flexibility / complexity of Perl makes it easier to write code that another author / coder has a hard time reading. PHP has a less confusing and stricter format without losing flexibility. PHP is easier to integrate into existing HTML than Perl. PHP has pretty much all the 'good' functionality of Perl: constructs, syntax and so on, without making it as complicated as Perl can be. Perl is a very tried and true language, it's been around since the late eighties, but PHP is maturing very quickly.

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} /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} 4 a été conçu pour être aussi compatible avec les anciennes versions de ^{PHP} que possible et très peu de fonctionnalité ont changé dans le processus. Si vous n'êtes pas sûrs de la compatibilité, vous devez installer ^{PHP} 4 dans un environnement de tests et y tester vos scripts.

Lisez aussi [l'appendice de la migration approprié](#) de ce manuel.

10.12.2 Les sessions fonctionnent-elles avec

Même si le [support natif des sessions](#) n'existait pas en ^{PHP} 3, il y'a des applications troisième-partie qui ont (et encore aujourd'hui) offert les fonctionnalités des sessions. La méthode la plus commune est d'utiliser ^{PHP} [LIB](#) .

10.12.3 Fonctions incompatibles ?

Puisque 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 Questions diverses

Il y'a quelques questions inclassables. Vous les trouverez ici.

10.13.1 Où sont passés les pop-ups du site web ? Puis-je en voir le code ?

Les fenêtres de pop-up jaunes de l'ancien site étaient plutôt bien, mais étaient dûres à maintenir (puisque certaines companies semblent aimer changer les comportements de leurs navigateurs avec chaque nouvelle édition).

Tout le code de la version précédente est encore disponible dans CVS. Le fichier qui vous intéresse est la dernière version de shared.inc (il contient tout le javascript et le DHTML des popups), il est disponible [ici](#) .

10.13.2 Comment puis-je manipuler le manuel compressé 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 Windows Commander, un plugin bz2 est disponible gratuitement sur le site de [Windows 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 signifie votre langage et x le format désiré, comme 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
- appelez 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](#) .

11 Appendices

11.1 Histoire de PHP

L'évolution de ^{PHP} s'est faite en quelques années. Devenir 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 librairie de scripts Perl, dont il se servait pour noter les accès à CV en ligne. Il donna le nom de 'Personal Home Page Tools' à cette librairie. Au fur et à mesure qu'il ajoutait de nouvelles fonctionnalités, Rasmus a transformé la librairie 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 Formulaire), 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 environs 50,000 noms de domaine indiquaient qu'ils avaient installé ^{PHP}. Cela représentait environs 1% des noms de domaines sur l'Internet. Même si le nombre de contributeur 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 beta. 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 mondial.

^{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 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.

^{PHP} 4 est actuellement la dernière version de ^{PHP}. Le projet de ^{PHP} 5.0 a déjà débuté, afin d'améliorer le moteur Zend, et y intégrer des capacités objets améliorées.

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 environs 20% des noms de domaines sur Internet.

L'équipe de développement de ^{PHP} inclue 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

Le futur de ^{PHP} est essentiellement constitué par son coeur, le Zend Engine. ^{PHP} 5 inclura la nouvelle version du Zend Engine 2.0. Pour plus d'informations sur ce moteur, [voyez sa 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) est la première version de classe de base en ^{PHP}, et deviendra à terme le moyen privilégié de distribuer ^{PHP} et les modules C.

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 développeur compétents, qui travaillent à mettre en place une librairie complète, réutilisable et commune à la communauté ^{PHP} entière. PEAR inclut une vaste sélection de classes de bases pour réaliser une couche d'abstraction de bases de données, du cache, des calculs mathématiques et des transactions de commerce électronique.

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 productions. 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 bugs 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, côté client. Andrei Zmievski se rappelle 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 librairie, 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, lorsque 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 avoir 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} 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 vient 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 trompe pas est le nombre de livres dédié à ^{PHP} a évolué parallèlement.

Autant que nous le sachions, le premier livre dédié à ^{PHP} a été 'php– dynamische webauftritte professionell realisieren' – un livre en 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 deux livres couvraient ^{PHP} 3.0.

Même si ces deux 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} en espagnol, coréen, japonais et hébreux.

Clairement, ce nombre grandissant de livres, écrit par différents auteurs, publiés par différents éditeurs et leur disponibilité en différentes langues est un témoignage du succès planétaire de ^{PHP} .

11.1.4 Publications sur PHP

Autant que nous le sachions, le premier article consacré à ^{PHP} dans un magazine papier fut publié dans un magazine Français, vers la fin 1998 et couvrait ^{PHP} 3.0. Comme tous 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. Des articles sur la migration d'applications ^{PHP} vers un environnement Windows existe dans la librairie ^{MSDN} de ^{Microsoft} .

11.2 Migration de PHP 3.0 à PHP 4.0

11.2.1 Ce qui a changé en PHP 4.0

PHP 4.0 et le moteur Zend ont significativement amélioré les performances et les possibilités de PHP, tout en assurant une compatibilité ascendante maximale. Le maximum de codes existants sous PHP 3.0 fonctionneront sous PHP 4.0. La migration de votre code de PHP 3.0 vers PHP 4.0 sera beaucoup plus facile que celle de PHP/FI 2.0 vers 3.0. Un grand nombre de scripts seront prêts sans modifications, 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.

11.2.2 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 2.9.1.0.25)
- Solaris 2.5 ou plus récent
- FreeBSD (testé avec 3.2, 4.0)

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
```

11.2.3 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 Mimes 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 directives de nom de PHP pour Apache ont aussi été modifiées.

Depuis PHP 4.0, 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 Admin ne peuvent être placées que dans le fichier de configuration général (i.e., `httpd.conf`).
- Les valeurs 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). À l'inverse, les valeurs Admin peuvent modifier n'importe quelle directive PHP.

Pour rendre le processus de transition plus agréable, PHP 4.0 est distribué avec des scripts qui convertissent automatiquement vos 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
<code>~/php4/scripts/apache:# ./apconf-conv.sh /usr/local/apache/conf/httpd.conf</code>

Votre configuration originale sera sauvée 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 <code>.htaccess</code> pour PHP 4
<code>~/php4/scripts/apache:# find / -name .htaccess -exec ./aphtaccess-conv.sh {} \;</code>

De la même façon, votre vieux fichier `.htaccess` sera sauvé sous le nom `.htaccess.orig`.

Les scripts de conversion requièrent l'installation préalable de `awk`.

11.2.4 Comportement de l'analyseur

L'analyse et l'exécution sont désormais deux étapes complètement dissociées, et l'exécution intervient lorsque le code, ainsi que tous ses inclusions et pré-requis, ont été complètement analysés et validés.

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 chaîne évaluée avec `eval`.

Cela ne devrait pas perturber trop de monde, car étaler son code de cette façon est plutôt un style à éviter.

Une autre nouveauté est qu'il est plus possible de faire retourner une valeur avec un fichier requis (`require`) (mais c'est plutôt rare en PHP 3.0). Retourner une valeur avec un fichier inclus (`include`) est toujours possible.

11.2.5 Rapport d'erreur

11.2.5.1 Changement de configuration

Avec PHP 3.0, 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 4.0 dispose d'un nombre significativement plus grand de niveaux de rapport d'erreur, et l'analyseur comprend désormais les constantes, lors des modifications.

Le niveau de rapport d'erreur doit désormais être explicitement configuré en supprimant les niveaux dont vous ne voulez pas du niveau maximal, 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 en plus 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

L'utilisation des vieilles valeurs de 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 de très étranges résultats, où le script n'affiche plus rien, malgré une erreur d'analyse.

Cela a conduit à un grand nombre de rapport d'erreur dans le passé, alors que les programmeurs n'étaient tout simplement pas capables de repérer l'accolade manquante, 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.

11.2.5.2 Nouveaux messages d'erreurs

Un grand nombre de scripts PHP PHP 3.0 utilisent des structures qui doivent être considérées comme un très mauvais style, même s'il effectue bien la tâche qui lui est affectée, car ils ne sont pas robustes. PHP 4.0 affichera de nombreux messages d'erreurs dans des situations où PHP 3.0 restera coi. La solution de facilité consiste à supprimer les messages de niveau `E_NOTICE`, mais c'est une meilleure idée de corriger le code à la place.

Le cas le plus courant qui génèrera des messages d'alertes est l'utilisation de constantes sans guillemets comme index de tableaux. PHP 3.0, comme PHP 4.0, 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 4.0 vous signalera tout oubli de guillemets comme 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 4.0 vous signalera les variables ou les éléments de tableaux non initialisés.

11.2.6 Initialiseur

Les variables statiques et les membres de classes n'acceptent plus que des initialiseurs scalaires, tandis que PHP 3.0 acceptait aussi les expressions. Cela 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 viennent à l'esprit.

11.2.7 empty("")

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 3.0.

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.

11.2.8 Fonctions manquantes

Bien que PHP 4.0 dispose de nombreuses nouvelles fonctionnalités fonctions et extensions, vous vous rencontrerez des fonctions PHP 3.0 qui manquent. Un petit nombre de fonctions de base n'ont pu être portées en PHP 4.0, 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 licences.

11.2.8.1 Fonctions manquantes pour des raisons de structure

Comme PHP 4.0 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 qui ne fait pas partie de ^{PHP} 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.

11.2.8.2 Fonctions et extensions obsolètes

Les extensions Adabas et Solid n'existent plus. Elles sont intégrées dans les fonctions ODBC Unifié.

11.2.8.3 Nouveau statut pour unset

`unset`, bien que toujours disponible, a été implémenté légèrement différemment en PHP 4.0, et elle n'est plus vraiment une 'fonction'.

Cela n'a pas de conséquence directe sur le comportement de `unset`, mais utiliser cette fonction pour faire un test avec `function_exists` retournera `FALSE` comme il se doit avec les fonction bas niveau 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 fonctionne plus.

11.2.9 Extensions PHP 3.0

Les extensions écrites pour PHP 3.0 ne fonctionnent plus avec PHP 4.0, ni les exécutables, ni les codes sources. Il n'est pas difficile de porter les extensions de PHP 3.0 à 4.0 si vous avez accès aux sources originales. Une description détaillée du processus de portage ne fait pas partie de cet appendice (pour le moment).

11.2.10 Substitution de variables dans les chaînes

PHP 4.0 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 cela, 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 : `"text { $obj->member } text"` ; alors qu'en PHP 3.0, il fallait faire comme ceci : `"texte".$objet->membre." texte"`.

Cette technique rend le code beaucoup plus lisible, mais risque de poser des problèmes dans certains scripts PHP 3.0. Vous pouvez facilement traquer ce problème en recherchant les séquences `{ $` dans votre code, et en les remplaçant par `\{ $` avec votre outil de remplacement habituel.

11.2.11 Cookies

PHP 3.0 avait la mauvaise habitude d'envoyer les cookies dans l'ordre inverse de celui du code (l'ordre des appels à `setcookie`). PHP 4.0 rétablit l'ordre naturel en les envoyant dans le même ordre que vous même.

Cela peut aussi prendre à contre-pied certains programmes, mais ce comportement était tellement étrange qu'il méritait un tel traitement un jour ou l'autre, pour éviter d'autres problèmes ultérieurs.

11.2.12 Gestion des variables globales

Alors que la gestion des variables globales a été surtout étudiée pour être simple en ^{PHP} 3 et aux premières versions de ^{PHP} 4, la sécurité est devenue la priorité avec les dernières versions. Alors qu'en ^{PHP} 3, le code suivant fonctionnait sans problème, en ^{PHP} 4, il faut utiliser `unset($GLOBALS["id"]);`. Ce n'est qu'un des problèmes liés à la gestion des variables. Il faut toujours utiliser la variable `$GLOBALS` avec les nouvelles versions de ^{PHP}. 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
?>
```

11.3 Migration de PHP/FI 2.0 à PHP 3.0

11.3.1 A propos des incompatibilités en 3.0

PHP 3.0 a été entièrement réécrit. 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 des syntaxes, qu'au niveau des fonctionnalités.

De plus, l'équipe de développement PHP a essayé de nettoyer la syntaxe et les sémantiques, 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 vieux script PHP/FI 2.0 existe. Il est disponible dans le dossier de convertisseur de la distribution PHP 3.0. Ce programme ne fait que repérer les modifications de syntaxe et ne vous épargnera pas une relecture attentive du script.

11.3.2 `old_function`

L'instruction `old_function` vous permet de déclarer une fonction en utilisant une syntaxe identique à celle de PHP/FI2 (hormis le fait que vous devez remplacer `'function'` par `'old_function'`).

C'est une fonctionnalité obsolète, qui ne doit être utilisée que par les convertisseurs PHP/FI2->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`.

11.3.3 Balises PHP

La première chose que vous remarquerez probablement est que les balises de PHP start et end ont changé. L'ancienne forme `<? >` a été remplacée par trois nouvelles balises possibles :

Migration: Migration: balises start/end

```
<?
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>
```

11.3.4 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 changé :

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).

11.3.5 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 !

11.3.6 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:

```
<?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:

```
<?php
while ((string)$key != "") {
?>
```

11.3.7 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.

11.3.8 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.

11.3.9 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 par exemple \$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 façon

```
<?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");
?>
```

11.3.10 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.

11.4 Débugueur PHP

11.4.1 A propos du débugeur

PHP 3 inclut le support d'un serveur de débuggage.

PHP 4 n'inclut aucun support de débuggage. 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

11.4.2 Utiliser le débugeur PHP

Le débugeur PHP sert à repérer les bugs récalcitrants. Le débugeur 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 débuggage", qui peut fonctionner avec un ^{IDE} ou un éditeur programmable (tel que Emacs).

Comment paramétrer le débugeur :

1. Réservez un port ^{TCP} pour le débugeur 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} .


```

):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

```

11.5 Développement PHP

11.5.1 Créer une fonction PHP 3

11.5.1.1 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.

11.5.1.2 Arguments de fonctions

Les arguments sont toujours de type val. Ce type contient un membre de type union, qui indique le type réel d'argument. De cette façon, si votre fonction prend deux arguments, elle ressemble à ceci :

Argument de fonction de lecture

```

pval *arg1, *arg2;
if (ARG_COUNT(ht) != 2 || getParameters(ht,2,&arg1,&arg2)==FAILURE) {
    WRONG_PARAM_COUNT;
}

```

NOTE: Les arguments peuvent être passé 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`.

11.5.1.3 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

```

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;
}

```

11.5.1.4 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

11.5.1.5 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/librairies, 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 un 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.

11.5.1.6 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)`

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 3.0, 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) { /*Initialiation échouée*/ };
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 inséré 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.

11.5.1.7 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.

11.5.1.8 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. Eventuellement, 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)` – Ajoute une entrée associative avec la clé 'key' et la valeur 'l', de type double
- `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

11.5.1.9 Utiliser la liste des ressources

PHP 3.0 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

```
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;
```

Utiliser une ressource existante

```
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) {
    php3_error(E_WARNING, "resource index %d has the wrong type", resource_id->value.lval);
    RETURN_FALSE;
}
/* ...utiliser 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).

11.5.1.10 Utiliser la table des ressources persistantes.

PHP 3.0 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 `mssql.c` pour voir comment implémenter 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 créées avec `emalloc()`, `estrdup()`, etc. Au contraire, il faut utiliser les fonctions standard `malloc()`, `strdup()`, etc. La raison est fort simple : à la fin de la requête, la mémoire sera supprimée par le gestionnaire. Etant donné que les liens persistants doivent être conservés, il ne faut pas utiliser le gestionnaire de mémoire.

Lorsque vous enregistrez une ressource qui sera placée 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...). 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 utilisés en conjonction avec la liste de ressources persistantes, et donc, vous ne devez pas utiliser `efree()` non plus.

11.5.1.11 Ajouter des directives de configuration à l'exécution

De nombreuses caractéristiques de PHP 3 peuvent être configurées 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`.

11.5.2 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 :

11.5.2.1 HashTable *function_table

La table de hash dans laquelle la fonction doit être recherchée.

11.5.2.2 pval *object

Un pointeur sur un objet sur lequel la fonction est invoquée. Il devrait être à `NULL`, si on invoque une fonction globale. Si il n'est pas à `NULL` (ie, il pointe sur un objet), l'argument `function_table` est ignorée, 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, vous quelque raison, vous ne le voulez pas, envoyez une copie de l'objet à la place.

11.5.2.3 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.

11.5.2.4 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.

11.5.2.5 int param_count

Le nombre de paramètre passé à la fonction.

11.5.2.6 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.

11.5.3 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. Tous les arguments suivants sont des paramètres de formats de chaîne. Les niveaux d'erreurs sont :

11.5.3.1 E_NOTICE

Les notes ne sont pas affichées par défaut, et indique 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.

11.5.3.2 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.

11.5.3.3 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.

11.5.3.4 E_PARSE

Les erreurs d'analyse de doivent être générées que par l'analyseur. Elles ne sont citées ici que dans le but d'être exhaustif.

11.5.3.5 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.

11.5.3.6 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.

11.5.3.7 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.

11.5.3.8 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.

11.5.3.9 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.

11.5.3.10 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.

11.5.3.11 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.

11.5.3.12 E_ALL

Toutes les erreurs ci-dessus. Utiliser ce niveau de rapport d'erreur provoque l'affichage de toutes les erreurs.

11.6 Liste d'alias

Voici la liste des alias. Tous les alias sont listés ci-dessous. 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 destinées à ceux qui doivent mettre à jour leur 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 [le site de Zend](#).

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>	Base syntax
<code>close</code>	<code>closedir</code>	Base syntax
<code>com_get</code>	<code>com_propget</code>	<code>COM</code>
<code>com_propset</code>	<code>com_propput</code>	<code>COM</code>
<code>com_set</code>	<code>com_propput</code>	<code>COM</code>
<code>cv_add</code>	<code>ccvs add</code>	<code>CCVS</code>
<code>cv_auth</code>	<code>ccvs auth</code>	<code>CCVS</code>
<code>cv_command</code>	<code>ccvs command</code>	<code>CCVS</code>
<code>cv_count</code>	<code>ccvs count</code>	<code>CCVS</code>

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)
mysql	mysql db query	MySQL
mysql_createdb	mysql create db	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_fielddata	mysql field table	MySQL
mysql_fielddtype	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_regcase	sql regcase	MySQL
mysql_selectdb	mysql select db	MySQL
mysql_tablename	mysql result	MySQL
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

Manuel PHP 4.3.0 – Version Française

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 multicolor	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	ociwriteblobtofile	OCI8
odbc_do	odbc exec	OCI8
odbc_field_precision	odbc field len	OCI8
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)
unlink	domxml unlink node	DOM XML
xptr_new_context	xpath new context	DOM XML

11.7 Mots réservés en PHP

Cette annexe est une liste d'identifiants prédéfinis 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 prédéfinies. Ces listes ne sont pas complètes ou exhaustives.

11.7.1 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 ou fonctions. Vous pouvez les utiliser comme noms de variables, mais cela risque de vous mener des confusions.

<u>and</u>	<u>or</u>	<u>xor</u>	<u>FILE</u>	<u>exception</u>	<u>php_user_filter</u>
<u>LINE</u>	<u>array</u>	<u>as</u>	<u>break</u>	<u>case</u>	
<u>cfunction</u>	<u>class</u>	<u>const</u>	<u>continue</u>	<u>declare</u>	
<u>default</u>	<u>die</u>	<u>do</u>	<u>echo</u>	<u>else</u>	
<u>elseif</u>	<u>empty</u>	<u>enddeclare</u>	<u>endfor</u>	<u>endforeach</u>	
<u>endif</u>	<u>endswitch</u>	<u>endwhile</u>	<u>eval</u>	<u>exit</u>	
<u>extends</u>	<u>for</u>	<u>foreach</u>	<u>function</u>	<u>global</u>	
<u>if</u>	<u>include</u>	<u>include_once</u>	<u>isset</u>	<u>list</u>	
<u>new</u>	<u>old_function</u>	<u>print</u>	<u>require</u>	<u>require_once</u>	
<u>return</u>	<u>static</u>	<u>switch</u>	<u>unset</u>	<u>use</u>	
<u>var</u>	<u>while</u>	<u>FUNCTION</u>	<u>CLASS</u>	<u>METHOD</u>	

11.7.2 Variables prédéfinies

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 prédéfinis `$HTTP_*_VARS`. Depuis PHP 5.0.0, les tableaux prédéfinis PHP peuvent être désactivés avec l'option de configuration register_long_arrays.

11.7.2.1 Variables de serveur : `$_SERVER`

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait <code>\$HTTP_SERVER_VARS</code> .

`$_SERVER` is an array containing information such as headers, paths, and script locations. The entries in this array are created by the webserver. There is no guarantee that every webserver will provide any of these; servers may omit some, or provide others not listed here. That said, a large number of these variables are accounted for in the CGI 1.1 specification, so you should be able to expect those.

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.

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 `$_SERVER` et `$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.

You may or may not find any of the following elements in `$_SERVER`. Note that few, if any, of these will be available (or indeed have any meaning) if running PHP on the command line.

• `PHP_SELF` •

Le nom du fichier du script en cours d'exécution, par rapport au document root. 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`.

Si PHP fonctionne en ligne de commande, cette variable n'est 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 '`.

`, 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.

`, 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 jusqu'au script courant.

`$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 chemin traduction virtuel->réel.

`$SCRIPT_NAME`

Contient le nom du script courant. Cela sert lorsque les pages doivent s'appeler elles-mêmes.

`$REQUEST_URI`

L'URI qui a été fourni pour accéder à cette page. Par exemple : `"/index.html"`.

11.7.2.2 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 valeur sont fournies par le shell qui exécute PHP, et différents systèmes pouvant disposer de différents shell, même un début de liste serait ici impossible. Reportez vous à la documentation de votre shell pour connaître une liste de variables prédéfinies.

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.

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 `$_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.

11.7.2.3 HTTP Cookies: `$_COOKIE`

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait <code>\$HTTP_COOKIE_VARS</code> .

Un tableau associatif des valeurs passées au script courant via les cookies HTTP. Elle est automatiquement global 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.

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.

11.7.2.4 HTTP GET variables: `$_GET`

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait <code>\$HTTP_GET_VARS</code> .

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.

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 `$_GET` et `$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.

11.7.2.5 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.

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 `$_POST` et `$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.

11.7.2.6 Variable de téléchargement de fichier via HTTP : `$_FILES`

Note

Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait `$HTTP_POST_FILES`.

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.

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 `$_FILES` et `$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.

11.7.2.7 Variables de requête : `$_REQUEST`

Note

Introduite en 4.1.0. Dans les versions plus anciennes, elle n'avait pas d'équivalent.

Un tableau associatif constitué du contenu des variables `$_GET`, `$_POST`, `$_COOKIE`, et `$_FILES`.

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 accessibles 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.

11.7.2.8 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.

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 `$_SESSION` et `$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.

11.7.2.9 Variables globales: \$GLOBALS

Note
<code>\$GLOBALS</code> 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.

11.7.2.10 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).

11.7.3 Classes prédéfinies

11.7.3.1 Classes standards

Ces classes sont définies dans le jeu de classe standard de PHP, inclus dans toute version de PHP.

`Directory`
La classe qui permet d'instancier `dir`.
`stdClass`
`__PHP_Incomplete_Class`

11.7.3.2 Classes prédéfinies en PHP 5

Ces classe supplémentaires ont été ajoutées en ^{PHP} 5.0.0.

`exception`
`php_user_filter`

11.7.3.3 Classes définies par Ming

Ces classes sont définies dans l'extension Ming et ne seront pas disponibles si PHP a été compilé pour être exécuté dynamiquement, ou bien si elles n'ont pas été compilées dans PHP.

```
swfshape
swffill
swfgradient
swfbitmap
swftext
swftextfield
swffont
swfdisplayitem
swfmovie
swfbutton
swfaction
swfmorph
swfsprite
```

11.7.3.4 Classes définies par Oracle 8

Ces classes sont définies dans l'extension Oracle 8 et ne seront pas disponibles si PHP a été compilé pour être exécuté dynamiquement, ou bien si elles n'ont pas été compilées dans PHP.

```
OCI-Lob
OCI-Collection
```

11.7.3.5 Classes définies par qtdom

Ces classes sont définies dans l'extension qtdom et ne seront pas disponibles si PHP a été compilé pour être exécuté dynamiquement, ou bien si elles n'ont pas été compilées dans PHP.

```
QDomDocument
QDomNode
```

11.7.4 Constantes prédéfinies en PHP

11.7.4.1 constantes prédéfinies de zend() : Constantes définies dans le coeur de PHP, Zend et les modules SAPI

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)
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*)

11.7.4.2 constantes standards prédéfinies() : Constantes standards dans PHP

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 (*nombre à virgule flottante*)
 M_LOG2E (*nombre à virgule flottante*)
 M_LOG10E (*nombre à virgule flottante*)
 M_LN2 (*nombre à virgule flottante*)
 M_LN10 (*nombre à virgule flottante*)
 M_PI (*nombre à virgule flottante*)
 M_PI_2 (*nombre à virgule flottante*)
 M_PI_4 (*nombre à virgule flottante*)

M_1_PI (*nombre à virgule flottante*)
M_2_PI (*nombre à virgule flottante*)
M_2_SQRTPI (*nombre à virgule flottante*)
M_SQRT2 (*nombre à virgule flottante*)
M_SQRT1_2 (*nombre à virgule flottante*)
CRYPT_SALT_LENGTH (*entier*)
CRYPT_STD_DES (*entier*)
CRYPT_EXT_DES (*entier*)
CRYPT_MD5 (*entier*)
CRYPT_BLOWFISH (*entier*)
DIRECTORY_SEPARATOR (*entier*)
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*)
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*)

11.8 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	Définition
aspell	aspell_new	aspell_check , aspell_check_raw , aspell_suggest	None	Dictionnaire Aspell
bzip2	bzopen	bzerrno , bzerror , bzerrstr , bzflush , bzread , bzwrite	bzclose	Fichier bzip2
COM	com_load	com_invoke , com_propget , com_get , com_propput , com_set , com_propput	None	Référence sur un objet COM
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_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_ipeg , cpdf_place_inline_image , cpdf_add_annotation	cpdf_close	Document PDF avec la librairie CPDF
cpdf outline				
curl	curl_init	curl_init , curl_exec	curl_close	Session Curl
dbm	dbmopen	dbmexists , dbmfetch , dbminsert , dbmreplace , dbmdelete , dbmfirstkey , dbmnextkey	dbmclose	Lien vers une base de données DBM
dba	dba_popen	dba_delete , dba_exists , dba_fetch , dba_firstkey , dba_insert , dba_nextkey , dba_optimize , dba_replace , dba_sync	dba_close	Lien vers une base de données DBA
dba persistent	dba_open	dba_delete , dba_exists , dba_fetch , dba_firstkey , dba_insert , dba_nextkey , dba_optimize , dba_replace , dba_sync	None	Lien persistant vers une base de données DBA
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	Lien vers une base de données dbase
dbx_link_object	dbx_connect	dbx_query	dbx_close	Connexion dbx
dbx_result_object	dbx_query		None	Résultat dbx
domxml attribute				
domxml document				
domxml node				
xpath context				
xpath object				
fbsql database	fbsql_select_db		None	Base de données fbsql
fbsql link	fbsql_change_user , fbsql_connect	fbsql_autocommit , fbsql_change_user , fbsql_create_db , fbsql_data_seek , fbsql_db_query , fbsql_drop_db , fbsql_select_db , fbsql_errno , fbsql_error , fbsql_insert_id , fbsql_list_dbs	fbsql_close	Lien vers une base de données fbsql
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_errno , fbsql_error , fbsql_insert_id , fbsql_list_dbs	None	Lien persistant vers une base de données fbsql
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	Résultat fbsql
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	Fichier FDF
ftp	ftp_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_quit	Connexion FTP

Manuel PHP 4.3.0 – Version Française

		ftp size , ftp mdtm , ftp rename , ftp delete , ftp site		
gd	imagecreate , imagecreatefromgif , imagecreatefromjpeg , imagecreatefrompng , imagecreatefromwbmp , imagecreatefromstring , imagecreatetruecolor	imagearc , imagechar , imagecharup , imagecolorallocate , imagecolorat , imagecolorclosest , imagecolorexact , imagecolorresolve , imagegammaadjust , imagegammacorrect , imagecolorset , imagecolorsforindex , imagecolorstotal , imagecolortransparent , imagecopy , imagecopyresized , imagedashedline , imagefill , imagefilledpolygon , imagefilledrectangle , imagefilltoborder , imagegif , imagepng , imagejpeg , imagebmp , imageinterlace , imageline , imagepolygon , imagepextstext , imagerectangle , imagesetpixel , imagestring , imagestringup , imagesx , imagesy , imagefttext , imagefiledarc , imageellipse , imagefilledellipse , imagecolorclosealpha , imagecolorexactalpha , imagecolorresolvealpha , imagecopymerge , imagecopymergegray , imagecopyresampled , imagetruecolortopalette , imagesetbrush , imagesettile , imagesetthickness	imagedestroy	Image GD
gd font	imagemagickloadfont	imagechar , imagecharup , imagefontheight	None	Police pour GD
gd PS encoding				
gd PS font	imagepsloadfont	imagepsextext , imagepsslantfont , imagepsextendfont , imagepsencodefont , imagepsbbox	imagepsfreefont	Police PostScript pour GD
GMP integer	gmp_init	gmp_intval , gmp_strval , gmp_add , gmp_sub , gmp_mul , gmp_div_q , gmp_div_r , gmp_div_qr , gmp_div , gmp_mod , gmp_divexact , gmp_cmp , gmp_neg , gmp_abs , gmp_sign , gmp_fact , gmp_sqrt , gmp_sqrtrm , gmp_perfect_square , gmp_pow , gmp_pown , gmp_prob_prime , gmp_gcd , gmp_gcdext , gmp_invert , gmp_legendre , gmp_jacobi , gmp_random , gmp_and , gmp_or , gmp_xor , gmp_setbit , gmp_clrbit , gmp_scan0 , gmp_scan1 , gmp_popcount , gmp_hamdist	None	Nombre GMP
hyperwave document	hw_cp , hw_docbyanchor , hw_getremote , hw_getremotechildren	hw_children , hw_childrenobi , hw_getparents , hw_getparentsobi , hw_getchildcoll , hw_getchildcollobi , hw_getremote , hw_getsrcbydestobi , hw_getandlock , hw_gettext , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobi , hw_getchilddoccoll , hw_getchilddoccollobi , hw_getanchors , hw_getanchorsobi , hw_inscoll , hw_pipedocument , hw_unlock	hw_deleteobject	Objet Hyperwave
hyperwave link	hw_connect	hw_children , hw_childrenobi , hw_cp , hw_deleteobject , hw_docbyanchor , hw_docbyanchorobi , hw_errormsg , hw_edittext , hw_error , hw_getparents , hw_getparentsobi , hw_getchildcoll , hw_getchildcollobi , hw_getremote , hw_getremotechildren , hw_getsrcbydestobi , hw_getobject , hw_getandlock , hw_gettext , hw_getobjectbyquery , hw_getobjectbyqueryobi , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobi , hw_getchilddoccoll , hw_getchilddoccollobi , hw_getanchors , hw_getanchorsobi , hw_mv , hw_incollections , hw_info , hw_inscoll , hw_insdock , hw_insertdocument , hw_insertobject , hw_mapid , hw_modifyobject , hw_pipedocument , hw_unlock , hw_who , hw_getusername	hw_close , hw_free_document	Lien vers un serveur Hyperwave
hyperwave link persistent	hw_pconnect	hw_children , hw_childrenobi , hw_cp , hw_deleteobject , hw_docbyanchor , hw_docbyanchorobi , hw_errormsg , hw_edittext , hw_error , hw_getparents , hw_getparentsobi , hw_getchildcoll , hw_getchildcollobi , hw_getremote , hw_getremotechildren , hw_getsrcbydestobi , hw_getobject , hw_getandlock , hw_gettext , hw_getobjectbyquery , hw_getobjectbyqueryobi , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobi , hw_getchilddoccoll , hw_getchilddoccollobi , hw_getanchors , hw_getanchorsobi , hw_mv , hw_incollections , hw_info , hw_inscoll , hw_insdock , hw_insertdocument , hw_insertobject , hw_mapid , hw_modifyobject , hw_pipedocument , hw_unlock , hw_who , hw_getusername	None	Lien persistant vers un serveur Hyperwave
icap	icap_open	icap_fetch_event , icap_list_events , icap_store_event , icap_snooze , icap_list_alarms , icap_delete_event	icap_close	Lien vers un serveur icap
imap	imap_open	imap_append , imap_body , imap_check , imap_createmailbox , imap_delete , imap_deletemailbox , 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_msghdr , imap_search , imap_fetch_overview	imap_close	Lien vers un serveur mail (IMAP, POP3)
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	Lien vers une base de données ingresII
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 ,	None	Lien persistant vers une base de données ingresII

		ingres_autocommit		
interbase blob				
interbase link	ibase connect	ibase query , ibase prepare , ibase trans	ibase close	Lien vers une base de données Interbase
interbase link persistent	ibase pconnect	ibase query , ibase prepare , ibase trans	None	Lien persistant vers une base de données Interbase
interbase query	ibase prepare	ibase execute	ibase free query	Requête Interbase
interbase result	ibase query	ibase fetch row , ibase fetch object , ibase field info , ibase num fields	ibase free result	Résultat Interbase
interbase transaction	ibase trans	ibase commit	ibase rollback	Transaction Interbase
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	Lien vers un serveur LDAP
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	Résultat de recherche LDAP
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	Lien vers un serveur Mcal
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	Lien vers une base de données MySQL
mysql link persistent	mysql pconnect	mysql , mysql create db , mysql createdb , mysql drop db , mysql drop db , mysql select db , mysql select db	None	Lien persistant vers une base de données MySQL
mysql query	mysql query	mysql , mysql affected rows , mysql data seek , mysql dbname , mysql fetch array , mysql fetch field , mysql fetch object , mysql fetch row , mysql fieldname , mysql field seek , mysql fieldtable , mysql fieldtype , mysql fieldflags , mysql fieldlen , mysql num fields , mysql num rows , mysql numfields , mysql numrows , mysql result	mysql free result , mysql free result	Résultat MySQL
mssql link	mssql connect	mssql query , mssql select db	mssql close	Lien vers une base de données Microsoft SQL Server
mssql link persistent	mssql pconnect	mssql query , mssql select db	None	Lien persistant vers une base de données Microsoft SQL Server

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	Résultat Microsoft SQL Server
mysql link	mysql connect	mysql affected rows , mysql change user , mysql create db , mysql data seek , mysql db name , mysql db query , mysql drop db , mysql errno , mysql error , mysql insert id , mysql list dbs , mysql list fields , mysql list tables , mysql query , mysql result , mysql select db , mysql tablename , mysql get host info , mysql get proto info , mysql get server info	mysql close	Lien vers une base de données MySQL
mysql link persistent	mysql pconnect	mysql affected rows , mysql change user , mysql create db , mysql data seek , mysql db name , mysql db query , mysql drop db , mysql errno , mysql error , mysql insert id , mysql list dbs , mysql list fields , mysql list tables , mysql query , mysql result , mysql select db , mysql tablename , mysql get host info , mysql get proto info , mysql get server info	None	Lien persistant vers une base de données MySQL
mysql result	mysql db query , mysql list dbs , mysql list fields , mysql list tables , mysql query	mysql data seek , mysql db name , mysql fetch array , mysql fetch assoc , mysql fetch field , mysql fetch lengths , mysql fetch object , mysql fetch row , mysql fetch row , mysql field flags , mysql field name , mysql field len , mysql field seek , mysql field table , mysql field type , mysql num fields , mysql num rows , mysql result , mysql tablename	mysql free result	Résultat MySQL
oci8 collection				
oci8 connection	ocilogon , ociplogon , ocinlogon	ocicommit , ociserverversion , ocinewcursor , ociparse , ocierror	ocilogoff	Lien vers une base de données Oracle
oci8 descriptor				
oci8 server				
oci8 session				
oci8 statement	ocinewdescriptor	ocirollback , ocinewdescriptor , ocirowcount , ocidefinebyname , ocibindbyname , ociexecute , ocinumcols , ocireresult , ocifetch , ocifetchinto , ocifetchstatement , ocicolumnisnull , ocicolumnname , ocicolumnsize , ocicolumntype , ocistatementtype , ocierror	ocifreestatement	Curseur Oracle
odbc link	odbc connect	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 close	Lien vers une base de données ODBC
odbc link persistent	odbc connect	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	None	Lien persistant vers une base de données ODBC
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	Résultat ODBC
velocis link				
velocis result				
OpenSSL key	openssl get privatekey , openssl get publickey	openssl sign , openssl seal , openssl open , openssl verify	openssl free key	Clé de cryptage OpenSSL
OpenSSL X.509	openssl x509 read	openssl x509 parse , openssl x509 checkpurpose	openssl x509 free	Clé publique
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	Curseur oracle
oracle link	ora logon	ora do , ora error , ora errorcode , ora rollback , ora commitoff , ora commiton , ora open , ora commit	ora logoff	Lien vers une base de données oracle
oracle link persistent	ora plogon	ora do , ora error , ora errorcode , ora rollback , ora commitoff , ora commiton , ora open , ora commit	None	Lien persistant vers une base de données oracle
pdf document	pdf new	pdf add bookmark , pdf add launchlink , pdf add localink , pdf add note , pdf add pdflink , pdf add weblink , pdf arc , pdf attach file , pdf begin page , pdf circle , pdf clip , pdf closepath , pdf closepath fill stroke , pdf closepath stroke , pdf concat , pdf continue text , pdf curveto , pdf end page , pdf endpath , pdf fill , pdf fill stroke , pdf findfont , pdf get buffer , pdf get image height , pdf get image width , pdf get parameter , pdf get value , pdf lineto , pdf moveto , pdf open ccitt , pdf open file , pdf open image file , pdf place image , pdf rect , pdf restore , pdf rotate , pdf save , pdf scale , pdf setdash , pdf setflat , pdf setfont , pdf setgray , pdf setgray fill , pdf setgray stroke , pdf setlinecap , pdf setlinejoin , pdf setlinewidth , pdf setmiterlimit ,	pdf close , pdf delete	Document PDF

Manuel PHP 4.3.0 – Version Française

		pdf setpolydash , pdf setrgbcolor , pdf setrgbcolor fill , pdf setrgbcolor stroke , 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 parameter , pdf set text pos , pdf set text rendering , pdf set value , pdf set word spacing , pdf show , pdf show boxed , pdf show xy , pdf skew , pdf stringwidth , pdf stroke , pdf translate , pdf open memory image		
pdf image	pdf open image , pdf open image file , pdf open memory image	pdf get image height , pdf get image width , pdf open ccitt , pdf place image	pdf close image	Image dans un document PDF
pdf object				
pdf outline				
pgsql large object	pg getlastoid , pg loimport , pg loimport	pg loopen , pg getlastoid , pg locreate , pg loexport , pg loread , pg loreadall , pg lounlink , pg lowrite	pg loclose	Objet de grande taille PostgreSQL
pgsql link	pg connect	pg cmdtuples , pg dbname , pg end copy , pg errormessage , pg host , pg locreate , pg loexport , pg loimport , pg loopen , pg lounlink , pg options , pg port , pg put line , pg set client encoding , pg client encoding , pg trace , pg untrace , pg tty	pg close	Lien vers une base de données PostgreSQL
pgsql link persistent	pg pconnect	pg cmdtuples , pg dbname , pg end copy , pg errormessage , pg host , pg locreate , pg loexport , pg loimport , pg loopen , pg lounlink , pg options , pg port , pg put line , pg set client encoding , pg client encoding , pg trace , pg untrace , pg tty	None	Lien persistant vers une base de données PostgreSQL
pgsql result	pg exec	pg fetch array , pg fetch object , pg fieldisnull , pg fetch row , pg fieldname , pg fieldnum , pg fieldortlen , pg fieldsize , pg fieldtype , pg getlastoid , pg numfields , pg result , pg numRows	pg freeresult	Résultat PostgreSQL
pgsql string				
printer				
printer brush				
printer font				
printer pen				
pspell	pspell new , pspell new config , pspell new personal	pspell add to personal , pspell add to session , pspell check , pspell clear session , pspell config ignore , pspell config mode , pspell config personal , pspell config repl , pspell config runtogether , pspell config save repl , pspell save wordlist , pspell store replacement , pspell suggest	None	Dictionnaire Pspell
pspell config	pspell config create	pspell new config	None	Configuration Pspell
Sablotron XSLT	xslt create	xslt closelog , xslt openlog , xslt run , xslt set sax handler , xslt errno , xslt error , xslt fetch result , xslt free	xslt free	Analyseur XSLT
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	Socket
sockets i/o vector				
dir	dir	readdir , rewinddir	closedir	Dossier
file	fopen	feof , fflush , fgetc , fgetcsv , fgets , fgetss , flock , fpassthru , fputs , fwrite , fread , fseek , ftell , fstat , ftruncate , set file buffer , rewind	fclose	Fichier
pipe	popen	feof , fflush , fgetc , fgetcsv , fgets , fgetss , fpassthru , fputs , fwrite , fread	pclose	Processus
socket	fsockopen	fflush , fgetc , fgetcsv , fgets , fgetss , fpassthru , fputs , fwrite , fread	fclose	Socket
stream				
sybase-db link	sybase connect	sybase query , sybase select db	sybase close	Lien vers une base de données Sybase avec la librairie db
sybase-db link persistent	sybase pconnect	sybase query , sybase select db	None	Lien persistant vers une base de données Sybase avec la librairie db
sybase-db result	sybase 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	Résultat Sybase avec la librairie DB
sybase-ct link	sybase connect	sybase affected rows , sybase query , sybase select db	sybase close	Lien vers une base de données Sybase avec la librairie CT
sybase-ct link persistent	sybase pconnect	sybase affected rows , sybase query , sybase select db	None	Lien persistant vers une base de données Sybase avec

				la librairie CT
sybase-ct result	sybase_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	Résultat Sybase avec la librairie CT
sysvsem	sem_get	sem_acquire	sem_release	Sémaphore Système V
sysvshm	shm_attach	shm_remove , shm_put_var , shm_get_var , shm_remove_var	shm_detach	Mémoire partagée Système V
wddx	wddx_packet_start	wddx_add_vars	wddx_packet_end	Paquet wddx
xml	xml_parser_create	xml_set_object , xml_set_element_handler , xml_set_character_data_handler , xml_set_processing_instruction_handler , xml_set_default_handler , xml_set_unparsed_entity_decl_handler , xml_set_notation_decl_handler , xml_set_external_entity_ref_handler , xml_parse , xml_get_error_code , xml_get_error_string , xml_get_current_line_number , xml_get_current_column_number , xml_get_current_byte_index , xml_parse_into_struct , xml_parser_set_option , xml_parser_get_option	xml_parser_free	Analyseur syntaxique XML
zlib	gzopen	gzeof , gzgetc , gzgets , gzgetss , gzpassthru , gzputs , gzread , gzrewind , gzseek , gztell , gzwrite	gzclose	Fichier compressé Gzip
ZZIP Directory				
ZZIP Entry				

11.9 Liste des protocoles supportés

Cette section recense une liste de protocole gérant des URL, qui sont intégrés dans les fonctions ayant accès aux fichiers, dans PHP. Par exemple, les fonctions [fopen](#) et [copy](#) . Ces fonctionnalités sont compilées comme des pilotes externes, et à partir de PHP 4.3.0, vous pouvez créer vos propres pilotes, avec la fonction [stream_register_wrapper](#) .

11.9.1 HTTP et HTTPS

PHP 3, PHP 4. <https://> depuis PHP 4.3

```

• http://exemple.com
• http://utilisateur:motdepasse@exemple.com
• https://exemple.com
• https://utilisateur:motdepasse@exemple.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 flôt, il sera aussi utilisé dans votre requête.

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.

Le flôt 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.

11.9.2 FTP and FTPS

PHP 3, PHP 4. `ftp://` depuis PHP 4.3

- `ftp://example.com/pub/file.txt`
- `ftp://user:password@example.com/pub/file.txt`
- `ftps://example.com/pub/file.txt`
- `ftps://user:password@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, vous pourrez essayer de l'ouvrir en écriture, mais la connexion échouera. Si vous devez modifier un fichier qui existe déjà, utilisez `ftp_connect`.

`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.

11.9.3 Flôts 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 flôt 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://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=<flôt à filtrer>` (**obligatoire**) Ce paramètre doit être situé à la fin de votre chaîne `php://filter` et doit pointer sur le flôt à filtrer.

```
<?php
/* Ceci est l'équivalent de :
   readfile("http://www.example.com");
   puisqu'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
/* Cette ligne va afficher le contenu de
   www.exemple.com tout en majuscule */
readfile("php://filter/read=string.toupper/resource=http://www.exemple.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.exemple.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
/* 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_set_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.

11.9.4 Flôts 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 flôt 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.

11.10 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 flôts qui exploitent les sockets, tels que `fsockopen` 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` .

11.10.1 Domaines Internet : TCP, UDP, SSL et TLS

PHP 3, PHP 4. ^{ssl://} & ^{tls://} depuis PHP 4.3.

Note

Si aucun transport n'est spécifié, `tcp://` est utilisé.

- 127.0.0.1
- fe80::1
- www.exemple.com
- tcp://127.0.0.1
- tcp://fe80::1
- tcp://www.exemple.com
- udp://www.exemple.com
- ssl://www.exemple.com
- tls://www.exemple.com

Les sockets du domaine Internet utilisent un numéro de port en plus de l'adresse de l'hôte. Dans le cas de `fsockopen`, 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.exemple.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 : `[fe80::1]`. 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.

Nom	Utilisation	Valeur par défaut
<code>verify_peer</code>	TRUE ou FALSE. Requiert la vérification du certificat SSL utilisé.	FALSE
<code>allow_self_signed</code>	TRUE ou FALSE. Permet les certificats auto-signés.	FALSE
<code>cafile</code>	Situation du fichier de l'autorité de certification sur le serveur local, qui doit être utilisé avec <code>verify_peer</code> comme option de contexte pour identifier un pair distant.	
<code>capath</code>	Si <code>cafile</code> n'est pas spécifié, ou si le certificat n'est pas trouvé, le dossier <code>capath</code> est scanné pour y trouver un certificat convenable. <code>capath</code> doit être un dossier de certificats corrects.	
<code>local_cert</code>	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.	
<code>passphrase</code>	Mot de passe avec lequel votre <code>local_cert</code> a été chiffré.	
<code>CN_match</code>	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://`.

11.10.2 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 flût 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.

11.11 Comparaison de types en PHP

La table suivante résume les différents comportements 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 ^{string}, alors que 42 est un ^{integer}. FALSE est ^{boolean} alors que "false" est une ^{string}.

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 ($x)` lorsque `$x` est indéfini par générer une erreur de niveau ^{E_NOTICE}. Alternativement, utilisez plutôt les fonctions empty ou isset, ou encore, initialisez toutes vos variables.

	<u>gettype</u>	<u>empty</u>	<u>is_null</u>	<u>isset</u>	boolean : if(\$x)
<code>\$x = "";</code>	string	TRUE	FALSE	TRUE	FALSE
<code>\$x = NULL</code>	NULL	TRUE	TRUE	FALSE	FALSE
<code>var \$x;</code>	NULL	TRUE	TRUE	FALSE	FALSE
<code>\$x</code> est indéfini	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>	integer	FALSE	FALSE	TRUE	TRUE
<code>\$x = 42;</code>	integer	FALSE	FALSE	TRUE	TRUE
<code>\$x = 0;</code>	integer	TRUE	FALSE	TRUE	FALSE
<code>\$x = -1;</code>	integer	FALSE	FALSE	TRUE	TRUE
<code>\$x = "1";</code>	string	FALSE	FALSE	TRUE	TRUE
<code>\$x = "0";</code>	string	TRUE	FALSE	TRUE	FALSE
<code>\$x = "-1";</code>	string	FALSE	FALSE	TRUE	TRUE
<code>\$x = "php";</code>	string	FALSE	FALSE	TRUE	TRUE
<code>\$x = "true";</code>	string	FALSE	FALSE	TRUE	TRUE
<code>\$x = "false";</code>	string	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	FALSE	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	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
1	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	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	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
NULL	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
array()	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
"php"	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

Note

Note PHP 3.0

La chaîne de valeur "" était considéré comme non vide en ^{PHP} 3. Ce comportement a changé en ^{PHP} 4, où il est désormais considéré comme vide.

11.12 Liste des token de l'analyseur

De nombreuses parties du langage PHP sont représentées en interne par des trucs (NDT : littéralement) comme T_SR. PHP affiche des identifiants comme celui-ci dans les erreurs d'analyse, comme par exemple : `unexpected T_SR, expecting ',' or ';' in script.php on line 10.`

Vous êtes supposé savoir ce que T_SR signifie. Pour ceux qui ne le savent pas, voici une liste de ces identifiant, la syntaxe PHP attenante, et les références appropriées au manuel.

Token	Syntaxe	Référence
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) or (boolean)	transtypage
T_BREAK	break;	break
T_CASE	case	switch
T_CHARACTER		
T_CLASS	class	classes et objets
T_CLOSE_TAG	?> ou %>	
T_COMMENT	// ou #	commentaires
T_CONCAT_EQUAL	.=	opérateurs d'assignation
T_CONST	const	
T_CONSTANT_ENCAPSED_STRING	"foo" or '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_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) or (float)	transtypage
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_FOR	for	for
T_FOREACH	foreach	foreach
T_FUNCTION	function or cfunction	fonctions
T_GLOBAL	global	scope de variable
T_IF	if	if
T_INC	++	opérateurs d'incrémention/décrémention
T_INCLUDE	include()	include

T_INCLUDE_ONCE	include_once()	include_once
T_INLINE_HTML		
T_INT_CAST	(int) or (integer)	transtypage
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	!= or <>	opérateurs de comparaison
T_IS_NOT_IDENTICAL	!==	opérateurs de comparaison
T_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	/* and */	commentaires
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	<?> or <%=	sortie du mode HTML
T_OR_EQUAL	=	opérateurs d'assignation
T_PAAMAYIM_NEKUDOTAYIM	::	::
T_PLUS_EQUAL	+=	opérateurs d'assignation
T_PRINT	print()	print
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_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

11.13 A propos du manuel

11.13.1 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 ces manuels. Nous n'en recommandons aucune, car elles sont rapidement obsolètes.

Vous pouvez lire le manuel en ligne sur le site <http://www.php.net/> et sur les nombreux miroirs. Pour un confort d'utilisation accru, il est recommandé d'utiliser un miroir proche de chez vous. Vous pouvez le voir en version HTML simple (et facilement imprimable), ou bien aux couleurs de ^{PHP}.net.

Un des points forts du manuel en ligne sur la plupart des formats téléchargeable, est l'intégration des notes des utilisateurs. Un inconvénient évident est que vous devez être en ligne pour profiter de ces formats.

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'.

Le format le plus portable est le manuel au format HTML. Le manuel est fourni dans un format en une seule page HTML, ou comme un ensemble de fichier de taille réduite (mais un bon millier de fichiers tout de même). Les formats HTML et texte simple sont disponibles sous forme de fichiers d'archives compressées avec bzip2.

Un autre format populaire et portable est le format ^{PDF}, le plus pratique pour les impressions. Mais avant de lancer une telle impression, pensez bien que le manuel au complet fait plus de 2000 pages, et qu'il est constamment révisé.

Note

Si vous n'avez pas de programme capable de lire les fichiers ^{PDF} , téléchargez le ^{Array} .

Pour les propriétaires de PDA Palm, deux formats Palm document et iSilo sont optimisés pour cette plate-forme. Vous pouvez emporter votre ^{PDA} avec vous dans vos déplacements quotidiens, et utiliser les programmes ^{ARRAY} et ^{Array} pour améliorer vos connaissances de ^{PHP}, ou bien profiter de ses références rapides.

Pour les plates-formes Windows, le format ^{Windows HTML Help} fournit une version HTML du manuel, 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.

11.13.2 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.

11.13.3 Comment lire la définition d'une fonction (prototype)

Chaque fonction est documentée pour permettre un accès rapide, et savoir décoder le manuel rendra votre lecture plus facile. Plutôt que de dépendre d'exemples prêts en copier/coller, il est mieux de savoir lire la définition d'une fonction. Voici comment :

Note

Pré-requis : Connaissances de bases 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

int strlen (string str)

strlen retourne la taille de la chaîne str .

Partie	Description
strlen	Le nom de la fonction.
(PHP 3, PHP 4)	strlen() est présente dans toutes les versions de PHP 3 et PHP 4.
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 strlen 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 par exemple 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 nombre à virgule flottante), ou encore un tableau . ^{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.

11.13.4 Versions de PHP documentées dans ce manuel

Cette documentation contient des informations sur ^{PHP} 4, 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.

Vous pouvez trouvez des portions de documentation destinées aux versions CVS de ^{PHP} , ce qui signifie que les toutes dernières versions de développement de ^{PHP} proposent ces fonctionnalités. Si vous n'êtes pas un développeur ^{PHP} , et que vous n'avez pas la possibilité d'utiliser la dernière version de ^{PHP} , les fonctionnalités indiquées comme "disponibles uniquement en CVS" ne vous seront pas accessibles. Toutefois, ces fonctionnalités seront surement disponible dans la prochaine version stable de ^{PHP} . Si vous voulez obtenir une version des derniers sources de ^{PHP} , passez par le serveur CVS anonyme .

Vous pouvez aussi rencontrer de la documentation qui est disponible pour une version de ^{PHP} qui n'est pas encore publiée (par exemple, des fonctions destinées à la version ^{PHP} 4.3.0, alors que la version publique actuelle est ^{PHP} 4.2.3). La plupart du temps, ce n'est pas une erreur de documentation, mais une indication sur les fonctionnalités qui seront disponible avec la prochaine publication de ^{PHP}.

11.13.5 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 orientée 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 bloqué par un problème, vous pourrez surement trouver de l'aide auprès de ces listes. Il existe un recensement des listes de diffusion à l'URL <http://www.php.net/support.php>, ainsi qu'une liste d'archives, et de ressources de support en ligne. De plus, à <http://www.php.net/links.php>, il y a une liste de sites web dédiés à ^{PHP}, avec des articles, des forum, des bibliothèques de code.

11.13.6 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 utiliser le système de rapport de bugs : <http://bugs.php.net/>. Classez le rapport sous la catégorie "Documentation Problem". Vous pouvez aussi poster vos problèmes ici même.

Note
N'abusez pas du système de bugs pour envoyer des demandes d'aide. Utilisez les listes de diffusion ou les sites de communauté, tels que mentionnés ci-dessus.

En contribuant aux notes, vous pouvez fournir de nouveaux exemples, mettre en lumière des effets de bords ou apporter des clarifications pour les autres lecteurs. Mais ne prenez pas le système d'annotation pour un système de soumission de bugs. Vous pouvez en savoir plus sur les annotations dans la section '[A propos des notes d'utilisateurs](#)'

Si vous connaissez l'anglais et une autre langue, vous pouvez aussi contribuer en participant à la traduction. Si vous voulez démarrer une nouvelle traduction, ou participer au projet, alors commencez par lire <http://cvs.php.net/co.php/phpdoc/howto/howto.html.tar.gz>.

11.13.7 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} format du manuel, et bien sur, ^{PHP} lui-même pour réaliser certaines conversions et formats.

Vous pouvez télécharger le manuel dans différentes langues, y compris un format en texte plein, ^{HTML}, ^{PDF}, ^{PalmPilot DOC}, ^{PalmPilot iSilo} et ^{Windows HTML Help}, à l'URL <http://www.php.net/docs.php>. Les manuels sont mis à jours automatiquement, avec la mise à jour des sources.

Vous pouvez trouver plus d'informations sur le téléchargement des sources ^{XML} de cette documentation à <http://cvs.php.net/>. La documentation est stockée dans le module ^{phpdoc}.

11.13.8 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** le format ^{HTML}, le texte plein, ou encore le format ^{PDF}. C'est le système d'édition qui se charge de faire les conversions du format ^{XML} vers un format lisible.

Note

Si vous voulez démarrer une nouvelle e traduction dans votre langue, 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 pour traduire la documentation dans une nouvelle langue, et quelqu'un viendra vous aider à démarrer un nouveau site de 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), Coréen, Espagnol, Finnois, Français, 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>.

11.13.9 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 damien.seguy@nexen.net et didou@keliglia.com.

12 Index

12.1 Index des fonctions

- 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_lookup_uri](#)
 - ♦ [apache_note](#)
 - ♦ [apache_request_headers](#)
 - ♦ [apache_response_headers](#)
 - ♦ [apache_setenv](#)
 - ♦ [array](#)
 - ♦ [array_change_key_case](#)
 - ♦ [array_chunk](#)
 - ♦ [array_combine](#)
 - ♦ [array_count_values](#)
 - ♦ [array_diff](#)
 - ♦ [array_diff_assoc](#)
 - ♦ [array_fill](#)
 - ♦ [array_filter](#)
 - ♦ [array_flip](#)
 - ♦ [array_intersect](#)
 - ♦ [array_intersect_assoc](#)
 - ♦ [array_key_exists](#)
 - ♦ [array_keys](#)
 - ♦ [array_map](#)
 - ♦ [array_merge](#)
 - ♦ [array_merge_recursive](#)
 - ♦ [array_multisort](#)
 - ♦ [array_pad](#)
 - ♦ [array_pop](#)
 - ♦ [array_push](#)
 - ♦ [array_rand](#)
 - ♦ [array_reduce](#)
 - ♦ [array_reverse](#)
 - ♦ [array_search](#)
 - ♦ [array_shift](#)
 - ♦ [array_slice](#)

- ♦ [array_splice](#)
- ♦ [array_sum](#)
- ♦ [array_unique](#)
- ♦ [array_unshift](#)
- ♦ [array_values](#)
- ♦ [array_walk](#)
- ♦ [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](#)
 - ♦ [bcddiv](#)
 - ♦ [bcmod](#)
 - ♦ [bcmul](#)
 - ♦ [bcpow](#)
 - ♦ [bcpowmod](#)
 - ♦ [bcscale](#)
 - ♦ [bcsqrt](#)
 - ♦ [bcsub](#)
 - ♦ [bin2hex](#)
 - ♦ [bind_textdomain_codeset](#)
 - ♦ [bindec](#)
 - ♦ [bindtextdomain](#)
 - ♦ [bzclose](#)
 - ♦ [bzcompress](#)
 - ♦ [bzdecompress](#)
 - ♦ [bzerrno](#)
 - ♦ [bzerror](#)
 - ♦ [bzerrstr](#)
 - ♦ [bzflush](#)
 - ♦ [bzopen](#)
 - ♦ [bzread](#)
 - ♦ [bzwrite](#)
- **C**
- - ♦ [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](#)
- ◆ [clearstatcache](#)
- ◆ [closedir](#)
- ◆ [closelog](#)
- ◆ [com_addrf](#)
- ◆ [com_get](#)
- ◆ [com_invoke](#)
- ◆ [com_isenum](#)
- ◆ [com_load](#)
- ◆ [com_load_typelib](#)
- ◆ [com_propget](#)
- ◆ [com_propput](#)
- ◆ [com_propset](#)
- ◆ [com_release](#)
- ◆ [com_set](#)
- ◆ [compact](#)
- ◆ [connection_aborted](#)
- ◆ [connection_status](#)
- ◆ [connection_timeout](#)
- ◆ [constant](#)
- ◆ [convert_cyr_string](#)
- ◆ [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_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](#)
- ♦ [cyrus_authenticate](#)
- ♦ [cyrus_bind](#)
- ♦ [cyrus_close](#)
- ♦ [cyrus_connect](#)
- ♦ [cyrus_query](#)
- ♦ [cyrus_unbind](#)
- **D**
- - ♦ [date](#)
 - ♦ [dba_close](#)
 - ♦ [dba_delete](#)
 - ♦ [dba_exists](#)
 - ♦ [dba_fetch](#)
 - ♦ [dba_firstkey](#)
 - ♦ [dba_handlers](#)
 - ♦ [dba_insert](#)
 - ♦ [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_rctxexact](#)
- ◆ [dbplus_rctxlike](#)
- ◆ [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](#)
- ◆ [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](#)
- ◆ [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](#)
- ◆ [dirname](#)
- ◆ [disk_free_space](#)
- ◆ [disk_total_space](#)
- ◆ [diskfreespace](#)
- ◆ [dl](#)
- ◆ [dngettext](#)
- ◆ [dns_check_record](#)
- ◆ [dns_get_mx](#)
- ◆ [dns_get_record](#)
- ◆ [domattribute->name](#)
- ◆ [domattribute->specified](#)
- ◆ [domattribute->value](#)
- ◆ [domdocument->add_root \[obsolète\]](#)
- ◆ [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->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](#)
- ◆ [domelement->get_attribute_node](#)
- ◆ [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_attributess](#)
- ◆ [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](#)
- ◆ [domxml_xslt_stylesheet_doc](#)
- ◆ [domxml_xslt_stylesheet_file](#)
- ◆ [domxsltstylesheet->process](#)
- ◆ [domxsltstylesheet->result_dump_file](#)
- ◆ [domxsltstylesheet->result_dump_mem](#)

- ♦ [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_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_change_user](#)
 - ♦ [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_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_erro](#)
- ◆ [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_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](#)
- ◆ [floatval](#)
- ◆ [flock](#)
- ◆ [floor](#)
- ◆ [flush](#)
- ◆ [fmod](#)
- ◆ [fnmatch](#)

- ♦ [fopen](#)
- ♦ [fpassthru](#)
- ♦ [fprintf](#)
- ♦ [fputs](#)
- ♦ [fread](#)
- ♦ [frenchtojd](#)
- ♦ [fribidi_log2vis](#)
- ♦ [fscanf](#)
- ♦ [fseek](#)
- ♦ [fsockopen](#)
- ♦ [fstat](#)
- ♦ [ftell](#)
- ♦ [ftok](#)
- ♦ [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](#)
- ♦ [ftruncate](#)
- ♦ [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_defined_constants](#)
- ◆ [get_defined_functions](#)
- ◆ [get_defined_vars](#)
- ◆ [get_extension_funcs](#)
- ◆ [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_sqrtrm](#)
- ◆ [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_sent](#)
 - ♦ [hebreve](#)
 - ♦ [hebrevc](#)
 - ♦ [hexdec](#)
 - ♦ [highlight_file](#)
 - ♦ [highlight_string](#)
 - ♦ [html_entity_decode](#)
 - ♦ [htmlentities](#)
 - ♦ [htmlspecialchars](#)
 - ♦ [http_build_query](#)
 - ♦ [hw_api->checkin](#)
 - ♦ [hw_api->checkout](#)
 - ♦ [hw_api->children](#)
 - ♦ [hw_api->content](#)
 - ♦ [hw_api->content](#)
 - ♦ [hw_api->copy](#)
 - ♦ [hw_api->dbstat](#)
 - ♦ [hw_api->dcstat](#)
 - ♦ [hw_api->dstanchors](#)
 - ♦ [hw_api->dstofsrcanchors](#)
 - ♦ [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->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_identify](#)
- ◆ [hw_incollections](#)
- ◆ [hw_info](#)

- ♦ [hw_inscoll](#)
- ♦ [hw_insdock](#)
- ♦ [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_username](#)
- ♦ [hw_who](#)
- ♦ [hwapi_hgcsp](#)
- ♦ [hypot](#)
- I
- - ♦ [ibase_add_user](#)
 - ♦ [ibase_affected_rows](#)
 - ♦ [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_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_modify_user](#)
 - ♦ [ibase_name_result](#)
 - ♦ [ibase_num_fields](#)
 - ♦ [ibase_num_params](#)
 - ♦ [ibase_param_info](#)

- ◆ [ibase_pconnect](#)
- ◆ [ibase_prepare](#)
- ◆ [ibase_query](#)
- ◆ [ibase_rollback](#)
- ◆ [ibase_rollback_ret](#)
- ◆ [ibase_set_event_handler](#)
- ◆ [ibase_timefmt](#)
- ◆ [ibase_trans](#)
- ◆ [ibase_wait_event](#)
- ◆ [iconv](#)
- ◆ [iconv_get_encoding](#)
- ◆ [iconv_set_encoding](#)
- ◆ [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](#)
- ◆ [image2wbmp](#)
- ◆ [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](#)
- ◆ [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](#)
- ◆ [imagefontheight](#)
- ◆ [imagefontwidth](#)
- ◆ [imageftbbox](#)
- ◆ [imagefttext](#)
- ◆ [imagegammacorrect](#)
- ◆ [imagegd](#)
- ◆ [imagegd2](#)
- ◆ [imagegif](#)

- ◆ [imageinterlace](#)
- ◆ [imageistruecolor](#)
- ◆ [imagejpeg](#)
- ◆ [imageline](#)
- ◆ [imagerloadfont](#)
- ◆ [imagepalettecopy](#)
- ◆ [imagepng](#)
- ◆ [imagepolygon](#)
- ◆ [imagepsbbox](#)
- ◆ [imagepscopyfont](#)
- ◆ [imagepsencodefont](#)
- ◆ [imagepsextendfont](#)
- ◆ [imagepsfreefont](#)
- ◆ [imagepsloadfont](#)
- ◆ [imagepslantfont](#)
- ◆ [imagepstext](#)
- ◆ [imagerectangle](#)
- ◆ [imagerotate](#)
- ◆ [imagesavealpha](#)
- ◆ [imagesetbrush](#)
- ◆ [imagesetpixel](#)
- ◆ [imagesetstyle](#)
- ◆ [imagesetthickness](#)
- ◆ [imagesettile](#)
- ◆ [imagestring](#)
- ◆ [imagestringup](#)
- ◆ [imagesx](#)
- ◆ [imagesy](#)
- ◆ [imagetruecolortopalette](#)
- ◆ [imagettfbbox](#)
- ◆ [imagettftext](#)
- ◆ [imagetypes](#)
- ◆ [imagewbmp](#)
- ◆ [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_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_uid](#)
- ♦ [imap_undelete](#)
- ♦ [imap_unsubscribe](#)
- ♦ [imap_utf7_decode](#)
- ♦ [imap_utf7_encode](#)
- ♦ [imap_utf8](#)
- ♦ [implode](#)
- ♦ [import_request_variables](#)
- ♦ [in_array](#)
- ♦ [ingres_autocommit](#)
- ♦ [ingres_close](#)
- ♦ [ingres_commit](#)
- ♦ [ingres_connect](#)
- ♦ [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](#)
- ♦ [intval](#)
- ♦ [ip2long](#)
- ♦ [iptcembed](#)
- ♦ [iptcparse](#)
- ♦ [ircg_channel_mode](#)
- ♦ [ircg_disconnect](#)
- ♦ [ircg_fetch_error_msg](#)
- ♦ [ircg_get_username](#)
- ♦ [ircg_html_encode](#)
- ♦ [ircg_ignore_add](#)
- ♦ [ircg_ignore_del](#)
- ♦ [ircg_is_conn_alive](#)
- ♦ [ircg_join](#)
- ♦ [ircg_kick](#)
- ♦ [ircg_lookup_format_messages](#)
- ♦ [ircg_msg](#)
- ♦ [ircg_nick](#)
- ♦ [ircg_nickname_escape](#)
- ♦ [ircg_nickname_unescape](#)
- ♦ [ircg_notice](#)
- ♦ [ircg_part](#)
- ♦ [ircg_pconnect](#)
- ♦ [ircg_register_format_messages](#)
- ♦ [ircg_set_current](#)
- ♦ [ircg_set_file](#)
- ♦ [ircg_set_on_die](#)
- ♦ [ircg_topic](#)
- ♦ [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_string](#)
- ♦ [is_subclass_of](#)
- ♦ [is_uploaded_file](#)
- ♦ [is_writable](#)
- ♦ [is_writeable](#)
- ♦ [isset](#)
- **J**
- - ♦ [java_last_exception_clear](#)
 - ♦ [java_last_exception_get](#)
 - ♦ [jddayofweek](#)
 - ♦ [jdmonthname](#)
 - ♦ [jdtofrrench](#)
 - ♦ [jdtogregorian](#)
 - ♦ [jdtojewish](#)
 - ♦ [jdtojulian](#)
 - ♦ [jdtonix](#)
 - ♦ [jewishtojd](#)
 - ♦ [join](#)
 - ♦ [jpeg2wbmp](#)
 - ♦ [juliantojd](#)
- **K**
- - ♦ [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_search](#)
- ♦ [ldap_set_option](#)
- ♦ [ldap_set_rebind_proc](#)
- ♦ [ldap_sort](#)
- ♦ [ldap_start_tls](#)
- ♦ [ldap_t61_to_8859](#)
- ♦ [ldap_unbind](#)
- ♦ [levenshtein](#)
- ♦ [link](#)
- ♦ [linkinfo](#)
- ♦ [list](#)
- ♦ [localeconv](#)
- ♦ [localtime](#)
- ♦ [log](#)
- ♦ [log10](#)
- ♦ [log1p](#)
- ♦ [long2ip](#)
- ♦ [lstat](#)
- ♦ [ltrim](#)
- **M**
- ♦ [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](#)
- ◆ [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_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](#)

- ◆ [mcaldeldelete_calendar](#)
- ◆ [mcaldeldelete_event](#)
- ◆ [mcaldel_event_add_attribute](#)
- ◆ [mcaldel_event_init](#)
- ◆ [mcaldel_event_set_alarm](#)
- ◆ [mcaldel_event_set_category](#)
- ◆ [mcaldel_event_set_class](#)
- ◆ [mcaldel_event_set_description](#)
- ◆ [mcaldel_event_set_end](#)
- ◆ [mcaldel_event_set_recur_daily](#)
- ◆ [mcaldel_event_set_recur_monthly_mday](#)
- ◆ [mcaldel_event_set_recur_monthly_wday](#)
- ◆ [mcaldel_event_set_recur_none](#)
- ◆ [mcaldel_event_set_recur_weekly](#)
- ◆ [mcaldel_event_set_recur_yearly](#)
- ◆ [mcaldel_event_set_start](#)
- ◆ [mcaldel_event_set_title](#)
- ◆ [mcaldel_expunge](#)
- ◆ [mcaldel_fetch_current_stream_event](#)
- ◆ [mcaldel_fetch_event](#)
- ◆ [mcaldel_is_leap_year](#)
- ◆ [mcaldel_list_alarms](#)
- ◆ [mcaldel_list_events](#)
- ◆ [mcaldel_next_recurrence](#)
- ◆ [mcaldel_open](#)
- ◆ [mcaldel_popen](#)
- ◆ [mcaldel_rename_calendar](#)
- ◆ [mcaldel_reopen](#)
- ◆ [mcaldel_snooze](#)
- ◆ [mcaldel_store_event](#)
- ◆ [mcaldel_time_valid](#)
- ◆ [mcaldel_week_of_year](#)
- ◆ [mcaldelcrypt_cbc](#)
- ◆ [mcaldelcrypt_cfb](#)
- ◆ [mcaldelcrypt_create_iv](#)
- ◆ [mcaldelcrypt_decrypt](#)
- ◆ [mcaldelcrypt_ecb](#)
- ◆ [mcaldelcrypt_enc_get_algorithms_name](#)
- ◆ [mcaldelcrypt_enc_get_block_size](#)
- ◆ [mcaldelcrypt_enc_get_iv_size](#)
- ◆ [mcaldelcrypt_enc_get_key_size](#)
- ◆ [mcaldelcrypt_enc_get_modes_name](#)
- ◆ [mcaldelcrypt_enc_get_supported_key_sizes](#)
- ◆ [mcaldelcrypt_enc_is_block_algorithm](#)
- ◆ [mcaldelcrypt_enc_is_block_algorithm_mode](#)
- ◆ [mcaldelcrypt_enc_is_block_mode](#)
- ◆ [mcaldelcrypt_enc_self_test](#)
- ◆ [mcaldelcrypt_encrypt](#)
- ◆ [mcaldelcrypt_generic](#)
- ◆ [mcaldelcrypt_generic_deinit](#)
- ◆ [mcaldelcrypt_generic_end](#)
- ◆ [mcaldelcrypt_generic_init](#)
- ◆ [mcaldelcrypt_get_block_size](#)
- ◆ [mcaldelcrypt_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](#)
- ◆ [mcrypt_module_is_block_algorithm_mode](#)
- ◆ [mcrypt_module_is_block_mode](#)
- ◆ [mcrypt_module_open](#)
- ◆ [mcrypt_module_self_test](#)
- ◆ [mcrypt_ofb](#)
- ◆ [mcve_adduser](#)
- ◆ [mcve_adduserarg](#)
- ◆ [mcve_bt](#)
- ◆ [mcve_checkstatus](#)
- ◆ [mcve_chkpwd](#)
- ◆ [mcve_chngpwd](#)
- ◆ [mcve_completeauthorizations](#)
- ◆ [mcve_connect](#)
- ◆ [mcve_connectionerror](#)
- ◆ [mcve_deleterespone](#)
- ◆ [mcve_deletetrans](#)
- ◆ [mcve_deleteusersetup](#)
- ◆ [mcve_deluser](#)
- ◆ [mcve_destroyconn](#)
- ◆ [mcve_destroyengine](#)
- ◆ [mcve_disableuser](#)
- ◆ [mcve_edituser](#)
- ◆ [mcve_enableuser](#)
- ◆ [mcve_force](#)
- ◆ [mcve_getcell](#)
- ◆ [mcve_getcellbynum](#)
- ◆ [mcve_getcommadelimited](#)
- ◆ [mcve_getheader](#)
- ◆ [mcve_getuserarg](#)
- ◆ [mcve_getuserparam](#)
- ◆ [mcve_gft](#)
- ◆ [mcve_gl](#)
- ◆ [mcve_gut](#)
- ◆ [mcve_initconn](#)
- ◆ [mcve_initengine](#)
- ◆ [mcve_initusersetup](#)
- ◆ [mcve_iscommadelimited](#)
- ◆ [mcve_liststats](#)
- ◆ [mcve_listusers](#)
- ◆ [mcve_maxconntimeout](#)
- ◆ [mcve_monitor](#)
- ◆ [mcve_numcolumns](#)
- ◆ [mcve_numrows](#)
- ◆ [mcve_override](#)
- ◆ [mcve_parsecommadelimited](#)

- ◆ [mcve_ping](#)
- ◆ [mcve_preauth](#)
- ◆ [mcve_preauthcompletion](#)
- ◆ [mcve_qc](#)
- ◆ [mcve_responseparam](#)
- ◆ [mcve_return](#)
- ◆ [mcve_returncode](#)
- ◆ [mcve_returnstatus](#)
- ◆ [mcve_sale](#)
- ◆ [mcve_setblocking](#)
- ◆ [mcve_setdropfile](#)
- ◆ [mcve_setip](#)
- ◆ [mcve_setssl](#)
- ◆ [mcve_setssl_files](#)
- ◆ [mcve_settimeout](#)
- ◆ [mcve_settle](#)
- ◆ [mcve_text_avs](#)
- ◆ [mcve_text_code](#)
- ◆ [mcve_text_cv](#)
- ◆ [mcve_transactionauth](#)
- ◆ [mcve_transactionavs](#)
- ◆ [mcve_transactionbatch](#)
- ◆ [mcve_transactioncv](#)
- ◆ [mcve_transactionid](#)
- ◆ [mcve_transactionitem](#)
- ◆ [mcve_transactionssent](#)
- ◆ [mcve_transactiontext](#)
- ◆ [mcve_transinqueue](#)
- ◆ [mcve_transnew](#)
- ◆ [mcve_transparam](#)
- ◆ [mcve_transsend](#)
- ◆ [mcve_ub](#)
- ◆ [mcve_uwait](#)
- ◆ [mcve_verifyconnection](#)
- ◆ [mcve_verifysslcert](#)
- ◆ [mcve_void](#)
- ◆ [md5](#)
- ◆ [md5_file](#)
- ◆ [mdecrypt_generic](#)
- ◆ [memory_get_usage](#)
- ◆ [metaphone](#)
- ◆ [method_exists](#)
- ◆ [mhash](#)
- ◆ [mhash_count](#)
- ◆ [mhash_get_block_size](#)
- ◆ [mhash_get_hash_name](#)
- ◆ [mhash_keygen_s2k](#)
- ◆ [microtime](#)
- ◆ [mime_content_type](#)
- ◆ [min](#)
- ◆ [ming_setcubicthreshold](#)
- ◆ [ming_setscale](#)
- ◆ [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_getdata](#)
- ◆ [msession_inc](#)
- ◆ [msession_list](#)
- ◆ [msession_listvar](#)
- ◆ [msession_lock](#)
- ◆ [msession_plugin](#)
- ◆ [msession_randstr](#)
- ◆ [msession_set](#)
- ◆ [msession_set_array](#)
- ◆ [msession_setdata](#)
- ◆ [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_dbname](#)
- ◆ [mysql_drop_db](#)
- ◆ [mysql_dropdb](#)
- ◆ [mysql_error](#)
- ◆ [mysql_fetch_array](#)
- ◆ [mysql_fetch_field](#)
- ◆ [mysql_fetch_object](#)
- ◆ [mysql_fetch_row](#)
- ◆ [mysql_field_seek](#)
- ◆ [mysql_fieldflags](#)
- ◆ [mysql_fieldlen](#)
- ◆ [mysql_fieldname](#)
- ◆ [mysql_fieldtable](#)
- ◆ [mysql_fieldtype](#)
- ◆ [mysql_free_result](#)
- ◆ [mysql_freeresult](#)
- ◆ [mysql_list_dbs](#)

- ♦ [mysql_list_fields](#)
- ♦ [mysql_list_tables](#)
- ♦ [mysql_listdbs](#)
- ♦ [mysql_listfields](#)
- ♦ [mysql_listtables](#)
- ♦ [mysql_num_fields](#)
- ♦ [mysql_num_rows](#)
- ♦ [mysql_numfields](#)
- ♦ [mysql_numrows](#)
- ♦ [mysql_pconnect](#)
- ♦ [mysql_query](#)
- ♦ [mysql_regcase](#)
- ♦ [mysql_result](#)
- ♦ [mysql_select_db](#)
- ♦ [mysql_selectdb](#)
- ♦ [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_affected_rows](#)
- ♦ [mysqli_autocommit](#)
- ♦ [mysqli_bind_param](#)
- ♦ [mysqli_bind_result](#)
- ♦ [mysqli_change_user](#)
- ♦ [mysqli_character_set_name](#)

- ♦ [mysqli_close](#)
- ♦ [mysqli_commit](#)
- ♦ [mysqli_connect](#)
- ♦ [mysqli_data_seek](#)
- ♦ [mysqli_debug](#)
- ♦ [mysqli_disable_reads_from_master](#)
- ♦ [mysqli_disable_rpl_parse](#)
- ♦ [mysqli_dump_debug_info](#)
- ♦ [mysqli_enable_reads_from_master](#)
- ♦ [mysqli_enable_rpl_parse](#)
- ♦ [mysqli_errno](#)
- ♦ [mysqli_error](#)
- ♦ [mysqli_execute](#)
- ♦ [mysqli_fetch](#)
- ♦ [mysqli_fetch_array](#)
- ♦ [mysqli_fetch_assoc](#)
- ♦ [mysqli_fetch_field](#)
- ♦ [mysqli_fetch_field_direct](#)
- ♦ [mysqli_fetch_fields](#)
- ♦ [mysqli_fetch_lengths](#)
- ♦ [mysqli_fetch_object](#)
- ♦ [mysqli_fetch_row](#)
- ♦ [mysqli_field_count](#)
- ♦ [mysqli_field_seek](#)
- ♦ [mysqli_field_tell](#)
- ♦ [mysqli_free_result](#)
- ♦ [mysqli_get_client_info](#)
- ♦ [mysqli_get_host_info](#)
- ♦ [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_num_fields](#)
- ♦ [mysqli_num_rows](#)
- ♦ [mysqli_options](#)
- ♦ [mysqli_param_count](#)
- ♦ [mysqli_ping](#)
- ♦ [mysqli_prepare](#)
- ♦ [mysqli_prepare_result](#)
- ♦ [mysqli_profiler](#)
- ♦ [mysqli_query](#)
- ♦ [mysqli_read_query_result](#)
- ♦ [mysqli_real_connect](#)
- ♦ [mysqli_real_escape_string](#)
- ♦ [mysqli_real_query](#)
- ♦ [mysqli_reload](#)
- ♦ [mysqli_rollback](#)
- ♦ [mysqli_rpl_parse_enabled](#)
- ♦ [mysqli_rpl_probe](#)
- ♦ [mysqli_rpl_query_type](#)

- ♦ [mysqli_select_db](#)
- ♦ [mysqli_send_long_data](#)
- ♦ [mysqli_send_query](#)
- ♦ [mysqli_slave_query](#)
- ♦ [mysqli_ssl_set](#)
- ♦ [mysqli_stat](#)
- ♦ [mysqli_stmt_affected_rows](#)
- ♦ [mysqli_stmt_close](#)
- ♦ [mysqli_stmt_errno](#)
- ♦ [mysqli_stmt_error](#)
- ♦ [mysqli_stmt_store_result](#)
- ♦ [mysqli_store_result](#)
- ♦ [mysqli_thread_id](#)
- ♦ [mysqli_thread_safe](#)
- ♦ [mysqli_use_result](#)
- ♦ [mysqli_warning_count](#)
- N
- - ♦ [natcasesort](#)
 - ♦ [natsort](#)
 - ♦ [ncurses_addch](#)
 - ♦ [ncurses_addchnstr](#)
 - ♦ [ncurses_addchstr](#)
 - ♦ [ncurses_addnstr](#)
 - ♦ [ncurses_addstr](#)
 - ♦ [ncurses_assume_default_colors](#)
 - ♦ [ncurses_attroff](#)
 - ♦ [ncurses_attron](#)
 - ♦ [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_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](#)
- ♦ [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](#)
- O
- - ♦ [ob_clean](#)
 - ♦ [ob_end_clean](#)
 - ♦ [ob_end_flush](#)
 - ♦ [ob_flush](#)
 - ♦ [ob_get_clean](#)
 - ♦ [ob_get_contents](#)
 - ♦ [ob_get_length](#)
 - ♦ [ob_get_level](#)
 - ♦ [ob_get_status](#)
 - ♦ [ob_gzhandler](#)
 - ♦ [ob_iconv_handler](#)
 - ♦ [ob_implicit_flush](#)
 - ♦ [ob_start](#)
 - ♦ [ocibindbyname](#)

- ◆ [ocicancel](#)
- ◆ [ocicloselob](#)
- ◆ [ocicollappend](#)
- ◆ [ocicollassign](#)
- ◆ [ocicollassignelem](#)
- ◆ [ocicollgetelem](#)
- ◆ [ocicollmax](#)
- ◆ [ocicollsize](#)
- ◆ [ocicolltrim](#)
- ◆ [ocicolumnname](#)
- ◆ [ocicolumnisnull](#)
- ◆ [ocicolumnprecision](#)
- ◆ [ocicolumnscale](#)
- ◆ [ocicolumnsize](#)
- ◆ [ocicolumntype](#)
- ◆ [ocicolumntyperaw](#)
- ◆ [ocicommit](#)
- ◆ [ocidefinebyname](#)
- ◆ [ocierror](#)
- ◆ [ociexecute](#)
- ◆ [ocifetch](#)
- ◆ [ocifetchinto](#)
- ◆ [ocifetchstatement](#)
- ◆ [ocifreecollection](#)
- ◆ [ocifreecursor](#)
- ◆ [ocifreedesc](#)
- ◆ [ocifreestatement](#)
- ◆ [ociinternaldebug](#)
- ◆ [ociloadlob](#)
- ◆ [ocilogoff](#)
- ◆ [ocilogon](#)
- ◆ [ocinewcollection](#)
- ◆ [ocinewcursor](#)
- ◆ [ocinewdescriptor](#)
- ◆ [ocinlogon](#)
- ◆ [ocinumcols](#)
- ◆ [ociparse](#)
- ◆ [ociplogon](#)
- ◆ [ocireresult](#)
- ◆ [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](#)
- ◆ [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_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](#)
- ◆ [overload](#)
- ◆ [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](#)
 - ♦ [parse_ini_file](#)
 - ♦ [parse_str](#)
 - ♦ [parse_url](#)
 - ♦ [passthru](#)
 - ♦ [pathinfo](#)
 - ♦ [pclose](#)
 - ♦ [pcntl_exec](#)
 - ♦ [pcntl_fork](#)
 - ♦ [pcntl_signal](#)
 - ♦ [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_locallink](#)
 - ♦ [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](#)
- ◆ [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_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](#)
- ◆ [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_pconnect](#)
- ◆ [pg_ping](#)
- ◆ [pg_port](#)
- ◆ [pg_put_line](#)
- ◆ [pg_query](#)
- ◆ [pg_result_error](#)
- ◆ [pg_result_seek](#)
- ◆ [pg_result_status](#)
- ◆ [pg_select](#)
- ◆ [pg_send_query](#)
- ◆ [pg_set_client_encoding](#)
- ◆ [pg_trace](#)
- ◆ [pg_tty](#)
- ◆ [pg_unescape_bytea](#)

- ◆ [pg_untrace](#)
- ◆ [pg_update](#)
- ◆ [php_ini_scanned_files](#)
- ◆ [php_logo_guid](#)
- ◆ [php_sapi_name](#)
- ◆ [php_uname](#)
- ◆ [phpcredits](#)
- ◆ [phpinfo](#)
- ◆ [phpversion](#)
- ◆ [pi](#)
- ◆ [png2wbmp](#)
- ◆ [popen](#)
- ◆ [pos](#)
- ◆ [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_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](#)
- ◆ [pspell_add_to_personal](#)
- ◆ [pspell_add_to_session](#)
- ◆ [pspell_check](#)
- ◆ [pspell_clear_session](#)
- ◆ [pspell_config_create](#)
- ◆ [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](#)
- **Q**
- - ♦ [qdom_error](#)
 - ♦ [qdom_tree](#)
 - ♦ [quoted_printable_decode](#)
 - ♦ [quotemeta](#)
- **R**
- - ♦ [rad2deg](#)
 - ♦ [rand](#)
 - ♦ [range](#)
 - ♦ [rawurldecode](#)
 - ♦ [rawurlencode](#)
 - ♦ [read_exif_data](#)
 - ♦ [readdir](#)
 - ♦ [readfile](#)
 - ♦ [readgzfile](#)
 - ♦ [readline](#)
 - ♦ [readline_add_history](#)
 - ♦ [readline_clear_history](#)
 - ♦ [readline_completion_function](#)
 - ♦ [readline_info](#)
 - ♦ [readline_list_history](#)
 - ♦ [readline_read_history](#)
 - ♦ [readline_write_history](#)
 - ♦ [readlink](#)
 - ♦ [realpath](#)
 - ♦ [recode](#)
 - ♦ [recode_file](#)
 - ♦ [recode_string](#)
 - ♦ [register_shutdown_function](#)
 - ♦ [register_tick_function](#)
 - ♦ [rename](#)
 - ♦ [reset](#)
 - ♦ [restore_error_handler](#)
 - ♦ [restore_include_path](#)
 - ♦ [rewind](#)
 - ♦ [rewinddir](#)
 - ♦ [rmdir](#)
 - ♦ [round](#)
 - ♦ [rsort](#)
 - ♦ [rtrim](#)
- **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_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_file_buffer](#)
- ◆ [set_include_path](#)
- ◆ [set_magic_quotes_runtime](#)
- ◆ [set_time_limit](#)
- ◆ [setcookie](#)
- ◆ [setlocale](#)
- ◆ [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](#)
- ◆ [sin](#)
- ◆ [sinh](#)
- ◆ [sizeof](#)
- ◆ [sleep](#)
- ◆ [snmp_get_quick_print](#)
- ◆ [snmp_set_quick_print](#)
- ◆ [snmpget](#)
- ◆ [snmprealwalk](#)
- ◆ [snmpset](#)
- ◆ [snmpwalk](#)
- ◆ [snmpwalkoid](#)
- ◆ [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_iovec_add](#)
- ◆ [socket_iovec_alloc](#)
- ◆ [socket_iovec_delete](#)
- ◆ [socket_iovec_fetch](#)
- ◆ [socket_iovec_free](#)
- ◆ [socket_iovec_set](#)
- ◆ [socket_last_error](#)
- ◆ [socket_listen](#)
- ◆ [socket_read](#)
- ◆ [socket_readv](#)
- ◆ [socket_recv](#)
- ◆ [socket_recvfrom](#)
- ◆ [socket_recvmsg](#)
- ◆ [socket_select](#)
- ◆ [socket_send](#)
- ◆ [socket_sendmsg](#)
- ◆ [socket_sendto](#)
- ◆ [socket_set_block](#)
- ◆ [socket_set_blocking](#)
- ◆ [socket_set_nonblock](#)
- ◆ [socket_set_option](#)
- ◆ [socket_set_timeout](#)
- ◆ [socket_shutdown](#)
- ◆ [socket_strerror](#)
- ◆ [socket_write](#)
- ◆ [socket_writev](#)

- ♦ [sort](#)
- ♦ [soundex](#)
- ♦ [split](#)
- ♦ [spliti](#)
- ♦ [sprintf](#)
- ♦ [sql_regcase](#)
- ♦ [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_fetch_array](#)
- ♦ [sqlite_fetch_single](#)
- ♦ [sqlite_fetch_string](#)
- ♦ [sqlite_field_name](#)
- ♦ [sqlite_has_more](#)
- ♦ [sqlite_last_error](#)
- ♦ [sqlite_last_insert_rowid](#)
- ♦ [sqlite_libencoding](#)
- ♦ [sqlite_libversion](#)
- ♦ [sqlite_next](#)
- ♦ [sqlite_num_fields](#)
- ♦ [sqlite_num_rows](#)
- ♦ [sqlite_open](#)
- ♦ [sqlite_popen](#)
- ♦ [sqlite_query](#)
- ♦ [sqlite_rewind](#)
- ♦ [sqlite_seek](#)
- ♦ [sqlite_udf_decode_binary](#)
- ♦ [sqlite_udf_encode_binary](#)
- ♦ [sqlite_unbuffered_query](#)
- ♦ [sqrt](#)
- ♦ [srand](#)
- ♦ [sscanf](#)
- ♦ [stat](#)
- ♦ [str_ireplace](#)
- ♦ [str_pad](#)
- ♦ [str_repeat](#)
- ♦ [str_replace](#)
- ♦ [str_rot13](#)
- ♦ [str_shuffle](#)
- ♦ [str_split](#)
- ♦ [str_word_count](#)
- ♦ [strcasecmp](#)
- ♦ [strchr](#)
- ♦ [strcmp](#)
- ♦ [strcoll](#)
- ♦ [strcspn](#)
- ♦ [stream_context_create](#)

- ◆ [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_get_filters](#)
- ◆ [stream_get_line](#)
- ◆ [stream_get_meta_data](#)
- ◆ [stream_get_transports](#)
- ◆ [stream_get_wrappers](#)
- ◆ [stream_select](#)
- ◆ [stream_set_blocking](#)
- ◆ [stream_set_timeout](#)
- ◆ [stream_set_write_buffer](#)
- ◆ [stream_socket_accept](#)
- ◆ [stream_socket_client](#)
- ◆ [stream_socket_get_name](#)
- ◆ [stream_socket_server](#)
- ◆ [stream_wrapper_register](#)
- ◆ [strftime](#)
- ◆ [strip_tags](#)
- ◆ [stripcslashes](#)
- ◆ [stripos](#)
- ◆ [stripslashes](#)
- ◆ [stristr](#)
- ◆ [strlen](#)
- ◆ [strnatcasecmp](#)
- ◆ [strnatcmp](#)
- ◆ [strncasecmp](#)
- ◆ [strncmp](#)
- ◆ [strpos](#)
- ◆ [strrchr](#)
- ◆ [strev](#)
- ◆ [stripos](#)
- ◆ [strrpos](#)
- ◆ [strspn](#)
- ◆ [strstr](#)
- ◆ [strtok](#)
- ◆ [strtolower](#)
- ◆ [strtotime](#)
- ◆ [strtoupper](#)
- ◆ [strtr](#)
- ◆ [strval](#)
- ◆ [substr](#)
- ◆ [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_fonttracking](#)
- ◆ [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_keypress](#)
- ◆ [swfdisplayitem](#)
- ◆ [swfdisplayitem->addcolor](#)
- ◆ [swfdisplayitem->move](#)
- ◆ [swfdisplayitem->moveto](#)
- ◆ [swfdisplayitem->multicolor](#)
- ◆ [swfdisplayitem->remove](#)
- ◆ [swfdisplayitem->rotate](#)
- ◆ [swfdisplayitem->rotateto](#)
- ◆ [swfdisplayitem->scale](#)
- ◆ [swfdisplayitem->scaletto](#)
- ◆ [swfdisplayitem->setdepth](#)
- ◆ [swfdisplayitem->setname](#)
- ◆ [swfdisplayitem->setratio](#)
- ◆ [swfdisplayitem->skewx](#)
- ◆ [swfdisplayitem->skewxto](#)
- ◆ [swfdisplayitem->skewy](#)
- ◆ [swfdisplayitem->skewyto](#)
- ◆ [swffill](#)
- ◆ [swffill->moveto](#)
- ◆ [swffill->rotateto](#)
- ◆ [swffill->scaletto](#)
- ◆ [swffill->skewxto](#)
- ◆ [swffill->skewyto](#)
- ◆ [swffont](#)
- ◆ [swffont->getwidth](#)
- ◆ [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](#)
- ◆ [swfshape](#)
- ◆ [swfshape->addfill](#)
- ◆ [swfshape->drawcurve](#)
- ◆ [swfshape->drawcurveto](#)
- ◆ [swfshape->drawline](#)
- ◆ [swfshape->drawlineto](#)
- ◆ [swfshape->movepen](#)
- ◆ [swfshape->movepento](#)
- ◆ [swfshape->setleftfill](#)
- ◆ [swfshape->setline](#)
- ◆ [swfshape->setrightfill](#)
- ◆ [swfsprite](#)
- ◆ [swfsprite->add](#)
- ◆ [swfsprite->nextframe](#)
- ◆ [swfsprite->remove](#)
- ◆ [swfsprite->setframes](#)
- ◆ [swftext](#)
- ◆ [swftext->addstring](#)
- ◆ [swftext->getwidth](#)
- ◆ [swftext->moveto](#)
- ◆ [swftext->setcolor](#)
- ◆ [swftext->setfont](#)
- ◆ [swftext->setheight](#)
- ◆ [swftext->setspacing](#)
- ◆ [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](#)
- ◆ [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](#)
 - ♦ [tempnam](#)
 - ♦ [textdomain](#)
 - ♦ [time](#)
 - ♦ [tmpfile](#)
 - ♦ [token_get_all](#)
 - ♦ [token_name](#)
 - ♦ [touch](#)
 - ♦ [trigger_error](#)
 - ♦ [trim](#)
- U
- - ♦ [uasort](#)
 - ♦ [ucfirst](#)
 - ♦ [ucwords](#)
 - ♦ [udm_add_search_limit](#)
 - ♦ [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_load_ispell_data](#)

- ♦ [udm_open_stored](#)
- ♦ [udm_set_agent_param](#)
- ♦ [uksort](#)
- ♦ [umask](#)
- ♦ [uniqid](#)
- ♦ [unixtojd](#)
- ♦ [unlink](#)
- ♦ [unpack](#)
- ♦ [unregister_tick_function](#)
- ♦ [unserialize](#)
- ♦ [unset](#)
- ♦ [urldecode](#)
- ♦ [urlencode](#)
- ♦ [user_error](#)
- ♦ [usleep](#)
- ♦ [usort](#)
- ♦ [utf8_decode](#)
- ♦ [utf8_encode](#)
- V
- - ♦ [var_dump](#)
 - ♦ [var_export](#)
 - ♦ [version_compare](#)
 - ♦ [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](#)

- ♦ [wordwrap](#)
- X
- - ♦ [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_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](#)
 - ♦ [xptr_eval](#)
 - ♦ [xptr_new_context](#)
 - ♦ [xslt_create](#)
 - ♦ [xslt_errno](#)
 - ♦ [xslt_error](#)
 - ♦ [xslt_free](#)
 - ♦ [xslt_process](#)
 - ♦ [xslt_set_base](#)
 - ♦ [xslt_set_encoding](#)
 - ♦ [xslt_set_error_handler](#)
 - ♦ [xslt_set_log](#)
 - ♦ [xslt_set_sax_handler](#)
 - ♦ [xslt_set_sax_handlers](#)
 - ♦ [xslt_set_scheme_handler](#)

- ♦ xslt_set_scheme_handlers
- Y
 - ♦ yaz_addinfo
 - ♦ yaz_ccl_conf
 - ♦ yaz_ccl_parse
 - ♦ yaz_close
 - ♦ yaz_connect
 - ♦ yaz_database
 - ♦ yaz_element
 - ♦ yaz_errno
 - ♦ yaz_error
 - ♦ 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

12.2 Index des concepts

- A
- >
- écrit des octets depuis un buffer vers un flôt
- .
-
- ...
- A
-
- Accepte une connexion sur une socket
- Accepte une connexion sur une socket créée par `stream_socket_server`
- Active l'écho d'entrée clavier
- Active l'approximation des translation d'objets.
- Active l'interruption de script sur déconnexion du visiteur
- Active la couche de contrôle des membres et méthodes
- Active la sauvegarde des paires de remplacement
- Active la validation automatique oracle
- Active le debugger interne de php
- Active le mode auto-validation
- Active le suivi d'une connexion postgresql
- Active les attributs suivants
- Active les couleurs
- Active ou désactive l'affichage des données de débogage oracle
- 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 les alertes frontbase
- Active ou désactive un code de clé (keycode)
- Active un utilisateur mcve inactif
- Active/désactive l'envoi implicite
- Active/désactive l'option magic_quotes_runtime.
- Active/désactive le mode bloquant d'une socket
- Active/désactive les informations de méta clé 8-bits
- Add data to a transaction
- Addition de 2 nombres gmp
- Additionne deux nombres de grande taille
- Affecte le gestionnaire xml par défaut
- Affecte les gestionnaires d'entités non déclarés
- 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
- Affectent les fonctions de flôt
- Affiche de nombreuses informations sur le 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 le contenu d'un blob ibase au navigateur
- Affiche le contenu d'un fichier en une seule fois
- Affiche le document pdf qui est en mémoire
- Affiche le frame courant
- Affiche le frame nommé.
- Affiche le résultat sous la forme d'une table html.
- Affiche le reste du fichier
- Affiche le texte à la position courante
- Affiche le texte à la position courante dans la fenêtre
- Affiche le texte dans un cadre
- Affiche les informations d'une variable.
- Affiche ou affecte le paramètre "apache request notes"
- Affiche tout ce qui est prêt, et rafraichit l'écran
- Affiche toutes les données restantes d'un flôt
- Affiche un caractère et rafraichit l'écran
- Affiche un fichier
- Affiche un résultat ovrimos sous forme de table html
- Affiche un text à une position donnée
- Affiche un texte à une position
- Affiche une backtrace
- Affiche une chaîne de caractères
- Affiche une chaîne de caractères
- Affiche une chaîne de caractères formattée
- Affiche une chaîne formatée
- Affiche une image wbmp
- Aggrège dynamiquement les méthodes d'une classe à un objet
- Aggrège dynamiquement les propriétés d'une classe à un objet
- Aggrège dynamiquement des classes et objets
- Aggrège sélectivement les méthodes d'une classe grâce à une expression régulière
- Aggrège sélectivement les méthodes d'une classe grâce à une liste
- Aggrège sélectivement les propriétés d'une classe grâce à une liste
- Aggrège sélectivement les propriétés d'une classe grâce à une expression régulière
- Ajout un alias pour un domaine virtuel
- Ajoute au texte
- Ajoute des anti-slashes dans une chaîne
- Ajoute des données d'introspection
- Ajoute des données dans un blob ibase fraîchement crée
- Ajoute des slashes dans une chaîne, à la mode du langage c
- Ajoute différentes limitations de recherche
- Ajoute du code javascript dans un document fdf
- Ajoute du texte
- Ajoute le mot au dictionnaire personnel
- Ajoute le mot au dictionnaire personnel de la session courante
- Ajoute un événement souris dans la queue
- Ajoute un alias à un domaine virtuel
- Ajoute un alias virtuel
- Ajoute un attribut à l'entrée courante
- Ajoute un attribut et une valeur à la structure globale
- 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 enregistrement dans une base dbase
- Ajoute un fichier attaché à la page pdf
- Ajoute un frère à un noeud

- [Ajoute un lien web sur la page pdf courante](#)
- [Ajoute un noeud](#)
- [Ajoute un nouveau domaine virtuel](#)
- [Ajoute un nouveau domaine virtuel](#)
- [Ajoute un nouveau fils new child at the end of the children](#)
- [Ajoute un nouveau vecteur dans le tableau](#)
- [Ajoute un nouvel attribut](#)
- [Ajoute un nouvel utilisateur à un domaine virtuel](#)
- [Ajoute un objet à un sprite](#)
- [Ajoute un objet à une collection oracle](#)
- [Ajoute un objet dans une animation](#)
- [Ajoute un paramètre à une procédure stockée mssql \(locale ou distante\)](#)
- [Ajoute un paramètre à une transaction](#)
- [Ajoute un remplissage plein à la forme flash](#)
- [Ajoute un signet à la page courante](#)
- [Ajoute un signet dans la page courante](#)
- [Ajoute un template dans le document fdf](#)
- [Ajoute un utilisateur à une base de données de sécurité \(uniquement pour ib6 ou plus récent\)](#)
- [Ajoute un utilisateur mcve avec une structure usersetup](#)
- [Ajoute un utilisateur sur la liste des utilisateurs indésirables](#)
- [Ajoute une action au bouton](#)
- [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 à la position courante](#)
- [Ajoute une chaîne à la position courante](#)
- [Ajoute une chaîne à une nouvelle position dans la fenêtre](#)
- [Ajoute une chaîne dans une boîte aux lettres.](#)
- [Ajoute une chaîne de taille donnée à la position courante](#)
- [Ajoute une contrainte sur une relation](#)
- [Ajoute une couleur à la liste du gradient](#)
- [Ajoute une couleur à une transoformation](#)
- [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 ressource curl à un curl multiple](#)
- [Ajoute une valeur dans une structure de configuration d'utilisateur](#)
- [Ajouter des variables à un paquet wddx](#)
- [Alias de `count`](#)
- [Alias de `disk\_free\_space`](#)
- [Alias de `exif\_read\_data`](#)
- [Alias de `floatval`](#)
- [Alias de `get\_included\_files`](#)
- [Alias de `ini\_set`](#)
- [Alias de `is\_float`](#)
- [Alias de `is\_int`](#)
- [Alias de `odbc\_field\_len`](#)

- Alias de [rtrim](#)
- Alias de [sqlite fetch single](#)
- Alias de [stream set timeout](#)
- Alias de [stream set write buffer](#)
- Alias de [stream wrapper register](#)
- Alias de [trigger error](#)
- 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 une couleur à une image
- Alloue une couleur pour une image
- Alloue une paire de couleur
- Alloue une session mnogosearch
- Analyse incrémentalement des données dans un buffer
- Analyse les adresses et retourne un tableau contenant ces données
- 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 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 au format csv pour que [mcve_getcell](#) et consorts fonctionnent
- Analyse une chaîne à l'aide d'un format
- Analyse une requête http
- Analyse une requête sql avec oracle
- 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 (rend void) une transaction dans la queue d'attente
- Annule la création d'un blob ibase
- Annule la fonction exécutée à chaque tick
- Annule les transactions oracle en cours
- Annule toutes les limites de recherche mnogosearch
- Annule un bit
- Annule une requête asynchrone
- Annule une transaction
- Annule une transaction
- 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
- Appelle une fonction d'échappement dans les plugins de personnalité msession
- Appelle l'analyseur ccl
- Appelle une fonction utilisateur
- Appelle une fonction utilisateur à chaque valeur fdf

- Appelle une fonction utilisateur avec les paramètres rassemblés en tableau
- Appelle une fonction windows 32
- Appelle une méthode d'un composant com
- Appelle une méthode utilisateur avec un tableau de paramètres
- Appelle une méthode utilisateur d'un objet
- Applique la syntaxe colorisée à du code php
- Applique sur fonction sur des tableaux
- Applique une correction gamma à l'image
- Applique une transformation xslt à un objet domdocument
- Arc cosinus
- Arc cosinus hyperbolique
- Arc sinus
- Arc sinus hyperbolique
- Arc tangeant hyperbolique
- Arc tangent
- Arc tangent de deux variables
- Archive un objet
- Arrête l'animation flash.
- Arrondi
- Arrondit à l'entier inférieur
- Arrondit au nombre supérieur
- Assigne l'élément val à une collection oracle à l'index ndx
- 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 collection depuis une autre collection oracle
- Assigne une valeur à un champ fdf
- Attache un filtre à un flôt en début de liste
- Attache un filtre à un flôt en fin de liste
- Attend l'exécution d'une requête
- Attend la fin de l'exécution d'un processus fils
- Attend un événement ibase
- Attend une connexion (listen) sur une socket
- Attend x microsecondes
- Attributs de l'objet dans l'ancrage
- Attributs des ancrages d'un document
- Attributs des documents fils d'un groupe
- Attributs des parents
- Audite mcve pour les transactions échouées
- Audite mcve pour les transactions finalisées
- Audite mcve pour les transactions non conclues
- Audite mcve pour une liste de transactions dans la queue de sortie
- Autentification au serveur ldap
- Avance d'un frame
- Avance le pointeur interne d'un tableau
- **B**
-
- Balise de corps d'un document
- **C**
-
- Calcul un chemin xptr à partir d'une chaîne
- Calcule des statistiques sur une table
- Calcule l'intersection de deux tableaux avec des tests sur les index
- Calcule l'intersection de tableaux

- Calcule la clé métaphone
- Calcule la clé soundex
- Calcule la différence de deux tableaux, en prenant en compte les clés
- Calcule la différence entre deux tableaux
- Calcule la distance levenshtein entre deux chaînes
- 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 la valeur de hash demandée par ezmlm
- Calcule le complémentaire d'un nombre
- Calcule le md5 d'un fichier
- Calcule le md5 d'une chaîne
- Calcule le nombre de couleurs d'une palette
- 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 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 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 nom d'utilisateur de la session active
- Change le nom de session de l'utilisateur actif
- Change les attributs d'un objet (obsolète)
- Charge les données ispell
- Charge un fichier sur un serveur ftp
- Charge un fichier sur un serveur ftp
- Charge un lob oracle

- 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 une extension php à la volée
- Charge une librairie de type (typelib)
- Charge une nouvelle police
- Charge une police
- Charge une police postscript type 1 depuis un fichier
- Check for pending fam events
- Check the status of an invoice
- Chemin du dossier de travail courant
- Cherche les notes par mots clés dans database_name
- Chiffre indéchiffrable (hashing)
- Chiffre
- Chiffre des données avec une clé privée
- Chiffre des données avec une clé publique
- Chiffre un message s/mime
- Chiffre un texte
- Chiffre/déchiffre des données en mode cfb
- Chiffre/déchiffre des données en mode ecb
- Chiffre/déchiffre des données en mode ofb
- Chiffre/déchiffre des données en mode cbc
- Choisi une fonction utilisateur comme gestionnaire d'erreurs
- Choisi une nouvelle position et dessine une ligne horizontale avec le caractère donné, et de taille maximale donnée
- Choisit l'encodage du client
- 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 par texture répétée.
- Choisit le mode de remplissage par texture.
- 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 haut ou bas.
- Choisit le point de départ de la recherche par expression régulière
- 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 dossier
- Clone un noeud
- Clone un objet
- Clos un paquet wddx
- Close fam connection
- Colorisation de la syntaxe d'un fichier
- Combine plusieurs tableaux ensembles, récursivement
- Commence l'analyse d'un fichier xml
- Commence l'utilisation de l'attribut 'standout'

- Commence la définition d'un bouton
- Commence la description d'une liste d'action pour la frame courante.
- Commence un nouveau chemin
- Commence un nouveau document
- Commence une forme complexe
- Commence une nouvelle page
- Commence une nouvelle page
- Commencer un nouveau paquet wddx avec une structure
- Communique avec mcve (envoie et reçoit des données) sans blocage
- 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 binaires
- Comparaison de chaînes localisée
- Compare des nombres gmp
- Compare deux dates
- Compare deux lignes afin de les trier
- Compare deux nombres de grande taille
- Compare deux versions de php
- Compare en binaire des chaînes de caractères
- Compare une entrée avec des valeurs d'attributs
- Complète la définition de la forme courante.
- Complète un tableau jusqu'à la longueur spécifiée, avec une valeur.
- Complète une chaîne jusqu'à une taille donnée
- Complète une preauthorization et la prépare pour la finalisation
- Compresse une chaîne
- Compresse une chaîne avec bzip2
- Compresse une chaîne avec la méthode deflate
- Compte de le nombre d'occurrences de segments dans une chaîne
- Compte de population
- Compte le nombre d'éléments d'une variable
- 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 de champs d'une base dbase
- Compte le nombre de ligne 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 une matrice avec le ctm
- Configure des pointillés complexes
- Configure l'échelle (??)
- Configure l'échelle du document
- Configure l'analyseur ccl
- Configure l'identifiant vers lequel les liens sont calculés
- Configure l'inclinaison d'un texte
- Configure l'uri de base de toutes les transformations ^{xsit}
- Configure la "miter limit"
- Configure la bufferisation de fichier pour un flôt
- Configure la connexion à l'imprimante
- Configure la connexion en mode bloquant/non bloquant

- Configure la connexion en mode drop-file
- Configure la connexion en mode ip
- Configure la connexion en mode ssl
- Configure la couleur de dessin à un niveau de gris
- 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 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 flôt
- 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 délai d'attente d'une base sqlite occupée
- Configure le fichier d'historique de la connexion
- 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 message 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 mode bloquant d'un flôt
- 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 des séquences clavier spéciales
- Configure le temps maximum d'une transaction (par transaction)
- Configure le verrouillage et l'isolation de transactions
- Configure les événements de souris à surveiller
- Configure les chaînes et les expressions régulières 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 opérations à exécuter lorsque la connexion s'interrompt
- Configure les paramètres pour un flôt/wrapper/contexte
- Configure mcve pour qu'il pingue ou pas la connexion, afin d'en vérifier l'existence
- Configure mcve pour qu'il vérifie ou pas le certificat ssl
- Configure un écran depuis un fichier de sauvegarde
- Configure un gestionnaire de schéma pour un processeur ^{xslt}
- 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 flôt/wrapper/contexte
- Connecte à un serveur irc
- Connexion à un serveur
- Connexion à une source
- Conserve des information sur le fichier ou l'url
- Conserve des informations sur un fichier durant une analyse de dossier
- Conserve les fonctions d'un gestionnaire de flôt
- Conserve les fonctions membres d'une implémentation de flôt
- Conserve les propriétés et le pointeur de filtre
- Conserve les propriétés et pointeur d'un gestionnaire de flôt
- Conserver les fonctions membres pour un filtre de flôt
- Considère deux mots accolés comme un composé.
- Constantes définies dans le coeur de php, zend et les modules sapi
- Constantes standards dans php
- Construit un 'struct iovec'
- Contenu d'un document
- Contrôle l'utilisation des informations d'environnement concernant la taille du terminal
- Contrôle la situation, l'apparence et la zone active du bouton courant
- Controle l'utilisation des noms étendus dans les descriptions terminfo
- Conversion d'encodage
- Convert it un dn en format ufn (user friendly naming)
- Convertit d'octal en décimal
- 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 de radians en degrés
- Convertit des données 8bit en texte utf-7.
- Convertit des données binaires en représentation hexadécimale
- 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 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 nouvelles lignes et retour chariot en saut 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 identifiatn de projet en une clé system v ipc
- Convertit un document dom xml en fichier
- Convertit un flôt en flôt avec un pointeur de position

- Convertit un flôt en une autre forme, comme un file* ou une socket
- Convertit un noeud en chaîne
- Convertit un nombre de degrés en radians
- Convertit un nombre de jours julien en timestamp unix
- 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 pointeur ansi file* en un flôt
- Convertit un pointeur de socket en flôt
- Convertit un tableau associatif en une commande postgresql
- 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 unix en nombre de jours julien
- Convertit une adresse ip (ipv4) en adresse ip numérique
- Convertit une chaîne à 8 bits en une chaîne à base64.
- Convertit une chaîne à 8 bits en une chaîne à guillemets.
- Convertit une chaîne à guillemets en une chaîne à 8 bits.
- 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 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
- 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
- Copie des données d'un flôt à l'autre
- Copie des données depuis un flôt dans un buffer
- Copie des données depuis un flôt 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 toutes les clés soft label à l'écran virtuel
- Copie un fichier
- Copie une partie d'une image
- 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
- Copies physically

- Cosinus
- Cosinus hyperbolic
- Coupe une chaîne en segments
- Coupe une chaîne en segments
- Coupe une partie de chaîne
- Crée un blob ibase, y copie un fichier et le referme
- Crée un nouveau conteneur et l'associe à une fenêtre
- 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 message
- 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 clob
- Crée un contexte d'imprimante
- Crée un contexte de flôt
- Crée un document xml vide
- Crée un dossier
- Crée un dossier sur un serveur ftp
- Crée un fichier avec un nom unique
- Crée un fichier fifo (first in, first out) (un pipe nommé)
- Crée un fichier temporaire
- Crée un lien
- Crée un lien avec un objet
- Crée un lien symbolique
- Crée un message mime
- Crée un morphing
- Crée un nom de navigateur dans database_name
- Crée un nombre gmp
- Crée un nouveau bouton
- Crée un nouveau calendrier
- Crée un nouveau champs texte
- Crée un nouveau contexte xpath
- Crée un nouveau contexte xpath
- Crée un nouveau document
- Crée un nouveau document fdf
- Crée un nouveau noeud élément
- Crée un nouveau noeud cdata
- Crée un nouveau noeud de commentaire
- Crée un nouveau noeud de texte
- Crée un nouveau noeud de traitement d'insctructions (pi)
- Crée un nouveau stylo
- Crée un nouvel élément avec un espace de noms associé
- Crée un nouvel analyseur xslt
- Crée un nouvel index secondaire pour une relation
- Crée un nouvel objet d'affichage displayitem
- Crée un nouvel objet pdf
- Crée un nouvel objet texte
- 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 à partir d'un fichier xml
- Crée un objet dom pour un document xml
- Crée un objet dumentityreference
- Crée un objet domstylesheet à partir d'un document xml
- Crée un objet domstylesheet à partir d'un document xsl dans un fichier
- Crée un objet domstylesheet à partir d'un objet domdocument
- Crée un objet gradient
- Crée un objet slob et l'ouvre
- Crée un processus de pointeur de fichier
- Crée un serveur xmlrpc
- Crée un sprite
- Crée un tableau
- Crée un tableau à partir de deux autres tableaux
- Crée un tableau contenant les variables et leur valeur
- Crée un tableau contenant un intervalle d'éléments
- Crée un vecteur d'initialisation à partir d'une source aléatoire.
- Crée une arbre à partir d'un code xml
- Crée une base de données
- Crée une base de données dbase
- Crée une base de données msq1
- Crée une base de données msq1
- Crée une base de données mysql
- Crée une base lotus notes
- Crée une chaîne compressée
- Crée une configuration utilisée pour ouvrir un dictionnaire
- Crée une connexion persistante
- Crée une connexion sur une socket
- Crée une copie exacte d'une relation, sans les indexes
- Crée une copie exacte mais vide d'une relation
- Crée une fonction anonyme (style lambda)
- Crée une image à partir d'un fichier xbm
- Crée une image à partir d'un fichier xpm
- Crée une image à partir d'une chaîne
- Crée une image depuis un fichier wbmp
- Crée une image wbmp
- Crée une instance de type de données et la remplit
- Crée une note à partir du formulaire form_name
- Crée une note à partir du formulaire form_name
- Crée une nouveau pad (fenêtre)
- Crée une nouvelle action
- Crée une nouvelle boîte aux lettres
- Crée une nouvelle brosse
- Crée une nouvelle fenêtre
- Crée une nouvelle forme
- Crée une nouvelle image ^{ieeg} 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 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 image en vraies couleurs
- 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 paire de socket identiques et les stocke dans fds.
- Crée une référence sur un composant com
- Crée une session
- Crée une socket (point de communication)
- Crée une socket serveur unix ou internet
- Crée une variable php à partir d'une valeur linéarisée
- Create a new, blank transaction
- Create new attribute
- **D**
- - 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
 - Déclenche une erreur utilisateur
 - Décode des données binaires, passées à une udf sqlite
 - 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 éléments mime d'une en-tête
 - Décode les données de session
 - Décode les entités html en caractères
 - Décode un texte encodé en base64
 - Décode une chaîne en mime base64
 - Décode une chaîne encodée url
 - Décode une chaîne modifiée utf-7.
 - Décode une chaîne url
 - Décode une en-tête mime
 - Décompresse une chaîne
 - Décompresse une chaîne bzip2
 - Décompresse une chaîne compressée
 - Déconditionne des données depuis une chaîne binaire.
 - Déconfigure le mode brut du terminal
 - Déconnecte d'un serveur ldap
 - Déconnecte d'un serveur oracle
 - Déconnexion d'une base sesam
 - Décrémente le compteur de références
 - Décrit la librairie dbm utilisée
 - Décrit une transition utilisée pour déclencher une liste d'actions.
 - Dédoublonne un tableau
 - Défini le point de vue de l'utilisateur en coordonnées polaire.
 - Défini un code de clé (keycode)
 - Définit l'option pour enregistrer les informations complètes du canal alpha lors de la

sauvegarde d'images png.

- Définit la couleur 0
- Définit la couleur transparente
- Définit le fond d'écran
- Définit les propriétés de fond d'écran pour le terminal
- Définit un polygone.
- 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 image bitmap
- Définit une ligne
- Définit une police.
- 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 projection orthogonale entre les coordonnées utilisateur et le port courant.
- Définit une transformation de vue
- Démarre tls
- Démarre une base de données fbsql
- Démarre une nouvelle transaction
- Démarre une section de texte
- 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'origine de l'objet swffill
- Déplace la position courante et ajoute un caractère
- Déplace la position d'affichage
- 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 la nouvelle position
- Déplace le pointeur de lecture
- Déplace le pointeur de résultat
- Déplace le pointeur de résultat
- Déplace le pointeur de résultat sqlite vers une ligne
- Déplace le pointeur de résultat sur le champ spécifié
- 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
- Déplace le pointeur interne de résultat fbsql
- Déplace le pointeur interne ms sql server au le 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 position d'affichage de la fenêtre
- 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
- Désactive l'écho clavier
- Désactive la validation automatique oracle
- Désactive le debugger interne de php.
- Désactive le remplissage.
- Désactive les attributs d'une fenêtre
- Désactive les attributs donnés
- Désactive un utilisateur mcve actif
- Désenregistre un gestionnaire pour l'api de flôts
- Détecte l'encodage mime
- Détecte le type d'encodage d'un caractère http
- Détecte le type mime d'un fichier
- Détecte un encodage
- Détermine la position du pointeur dans un flôt
- Détermine le meilleur encodage pour un fichier
- 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 un flôt est d'un type particulier
- Détermine si un flôt est un flôt persistant
- Détermine si un flôt peut être converti en une autre forme, comme file* ou socket
- 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 variable contient une valeur non nulle
- Détermine si une variable est affectée
- Détermine si une variable est de type chaîne
- Détermine si une variable est de type float
- Détermine si une variable est de type int
- Détermine si une variable est de type integer
- Détermine si une variable est de type object
- Détermine si une variable est de type real
- 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 session
- Détruit le vecteur iovec spécifié par iovec_id
- 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 document
- Détruit un pointeur de lob oracle
- Détruit un sémaphore
- Détruit un serveur xmlrpc
- Détruit une file de messages
- Détruit une image
- Détruit une relation
- Détruit une session
- Détruit une session mnogosearch
- Détruit une variable
- Déverrouille un objet
- Déverrouille un objet hyperwave verrouillé
- Déverrouille une session
- Decodes encoded nickname
- Delete a transaction
- Demande au serveur de terminer un thread mysql
- Demande d'informations d'arbre sur une entité du réseau.
- Desaggrège un objet
- Dessine la ligne le long du chemin
- Dessine le bord d'une fenêtre avec un caractère qualifié
- 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 anti-horaire
- 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 dans un document pdf
- Dessine un fichier bmp
- Dessine un pixel
- Dessine un polygone
- Dessine un polygone rempli
- Dessine un rectangle
- Dessine un rectangle
- Dessine un rectangle
- 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 arc de cercle
- Dessine une chaîne horizontale
- Dessine une chaîne verticale
- Dessine une courbe
- Dessine une courbe
- Dessine une courbe bézier cubique
- Dessine une courbe de bézier quadratique entre deux points.
- Dessine une courbe de bezier

- 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 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 à la position courante en utilisant un caractère donnée, pour une taille donnée
- Dessine une ligne verticale dans une fenêtre, à la position courante, en utilisant un caractère qualifié, avec une taille maximale
- Dessine une ligne, relativement
- Distance de hamming
- Divise deux nombres de grande taille
- Divise deux nombres gmp
- Divise deux nombres gmp
- Division exacte de nombres gmp
- Divisions de 2 nombres gmp
- Duplique un objet blob
- E
- - Echappe les méta-caractères
 - Echappement des caractères spéciaux des expressions régulières.
 - Eclate une chaîne par expression régulière
 - Ecrire dans un bloc de mémoire partagée
 - Ecrit dans l'historique
 - Ecrit dans un fichier
 - Ecrit dans un fichier compressé, en gzip
 - Ecrit dans une socket
 - Ecrit des données dans le fichier
 - Ecrit des données dans un descripteur de fichier, fd, en utilisant le tableau scatter-gather définie par iovec_id
 - Ecrit des données sur l'imprimante
 - Ecrit du texte dans une image avec la police courante et freetype 2
 - Ecrit la valeur courante de l'option quick_print de la librairie ucd
 - Ecrit le résultat d'un transformation xslt dans un fichier
 - Ecrit le résultat d'un transformation xslt dans une chaîne
 - Ecrit les données de session et ferme la session
 - Ecrit les informations de débogage dans le log
 - Ecrit toutes les modifications apportées à une relation
 - Ecrit un blob oracle temporaire
 - Ecrit un document pdf dans un fichier

- Ecrit un fichier sur un serveur ftp, et le lit depuis un fichier (non bloquant)
- Ecrit un lob oracle dans un fichier
- Ecrit un objet de grande taille pgsql
- Ecrit une chaîne dans un fichier
- Ecrit une chaîne dans un fichier compressé
- Ecrit une chaîne dans un objet slob
- Ecrit une chaîne formatée dans un flôt
- Ecriture binaire dans un fichier bzip2
- Ecriture du fichier en mode binaire
- Efface des lignes postgresql
- Efface des objets
- Efface et remplace une portion de tableau
- Efface l'écran
- Efface l'écran depuis la position courante jusqu'à la fin de la ligne
- Efface l'écran depuis la position courante jusqu'au bas de l'écran
- Efface l'écran du terminal
- Efface l'historique
- 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 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 le résultat de la mémoire
- 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 toutes les erreurs précédemment générées par une socket
- Efface un événement dans un calendrier mcal.
- Efface un attribut à l'entrée courante
- Efface un calendrier
- Efface un compte d'utilisateur mcve
- Efface un contexte d'imprimante
- Efface un domaine virtuel
- Efface un domaine virtuel
- Efface un dossier
- Efface un dossier ftp
- Efface un enregistrement dans une base dbase
- Efface un fichier
- Efface un fichier sur un serveur ftp
- Efface un noeud
- Efface un objet
- Efface un objet de grande taille
- Efface un objet pdf
- Efface un stylo
- 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 vecteur
- Efface une base de données msql

- Efface une base de données msql
- Efface une base de données mysql
- Efface une boîte aux lettres
- Efface une brosse
- Efface une entrée
- Efface une entrée dans un dossier
- Efface une fenêtre ncurses
- Efface une police
- Efface une transaction spécifiée dans une structure mcve_conn
- Efface une transaction spécifiée dans une structure mcve_conn
- Efface une valeur
- Efface une variable de la mémoire partagée.
- Effacer
- Effectue des actions de débogage
- Effectue la césure d'une chaîne
- Effectue une recherche
- Effectue une requête partielle pour l'uri spécifiée et renvoie toutes les informations.
- Effectue une rotation
- Effectue une selection
- Effectue une sous-requête apache
- Effectue une sous-requête nsapi
- Effectue une transaction avec payflow pro
- Effectue une transformation rot13
- Effectue une vérification de mot de passe
- Elève un nombre à la puissance n-ième.
- Emet les en-têtes http spécifiques à fdf
- Emet une commande return ou credit vers mcve
- Emet une résolution pour faire un dépôt batch
- Empile la matrice de transformation courante dans la pile.
- Empile un ou plusieurs éléments à la fin d'un tableau
- Empile un ou plusieurs éléments au début d'un tableau
- Enclenche la bufferisation de sortie
- Encode des entités html
- Encode les données binaires avant de les retourner d'une udf sqlite
- Encode les données de session
- Encode une chaîne en mime base64
- Encode une chaîne en url
- Encode une chaîne en url, selon la rfc1738
- Encode une chaîne pour une en-tête mime
- Encore non implémentée
- Enlève la marque d'effacement d'un message
- Enlève un objet
- Enregistre un filtre de flôt
- Enregistre un gestionnaire pour l'api de flôts
- Enregistre un nouveau format d'affichage des messages irc
- Enregistre un nouvel événement dans un calendrier mcal
- Enregistre une enveloppe url, implémentée comme classe php
- Enregistre une fonction de callback sur événement ibase
- Enregistre une fonction de complétion
- Enregistre une fonction exécutée à chaque tick
- Enregistre une fonction php avec une méthode
- 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 usine à filtres avec l'api des flôts
- Enregistre une variable dans une session
- Enregistrer plusieurs valeurs dans un paquet wddx
- Enregistrer une valeur dans un paquet wddx
- Enregistre une fonction php pour générer la documentation
- Envoi tout le contenu généré dans un fichier
- Envoie de mail
- Envoie des données à une socket connectée
- Envoie la commande site au serveur
- Envoie le buffer de sortie
- Envoie les données du buffer de sortie, et éteint la bufferisation de sortie
- Envoie un cookie
- Envoie un en-tête http
- Envoie un fichier mp3 en streaming
- Envoie un fichier sur le serveur ftp (non-bloquant)
- Envoie un mail encodé iso-2022-jp (mail japonais)
- Envoie un message à un canal ou un utilisateur
- Envoie un message à une socket, qu'elle soit orientée connexion ou pas
- Envoie un message dans une file
- Envoie un message mail
- Envoie un objet snmp
- Envoie un signal à un processus
- Envoie une chaîne à l'api de base muscat
- Envoie une chaîne au serveur postgresql
- Envoie une commande de vente (sale) à mcve
- Envoie une commande force à mcve. (généralement une autorisation par téléphone)
- Envoie une commande ftp brute
- Envoie une commande override à mcve
- Envoie une commande preauthorization à mcve
- 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 une socket, qu'elle soit connectée ou pas
- Envoie une note (notice) à un utilisateur
- Envoie une requête à la base frontbase
- Envoie une requête à sybase et ne bloque pas
- Envoie une requête à une base sybase
- Envoie une requête de ping à mcve
- Envoie une requête et lit tous les résultats
- Envoie une requête informix
- Envoie une requête msql
- Envoie une requête mysql à un serveur mysql
- Envoie une requête sql à un serveur ingres ii
- Envoie une requête sql à un serveur mysql
- Envoie une requête sql au serveur mssql
- Envoie une transaction brute à payflow pro
- Envoie votre animation au navigateur
- Envoye une requête à un serveur cyrus imap
- Et logique
- Etablit la connexion avec le système mcve
- Etablit une connexion à un serveur oracle
- Etablit une connexion postgresql persistante

- Etablit une nouvelle correction avec un serveur oracle
- Eteind l'alarme d'un événement
- Eteind une socket en lecture et/ou écriture
- Eteint la librairie payflow pro
- Etend ou condense une police de caractères
- Etire un objet en coordonnées globales
- Etire un objet relativement
- Exécute du code tcl coté 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 d'un curl
- Exécute un programme externe
- Exécute un programme externe et affiche le résultat brut.
- Exécute un programme externe et affiche le résultat.
- Exécute une chaîne comme un script ^{php}
- Exécute une commande analysée sur un pointeur oracle
- Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties
- Exécute une commande sql oracle
- Exécute une commande sur le serveur ftp
- 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 membres d'un tableau.
- Exécute une procédure stockée sur un serveur ms sql
- Exécute une requête
- Exécute une requête aql
- Exécute une requête ibase préparée
- Exécute une requête msql
- Exécute une requête ovrinos préparée
- Exécute une requête postgresql asynchrone
- Exécute une requête préparée
- Exécute une requête sesam
- Exécute une requête sql
- Exécute une requête sql
- Exécute une requête sql
- Exécute une requête sql avec sqlite et retourne un tableau
- Exécute une requête sql préparée.
- Exécute une requête sql préparée.
- Exécute une requête sql sans mobiliser les résultats
- Exécute une requête sqlite et lit le résultat
- Exécute une requête sqlite non bufferisée
- Exécute une requête sur un serveur frontbase
- Exécute une requête sur une base ibase
- Exécute une session curl
- Exponentielle
- Exporte un cert vers un fichier ou une variable
- Exporte un cert vers une variable
- Exporte un csr vers un fichier ou une variable
- Exporte un objet de grande taille vers un fichier
- Exporte une csr vers un fichier
- Expression exponentielle
- Expression régulière avec support des caractères multi-octets, sans tenir compte de la casse
- Expression régulière globale

- [Expression régulière posix pour les chaînes multi-octets](#)
- [Expression régulière standard](#)
- [Expression régulière 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. si la fonction de callback n'est pas spécifiée, le contenu sera envoyé à la sortie standard](#)
- [Extrait les informations d'une référence d'entrée](#)
- [Extrait tous les méta tags d'un fichier](#)
- [Extrait un fichier intégré dans un document fdf](#)
- [Extrait un objet](#)
- [Extrait une clé privée d'un certificat, et la prépare](#)
- [Extrait une clé publique d'un certificat](#)
- [Extrait une portion de tableau](#)
- 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 de fichiers dans un dossier](#)
- [Fait une pause](#)
- [Fausse documentation pour main](#)
- [Ferme l'accès au fichier](#)
- [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 connexion au serveur msession](#)
- [Ferme la connexion avec un serveur](#)
- [Ferme la connexion frontbase](#)
- [Ferme la connexion hyperwave](#)
- [Ferme la connexion mnogosearch avec le document enregistré](#)
- [Ferme la connexion mysql](#)
- [Ferme la page active](#)
- [Ferme la page pdf](#)
- [Ferme le chemin](#)
- [Ferme le document](#)
- [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 pointeur sur le dossier](#)
- [Ferme toutes les connexions odbx](#)
- [Ferme un blob ibase](#)
- [Ferme un bloc de mémoire partagée](#)
- [Ferme un document fdf](#)
- [Ferme un dossier d'archive](#)

- [Ferme un fichier](#)
- [Ferme un fichier bzip2](#)
- [Ferme un fichier compressé gzip](#)
- [Ferme un fichier pdf](#)
- [Ferme un flôt](#)
- [Ferme un flôt de dossier, et libère les ressources](#)
- [Ferme un lob oracle](#)
- [Ferme un objet de grande taille postgresql](#)
- [Ferme un objet slob](#)
- [Ferme un pointeur oracle](#)
- [Ferme un processus de pointeur de fichier](#)
- [Ferme un processus ouvert par proc_open\(\) et retourne le code de sortie du processus.](#)
- [Ferme une archive zip](#)
- [Ferme une base](#)
- [Ferme une base dbase](#)
- [Ferme une base de données dbm](#)
- [Ferme une connexion](#)
- [Ferme une connexion à un serveur informix](#)
- [Ferme une connexion à un serveur ingres](#)
- [Ferme une connexion à une base](#)
- [Ferme une connexion à une base de données interbase](#)
- [Ferme une connexion à une imprimante](#)
- [Ferme une connexion au serveur ldap](#)
- [Ferme une connexion ftp](#)
- [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 ovrmos](#)
- [Ferme une connexion sybase](#)
- [Ferme une connexion yaz](#)
- [Ferme une image dans un document pdf](#)
- [Ferme une relation](#)
- [Ferme une requête](#)
- [Ferme une session curl](#)
- [Ferme une socket](#)
- [Filtre les éléments d'un tableau](#)
- [Filtre pour ncurses](#)
- [Finalise et envoie une transaction](#)
- [Find out how many transactions of a given type are stored in the system](#)
- [Fixe d'offset d'un champ msql](#)
- [Fixe et lit différentes options d'assertions](#)
- [Fixe l'échelle horizontale du texte](#)
- [Fixe l'élévation du texte](#)
- [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'espacement des caractères](#)
- [Fixe l'espacement des mots](#)
- [Fixe l'identifiant de group de processus](#)
- [Fixe l'offset du pointeur de champ ms sql server](#)
- [Fixe l'uid effective du processus courant.](#)

- Fixe la catégorie de la structure globale
- Fixe la classe de la structure globale
- Fixe la couleur globale d'addition (? : the global add color).
- Fixe la couleur globale de multiplication (? : the global multiply color).
- Fixe la couleur pour le style courant de remplissage.
- 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 largeur de ligne
- Fixe la matrice du texte
- Fixe la page courante
- Fixe la platitude (flatness)
- Fixe la position du texte
- 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 la clé /status
- 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 fichier courant, ou la position courante
- Fixe le format de date pour les prochaines requêtes.
- Fixe le frame courant
- Fixe le gid effective du processus courant.
- Fixe le motif de pointillé
- Fixe le niveau de rapport d'erreurs php
- Fixe le niveau de sévérité des erreurs ms sql server
- Fixe le niveau de sévérité des messages d'erreurs ms sql server
- Fixe le niveau de seuil cubique (??)
- Fixe le nombre maximum d'image dans le sprite
- Fixe le paramètre linecap
- Fixe le paramètre linejoin
- Fixe le paramètre miter limit
- Fixe le point courant
- Fixe le point courant relativement
- Fixe le style courant de ligne
- Fixe le sujet d'un document pdf
- 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 limites d'un document pdf
- Fixe les mots clés d'un document pdf
- Fonction de callback pour la compression automatique des buffers
- Fonction de traitement des affichages web
- Fonction sans objet pour hyperwave

- Fonctionnement des expressions régulières.
- Force l'écriture de toutes les données compressées
- Force l'affichage lorsque ncurses_slk_noutrefresh est utilisé
- Force l'exécution d'une requête sur le maître dans une configuration maître/esclave (master/slave)
- Force l'exécution d'une requête sur un esclave dans une configuration maître/esclave (master/slave)
- Forks the currently running process
- 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.
- Free result memory
- Fusionne plusieurs tableaux
- **G**
-
- Générateur de congruence combinée linéaire
- Génère la syntaxe colorisée d'un fichier ^{php}
- Génère le code xml d'une requête de méthode
- Génère le code xml pour une valeur php
- Génère le contexte de débogage
- Génère un fichier temporaire et ouvre un flôt avec
- Génère un identifiant unique
- Génère un message dans l'historique système
- Génère une chaîne en encodage url
- Génère une clé
- Génère une clé privée et un 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 nouvelle clé privée
- Génère une valeur aléatoire
- Génère une valeur aléatoire (meilleure méthode)
- Gestion des colonnes de type long
- Gestionnaire de sortie pour maitriser le jeu de caractères de sortie
- Get text return value for previous function call
- Get_elements_by_tagname()
- **H**
-
- Homothétie
- **I**
-
- Identifiant d'objet de l'objet dans l'ancrage
- Identifiant d'objet des groupes fils
- Identifiant d'objet des parents
- Identifiants des ancrages d'un document
- Identification avec une connexion ftp
- Identifie un utilisateur
- Ids des documents fils d'un groupe
- Ignore les actions si le frame n'est pas chargé.
- Ignore les mots des moins de n caractères
- Importe les variables dans la table des symboles
- Importe les variables de get/post/cookie dans l'environnement global
- Importe un objet de grande taille depuis un fichier

- Imprime le texte à la ligne suivante
- Imprime les crédits de ^{php}
- Imprime un texte à la position courante
- Imprime un texte avec des options
- 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
- Incrémente le compteur de références
- 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 pré-lues par oracle
- Indique si il reste des lignes sqlite à lire
- Indique si l'internaute a abandonné la connexion http
- Indique si la fin d'un fichier compressé est atteinte
- Indique si le fichier a été téléchargé par http post
- Indique si le fichier est exécutable
- Indique si le fichier est un lien symbolique.
- Indique si le fichier est un véritable fichier.
- Indique si le nom de fichier est un dossier
- Indique si le script a expiré
- Indique si le thread safety est donné ou non
- Indique si les en-têtes http ont déjà été envoyés
- Indique si un algorithme fonctionne par blocs
- Indique si un fichier est autorisé en écriture
- 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}
- Indique un fichier est autorisé en lecture
- Informations à propos d'une connexion
- Initialise la librairie payflow pro
- Initialise la récupération d'un jeu de résultats
- Initialise la structure globale d'un flôt mcal
- Initialise le générateur de nombres aléatoires
- 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 nouveau pattern pdf
- Initialise un nouveau pointeur vide de lob/file

- Initialise un nouveau template pdf
- Initialise une nouvelle collection
- Initialise une nouvelle page de document pdf
- Initialise une procédure stockée ms sql server locale ou distante
- Initialise une session
- Initialise une session curl
- Initialise une structure pour stocker des données utilisateurs
- Initialise une valeur aléatoire (meilleure méthode)
- Initialises les fonctions de clés soft label
- Initialize ccvs for use
- Initilise un écran depuis un fichier de sauvegarde
- 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 object 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 entrée
- Insère une ligne et décale le reste de l'écran vers le bas
- Insère une valeur
- Insère uniquement des ancrs dans du texte
- Installe un gestionnaire de signaux
- Intègre des données binaires iptc dans une image jpeg
- Interroge l'autre extrémité de la communication
- Interroge la socket locale
- Inverse l'ordre des éléments d'un tableau
- Inverse modulo
- Inverse une chaîne
- J
-
- Joue l'animation flash à partir du frame courant.
- Joue un frame puis stoppe
- L
-
- 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
- La durée maximum de tentative de connexion à mcve
- La plus grand valeur aléatoire possible
- La plus grande valeur
- La plus petite valeur

- Le jour de l'année.
- Le jour de la semaine.
- Lecture binaire d'un fichier bzip2
- Lecture de fichier compressé binaire
- Lecture du fichier en mode binaire
- Lecture du pointeur de fiche courante (cursorname)
- Let et/ou modifie l'identifiant courant de session
- Libère la mémoire
- Libère la mémoire allouée pour ispell par mnogosearch
- Libère la mémoire associée à la connectivité ip/ssl
- Libère la mémoire associée à un résultat
- Libère la mémoire des ressources ms sql server
- 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 les ressources
- 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 occupées par une commande oracle
- Libère toutes les ressources réservées par un résultat oracle
- Libère un gestionnaire d'événements ibase
- Libère un résultat ibase
- Libère un résultat mnogosearch
- Libère un résultat sybase de la mémoire
- Libère un sémaphore
- Libère un segment de mémoire partagée
- Libère un verrou en écriture posé sur une relation
- Libère une commande ms sql server de la mémoire
- Lie des variables à une requête préparée
- Lie des variables à une requête préparée pour le stockage de résultat
- Lie un nom à une socket
- Lie une variable ^{php} à un paramètre oracle
- Lie une variable php à une variable sql d'oracle
- Linéarise une variable
- Lire la taille du bloc de mémoire partagée
- Lire le nom du groupe
- Lire un paquet wddx
- Liste ... (??)
- Liste de tous les utilisateurs du système mcve
- Liste des object ids des objets fils
- Liste des object records des objets fils
- Liste des utilisateurs actuellement identifiés
- Liste l'historique
- 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 bases de données msql sur un serveur
- Liste les boîtes aux lettres

- [Liste les boîtes aux lettres souscrites](#)
- [Liste les boîtes aux lettres, et retourne le détail pour chacune.](#)
- [Liste les champs d'un résultat frontbase](#)
- [Liste les champs d'une table](#)
- [Liste les champs d'une table](#)
- [Liste les champs du résultat mysql](#)
- [Liste les champs informix sql](#)
- [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 dossier dans un dossier](#)
- [Liste les filtres enregistrés](#)
- [Liste les fonctions d'une extension](#)
- [Liste les gestionnaires dba disponibles](#)
- [Liste les gestionnaires de flôts](#)
- [Liste les gestionnaires de transports de sockets disponibles](#)
- [Liste les paramètres des procédures](#)
- [Liste les procédure stockées](#)
- [Liste les processus mysql](#)
- [Liste les propriétés des champs sql](#)
- [Liste les sessions contenant une variable](#)
- [Liste les statistiques de tous les utilisateurs mcve](#)
- [Liste les tables d'une base de données](#)
- [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 tables msql sur une base de données](#)
- [Liste les types de données supportés par une source](#)
- [Liste tous les algorithmes de chiffrement supportés](#)
- [Liste tous les fichiers de bases de données 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 variables définies](#)
- [Lit certains paramètres](#)
- [Lit des données d'une socket](#)
- [Lit des données dans un fd, en utilisant le tableau scatter-gather définit par iovect_id](#)
- [Lit des données dans un résultat](#)
- [Lit des informations à propos de la dernière requête](#)
- [Lit des informations sur le fichier](#)
- [Lit des informations sur une colonne dans un résultat, et retourne un objet](#)
- [Lit des octets dans un fichier](#)
- [Lit des octets depuis un flôt dans un buffer](#)
- [Lit des options passés dans la ligne de commande](#)
- [Lit des paramètres numériques](#)
- [Lit des paramètres numériques dans le document pdf d'entrée](#)
- [Lit des paramètres textuels dans le document pdi](#)
- [Lit différentes options pour la connexion ftp courante](#)

- Lit directement le type de colonne oracle
- Lit et modifie les options des fonctions d'expression régulières à support de caractères multi-octets
- Lit et vérifie un fichier
- 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'échelle d'un champ
- Lit l'état du fichier ou de l'url
- Lit l'adresse de vérification (address verification) dans le statut retourné
- Lit l'apparence d'un champ
- Lit l'attribut suivant
- Lit l'en-tête du message
- Lit l'en-tête d'un message
- Lit l'encodage du client
- Lit l'heure locale
- Lit l'historique
- Lit l'identifiant de processus du serveur
- Lit l'identifiant système unique d'une transaction
- Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir (hue, white and blackness)
- Lit la clé suivante
- Lit la clé suivante
- Lit la configuration du cookie de session
- Lit la configuration interne de l'extension mbstring
- Lit la configuration locale
- Lit la dernière erreur générée par une socket
- 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 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 utilisée sur un dossier d'archive
- Lit la position courante du pointeur de lecture
- Lit la précision d'une colonne oracle
- Lit la première clé
- Lit la première clé
- 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 référence suivante
- Lit la réponse de l'api muscat
- 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'archive
- Lit la taille décompressée d'un dossier d'archive
- Lit la taille d'une colonne oracle
- Lit la taille d'une colonne oracle
- Lit la taille de chaque colonne d'un résultat
- 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 librairie ucd
- Lit la valeur d'un élément d'une collection oracle à l'index ndx
- Lit la valeur d'un champ msql
- Lit la valeur d'un propriété d'un composant com
- Lit la valeur d'un propriété d'un composant com
- Lit la valeur d'une colonne dans un résultat sqlite
- Lit la valeur d'une option de configuration
- 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 flôt/wrapper/contexte
- Lit la valeur rgb d'une couleur
- Lit la version courante du moteur zend
- Lit la version de lotus notes
- Lit la version des api mnogosearch
- 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 code retourné exact de la transaction
- 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 contenu du noeud
- 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 fichier suivant dans un dossier
- Lit le jeu de caractères courant
- 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 de notify
- Lit le message retourné par mcve ou par l'institution de traitement
- Lit le mime-type d'un type d'image
- Lit le nom d'un élément de code source
- 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 champ msql
- Lit le nom d'une colonne dans une réponse au format csv
- Lit le nom de l'utilisateur de la connexion
- Lit le nom de la base de données courante
- 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 qui contient un champ
- Lit le nom des sockets locale ou distante
- Lit le nom du champ fdf suivant
- Lit le nom du chiffrement utilisé
- Lit le nom du dossier d'archive
- Lit le nombre de champs dans un résultat
- Lit le nombre de ligne 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 d'autorisation de la transaction (alpha-numérique)
- Lit le numéro d'item d'une transaction dans un batch
- Lit le numéro de batch associé à la transaction
- Lit le numéro de version de l'api fdf
- Lit le numéro de version de la librairie xml
- Lit le numéro de version de payflow pro
- Lit le statut courant d'une base de données
- Lit le statut de cvc2/cvv2/cid
- Lit le statut de la connexion postgresql
- Lit le statut du buffer de sortie
- Lit le statut du cache associé au flôt
- Lit le statut du résultat
- Lit le tampon contenant le fichier pdf généré
- Lit le total des paiements non finalisés en mode batch
- Lit le type d'une colonne
- Lit len octets de données dans un blob ibase ouvert
- Lit les éléments par nom de balise
- Lit les événements souris
- Lit les attributs d'un champ fdf
- Lit les attributs d'un objet hyperwave
- Lit les attributs d'une entrée
- Lit les données ... (??)
- Lit les données d'un résultat
- Lit les données dans un fichier, applique l'encodage et envoie le résultat à destftp
- Lit les données dns associées à un hôte
- Lit les données du vecteur spécifié par iovec_id[iovec_position]
- Lit les données liées à une clé
- Lit les données méta de la table postgresql
- Lit les en-têtes et données méta des flôts
- Lit les en-têtes exif des images jpeg et tiff
- Lit les informations à propos de la boîte aux lettres courante.
- Lit les informations concernant un processus
- Lit les informations détaillant un transfert curl
- Lit les informations d'attribut
- Lit les informations d'entête d'une base dbase
- Lit les informations d'un champ sybase

- Lit les informations de configuration de l'imprimante
- 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 noms des bases de données
- Lit les options associé à une colonne de résultat
- Lit les options d'un analyseur xml
- Lit les options de socket
- Lit les paramètres de résultats mnogosearch
- Lit les quotas de chaque utilisateur
- Lit les quotas des boîtes aux lettres
- 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 la requête http
- Lit tous les en-têtes de réponse http
- Lit tous les en-têtes http de la requête
- Lit tous les entêtes de réponse http sur netscape serveur
- Lit tous les entêtes de requête http sur un serveur netscape
- Lit tout le fichier compressé
- Lit tout un fichier dans une chaîne
- Lit toute une ligne de résultat dans un tableau associatif
- Lit toute une ligne de résultat dans un tableau.
- 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 tableau, et la 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 bloc
- Lit un caractère dans un fichier
- Lit un caractère dans un fichier compressé
- Lit un caractère depuis le clavier (fenêtre)
- 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, 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 identifiant unique

- Lit un identifiant unique pour une relation
- Lit un message représentant la raison de l'échec de la connexion
- Lit un objet de grande taille
- Lit un objet de grande taille en totalité
- Lit un octet dans un flôt
- Lit un paramètre de réponse personnalisé
- Lit un paramètre de réponse utilisateur
- Lit un résultat asynchrone
- Lit un sommaire des en-têtes de messages
- Lit un tableau de ... (??)
- Lit une cellule spécifique dans une réponse au format csv, en utilisant le nom de colonne
- Lit une cellule spécifique dans une réponse au format csv, en utilisant le numéro de colonne
- 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 ligne
- Lit une ligne brute, au format csv retournée par mcve
- Lit une ligne dans un fichier compressé
- Lit une ligne dans un fichier compressé, et supprime les balises
- Lit une ligne dans un flôt
- Lit une ligne dans un résultat dbx ayant l'option `dbx_result_unbuffered` activée
- Lit une ligne dans un résultat ovrmos
- Lit une ligne dans un résultat ovrmos
- 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 données d'un flôt dans un buffer
- 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 mssql dans un tableau
- Lit une ligne de résultat mssql dans un tableau numérique
- Lit une ligne de résultat oracle dans un tableau
- Lit une ligne de résultat pgsql dans un objet
- Lit une ligne de résultat pgsql dans un tableau
- Lit une ligne de résultat pgsql 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 de résultats dans un tableau associatif
- Lit une ligne sous la forme d'un objet
- Lit une ligne sous la forme d'un tableau
- Lit une liste des paiements en attente en mode batch
- Lit une représentation exportable de la clé dans une chaîne ou un fichier
- Lit une représentation textuelle de `return_avs`
- Lit une représentation textuelle de `return_code`

- [Lit une représentation textuelle de return_cv](#)
- [Lit une sous-chaîne](#)
- [Lit une valeur](#)
- [Lit une valeur dans un résultat](#)
- [Lit une valeur dans un tableau de valeur d'un champ fd](#)
- [Lit une valeur dans une session](#)
- [Lit une valeur dans une structure usersetup](#)
- [Lit une variable dans la mémoire partagée.](#)
- [Lit/écrit la valeur courante d'une option](#)
- [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](#)
- [Logarithme en base 10](#)
- [Logarithme naturel \(népérien\)](#)
- [Look up an item of a particular type in the database #](#)
- M
- - [Mélange les éléments d'un tableau](#)
 - [Mélange les caractères d'une chaîne de caractères](#)
 - [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](#)
 - [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](#)
 - [Modifie certains paramètres](#)
 - [Modifie des informations dans la file de messages](#)
 - [Modifie l'échelle](#)
 - [Modifie l'échelle de la forme](#)
 - [Modifie l'épaisseur d'un trait](#)
 - [Modifie l'état du curseur](#)
 - [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'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 la brosse pour le dessin des lignes](#)
 - [Modifie la casse d'une chaîne](#)
 - [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 champs 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 champs 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 champs
- Modifie la position dans un objet de grande taille
- Modifie la position du pointeur de flôt
- Modifie la position du texte
- Modifie la position et dessine une ligne verticale avec le caractère donné, de taille maximale donnée
- Modifie la structure usersetup d'un utilisateur
- 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 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 courant
- 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 administrateur
- 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 pointeur de fichier
- 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
- Modifie le type xmlrpc, base64 ou datetime, pour une chaînes de caractères php
- Modifie les étiquettes de clé de fonction (function key labels)
- Modifie les attributs d'objet 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 données \(??\)](#)
- [Modifie les données du vecteur iovec_id\[iovec_position\]](#)
- [Modifie les droits d'un fichier via ftp](#)
- [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 champs texte flash](#)
- [Modifie les options](#)
- [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 numériques](#)
- [Modifie les paramètres odbc.](#)
- [Modifie un événement dans un calendrier mcal](#)
- [Modifie un bit](#)
- [Modifie un niveau de gris comme couleur de dessin et de remplissage.](#)
- [Modifie un objet](#)
- [Modifie un tableau de ... \(??\)](#)
- [Modifie un utilisateur dans une base de données de sécurité \(uniquement pour interbase6 ou plus récent\)](#)
- [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 propriété d'un composant com](#)
- [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](#)
- [Monitor a collection of files in a directory for changes](#)
- [Monitor a directory for changes](#)
- [Monitor a regular file for changes](#)
- [Multiplication de 2 nombres gmp](#)
- [Multiplie deux nombres de grande taille](#)
- [Multiplie la couleur de transformation](#)
- N
- - [Ncurses est en mode endwin, des affichages normaux sont possibles](#)
 - [Ncurses_putp \(???\)](#)
 - [Ncurses_slk_attroff](#)
 - [Ncurses_slk_attron](#)
 - [Ncurses_slk_attrset](#)
 - [Ncurses_vidattr](#)

- Ne convertit par les nouvelles lignes et retour chariot en saut de ligne
 - Ne vide pas les buffers sur reception des caractères de signaux
 - Nom de l'utilisateur actuellement identifié
 - Nom de la base de données
 - Nombre de colonnes dans un résultat
 - Nombre de colonnes retournées dans une ligne au format csv
 - Nombre de lignes dans un résultat
 - Nombre de lignes retournées dans une réponse au format csv
 - Nombre de transactions dans la queue client
 - Nombre gmp aléatoire
 - Nombre gmp probablement premier
 - Nomme le champs texte
 - Nomme le frame courant
 - Nomme un objet
 - Numéro d'erreur mnogosearch
 - Numéro de colonne
- O
-
- Object id de la racine
 - Object record de hw_document
 - Object records d'un groupe d'enfants
 - Obsolète : ajoute un signet dans la page courante
 - Obsolète : ajoute une annotation
 - Obsolète : configure l'échelle horizontale du texte
 - Obsolète : configure l'élévation de texte
 - Obsolète : configure l'espace entre deux mots
 - Obsolète : configure l'espacement de caractères character spacing
 - 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 : 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
 - Open connection to fam daemon
 - Opposé de nombre gmp
 - Optimise une base
 - Options disponibles pour les expressions régulières.
 - Ou exclusif logique
 - Ou logique

- Ouverture d'un fichier ou d'une url
- Ouvre des données scellées
- Ouvre la connexion à l'historique système
- Ouvre la connexion au serveur msession
- 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 spécifié
- Ouvre le module de l'algorithme et le mode à utiliser
- Ouvre un blob ibase pour récupérer des parties de données
- Ouvre un document fdf
- Ouvre un dossier pour le lister
- Ouvre un dossier, et récupère un pointeur dessus.
- Ouvre un fichier compressé avec bzip2
- Ouvre un fichier compressé avec gzip
- Ouvre un fichier d'archive en lecture
- Ouvre un fichier de relation
- Ouvre un fichier de relation local
- Ouvre un fichier pdf
- Ouvre un flôt ^{imap} vers une boîte aux lettres
- Ouvre un flôt avec un fichier ou une url, en spécifiant le contexte
- Ouvre un flôt avec un fichier ou une url, et le convertit en pointeur file*
- Ouvre un flôt imap vers une nouvelle boîte aux lettres.
- Ouvre un flôt sur un fichier ou une url
- Ouvre un nouveau dictionnaire cracklib dictionary
- Ouvre un nouveau document pdf
- 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 file* avec la fonction tmpfile() et la convertit en flôt
- Ouvre un pointeur oracle
- Ouvre une archive zip
- Ouvre une base dbase
- Ouvre une base de données
- Ouvre une base de données dbm
- Ouvre une base sqlite
- 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 connexion avec un serveur, et retourne un flôt
- Ouvre une connexion avec une socket unix et la convertit en flôt
- Ouvre une connexion ftp
- Ouvre une connexion ftp sécurisée avec ssl
- Ouvre une connexion hyperwave
- Ouvre une connexion mcal
- Ouvre une connexion mnogosearch avec un document stocké
- Ouvre une connexion msql

- [Ouvre une connexion oracle](#)
- [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 msql](#)
- [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 persistante mcal](#)
- [Ouvre une connexion postgresql](#)
- [Ouvre une connexion sesam](#)
- [Ouvre une connexion socket internet ou unix](#)
- [Ouvre une connexion sqlite persistante](#)
- [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](#)
- [Ovre un nouveau fichier](#)

• P

•

- [Paramètre un hyper lien](#)
- [Pas de documentation](#)
- [Pas encore implémenté](#)
- [Pas encore implémentée](#)
- [Pas encore implémentée](#)
- [Pas encore implémentée](#)
- [Pas encore implémentée](#)
- [Passe à l'image suivante](#)
- [Passe en revue une carte, et appelle une fonction à chaque entrée](#)
- [Passe le terminal en mode brut \(raw\)](#)
- [Passe le terminal en mode cooked](#)
- [Passe une fenêtre en mode standout](#)
- [Perform a full reversal on a completed transaction](#)
- [Perform a full reversal on an already-processed authorization](#)
- [Perform credit authorization test on a transaction](#)
- [Performe une requête aql locale](#)
- [Performe une requête sur la base de données](#)
- [Performs a command which is peculiar to a single protocol, and thus is not available in the general ccvs api](#)
- [Pgcd](#)
- [Pgcd étendu](#)
- [Ping la connexion au serveur et reconnecte si elle n'existe plus](#)
- [Pingue la connexion à la base](#)
- [Pingue le serveur, et se reconnecte au besoin](#)
- [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 page
- Place une page dans le document
- 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
- Précise les dossiers à fouiller pour rechercher des polices externes
- Prépare à la lecture (z39.50 present).
- Prépare et exécute une requête sql
- Prépare l'affichage pour le html
- Prépare la connexion courante pour l'affichage
- Prépare le client pour la connexion ip/ssl
- Prépare le module pour le téléchargement
- Prépare les caractères spéciaux d'un nick pour être compatible irc
- Prépare un scan
- Prépare une chaîne pour une recherche par expression régulière insensible à la casse.
- Prépare une clé privée au format pem
- Prépare une commande pour l'exécution
- Prépare une connexion à un hôte yaz
- Prépare une expression régulière 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 ovrinos
- 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 transaction ibase
- Prend une ou plusieurs valeurs, au hasard dans un tableau
- Protège les caractères d'une chaîne binaire en mode bytea
- 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
- 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 la passer à mysql_query()
- Puissance
- Puissance et modulo
- Quitte le canal

• Q
•

• R
•

- Réactive l'ancienne fonction de gestion des erreurs
- Récupère la miniature d'une image tiff ou jpeg
- Récupère les données métas d'un champ unique
- Récupère un ienumvariant
- Récupère un résultat à partir d'une requête préparée dans les variables
- Récupère une ligne de résultat dans un objet
- Récupère une ligne de résultat dans un 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
- Réduit itérativement un tableau
- Réouvre une connexion mcal
- Répète une chaîne
- Réserve un sémaphore
- Résolution dns d'une adresse ip
- 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 une fichier de sauvegarde
- Racine carrée
- Racine carrée avec reste gmp
- Racine carrée gmp
- Rafrâichit l'écran virtuel pour prendre en compte les relations entre les conteneurs de la pile
- Rafrâichit l'écran
- Rafrâichit la fenêtre sur l'écran du terminal
- Rapporte le message d'erreur du dernier appel de fonction
- Rassemble des informations sur la langue et la configuration locale.
- Rassemble les éléments d'un tableau en une chaîne
- Rassemble les éléments d'un tableau en une chaîne
- Reçoit des données d'une socket connectée
- Reçoit des données d'une socket, connectée ou pas
- Reçoit des messages sur uen socket, qu'elle soit orientée connexion ou pas
- Reçoit tous les objets ^{snmp} d'un agent
- Reçoit un message depuis une file de messages
- Reçoit un objet ^{snmp}
- Recheche des chemins qui vérifie un masque
- Recherche 0
- Recherche 1
- Recherche dans un niveau
- Recherche dans un tableau la clé associée à une valeur
- Recherche des objets
- Recherche par expression régulière avec support des caractères multi-octets
- Recherche par expression régulière insensible à la casse
- Recherche par expression régulière multi-octets
- Recherche sur le serveur ldap
- Recherche un élément avec son identifiant
- Recherche un événement dans le calendrier
- 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
- Rechercher et remplacer par expression régulière standard
- Rechercher/remplacer avec une expression régulière et fonction de callback
- Recode de fichier à fichier, en fonction de la requête.
- Recode une chaîne en fonction de la requête
- Recode une fonction grâce à une requête
- Recommence la lecture du fichier
- Recule d'un frame
- Recule le pointeur courant de tableau
- Referme le dictionnaire cracklib
- Regular expression match ignoring case with multibyte support
- Rejoint un canal irc
- Relance la connexion au serveur postgresql
- Remet à zéro l'environnement graphique pdf
- Remet à zéro la session courante
- Remet le pointeur interne de tableau au début
- Remplace des chaînes dans une chaîne
- Remplace des segments de chaînes, avec le support des expressions régulières mutli-octet
- 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 ou insère une entrée
- Remplace toutes les occurrences dans une chaîne
- Remplace un attribut dans l'entrée courante
- Remplace un enregistrement dans une base dbase
- Remplace un noeud domxml
- Remplace un noeud fils
- Remplace un objet
- Remplace un segment dans une chaîne
- Remplace une valeur
- Remplacement par expression régulière
- Remplacement par expression régulière insensible à la casse.
- Remplis un tableau avec une même valeur
- Remplit
- Remplit avec une région avec une couleur spécifique
- Remplit et passe le pinceau sur le chemin pdf 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
- Renomme un calendrier
- Renomme un fichier
- Renomme un fichier sur un serveur ftp
- Renomme une boîte aux lettres
- Renomme une relation
- Renvoie l'espace disque disponible dans le répertoire.
- Renvoie l'heure à laquelle l'inode a été accédé pour la dernière fois.

- Renvoie la date à laquelle le fichier a été accédé pour la dernière fois.
- Renvoie la date de dernière modification du fichier.
- Renvoie la ligne courante et cherche les champs csv
- Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier et élimine les balises html
- 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 du 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 à propos d'un fichier
- Renvoie les informations à propos d'un lien
- Renvoie les permissions affectées au fichier.
- Renvoie une chaîne en majuscules
- Renvoie une chaîne en minuscules
- 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
- Remplace le flût de dossier au début
- Remplace le pointeur de fichier au début
- Remplace un caractère dans la queue d'entrée
- Représente un id global en un id virtuel local
- Reprend le téléchargement d'un fichier (non bloquant)
- Requiert les informations sur un utilisateur
- Recherche la position d'une occurrence dans une chaîne, sans tenir compte de la casse
- 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 ar def_shell_mode
- Restaure un environnement
- Reste de la division de deux nombres gmp
- Resume suspended monitoring
- Retarde l'affichage sur les terminaux utilisant des caractères de remplissage
- Retarde l'exécution
- 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
- Retour le conteneur sous le conteneur. si ce dernier est null, le premier conteneur de la pile est retourné
- Retourne ^{true} si le code de retour représente une fin normale
- Retourne ^{true} si un objet a pour parent une classe donnée
- Retourne à la première entrée du dossier
- Retourne chaque paire clé/valeur d'un tableau
- Retourne des détails sur un calendrier
- Retourne des détails sur une colonne mysql
- Retourne des information sur la configuration du serveur
- Retourne des informations à propos d'un paramètre dans une requête préparée ibase
- Retourne des informations à propos de la dernière requête exécutée
- 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 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](#)
- [Retourne des informations à propos de la librairie gd installée](#)
- [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'échelle d'un champ](#)
- [Retourne l'élément courant d'un tableau](#)
- [Retourne l'état de la dernière requête sesam](#)
- [Retourne l'adresse ip correspondant à un hôte](#)
- [Retourne l'arbre de reference organisé par thread](#)
- [Retourne l'attribut de la clé soft label courante](#)
- [Retourne l'en-tête d'un message](#)
- [Retourne l'encodage utilisé par la librairie sqlite](#)
- [Retourne l'ensemble optimal de colonnes, qui permet de définir uniquement une ligne dans une table](#)
- [Retourne l'heure actuelle](#)
- [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 note non lue pour l'utilisateur user_name](#)
- [Retourne l'identifiant du dernier objet](#)
- [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 courant](#)
- [Retourne l'identifiant du thread pour la connexion courante](#)
- [Retourne l'identifiant généré par la dernière requête insert.](#)
- [Retourne l'identifiant maximal de hash](#)
- [Retourne l'identifiant public du type de document](#)
- [Retourne l'identifiant système du type de document](#)
- [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'inode du script](#)
- [Retourne l'objet propriétaire d'une ancre](#)
- [Retourne l'offset du début du prochain segment repéré par une expression régulière](#)
- [Retourne l'option d'actino pour keypress \(char\)](#)
- [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'url d'une animation shockwave flash](#)

- Retourne la chaîne correspondante à l'erreur donnée
- Retourne la cible du script
- Retourne la clé d'un attribut
- Retourne la classe d'un objet
- 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/heure
- Retourne la dernière erreur (si elle existe) qui est survenu lors de la dernière requête.
- Retourne la dernière erreur de `stmt|conn|global` sur oracle
- Retourne la dernière note du serveur postgresql
- Retourne la description d'une raison
- Retourne la description du terminal
- Retourne la destination d'une ancre source
- Retourne la fenêtre associée avec un conteneur
- Retourne la fin de la chaîne
- Retourne la hauteur d'une bitmap
- Retourne la hauteur d'une image
- 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
- Retourne la largeur d'une chaîne
- Retourne la largeur d'une image
- 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 du caractère actuellement supprimé
- Retourne la ligne suivante d'un résultat oracle dans un tableau
- 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 leur valeur
- 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 imprimantes attachées à un serveur
- Retourne la liste des notations
- Retourne la liste des utilisateurs connectés
- Retourne la longueur d'un champ `mysql`
- Retourne la longueur d'une chaîne
- Retourne la longueur du contenu du buffer de sortie
- Retourne la méthode de compression utilisée avec `gzip`
- Retourne la plus grande valeur d'une collection oracle
- Retourne la position courante dans un objet de grande taille
- Retourne la position courante du curseur pour une fenêtre

- 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 régulière
- Retourne la précision d'un champ
- Retourne la première entrée
- Retourne la première raison d'erreur
- Retourne la prochaine occurrence d'un événement
- Retourne la somme de contrôle crc32 d'une chaîne
- Retourne la table de traduction des entités utilisée par [htmlspecialchars](#) et [htmlentities](#)
- 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 dossier
- Retourne la taille d'un fichier.
- Retourne la taille d'une chaîne
- Retourne la taille d'une collection oracle
- Retourne la taille d'une colonne 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
- Retourne la taille de la chaîne
- Retourne la taille de la chaîne
- Retourne la taille de la clé d'un chiffrement
- 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 d'un attribut
- Retourne la valeur d'un attribut
- 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'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 hw
- 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 version courante de curl
- Retourne la version de la librairie sqlite
- Retourne la version du protocole mysql utilisé
- Retourne la version du serveur mysql
- Retourne le baudrate du terminal
- Retourne le chemin canonique absolu
- Retourne le chemin du terminal
- Retourne le code ascii d'un caractère
- Retourne le code d'erreur
- Retourne le code d'erreur de la dernière opération fd
- 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 php utilisé pour générer une variable
- Retourne le conteneur au dessus du conteneur. si ce dernier est null, retourne le dernier conteneur de la pile
- 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 obtenu avec l'option `curl_opt_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 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
- Retourne le fichier courant, ou la position courante
- Retourne le fils d'un objet
- Retourne le flag d'un champ msq
- Retourne le frère précédent d'un noeud
- Retourne le gid du propriétaire du script
- Retourne le jeu de caractères courant pour la connexion
- Retourne le jeu de caractères courant pour les expressions régulières
- Retourne le logo de zend
- Retourne le message d'erreur de la dernière opération qtdom, ou `false` si aucune erreur n'est survenue
- Retourne le message d'erreur bzip2
- Retourne le message d'erreur courant
- Retourne le message d'erreur de la dernière opération ircg
- 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 msq
- 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 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 élément
- 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
- Retourne le nom d'une colonne oracle
- Retourne le nom d'une colonne ovrinos
- Retourne le nom d'une table à partir d'un nom de champ mysql
- Retourne le nom d'une table à partir d'un nom de champs
- Retourne le nom de device du terminal
- Retourne le nom de l'algorithme
- Retourne le nom de l'attribut
- 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 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
- Retourne le nom du champ sqlite
- Retourne le nom du curseur ovrinos
- 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 courant de colonne d'un analyseur xml.
- Retourne le nombre d'arguments passé à la fonction
- Retourne le nombre d'attributs
- Retourne le nombre d'avertissements généré par la dernière requête
- Retourne le nombre de champ
- Retourne le nombre de champs d'un résultat mysql
- Retourne le nombre de champs dans un résultat
- 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 dans une base filepro.

- Retourne le nombre de champs renvoyés par la dernière requête.
- Retourne le nombre de colonne dans un résultat
- Retourne le nombre de colonnes
- Retourne le nombre de colonnes
- Retourne le nombre de colonnes dans un résultat
- Retourne le nombre de colonnes dans un résultat ibase
- 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 jour dans un mois
- Retourne le nombre de jours dans un mois, pour une année donnée
- Retourne le nombre de jours entre le 21 mars et pâques, pour une année donnée.
- Retourne le nombre de ligne affectées
- Retourne le nombre de ligne affectées par la dernière opération mysql
- Retourne le nombre de ligne affectées par une requête ms sql server
- Retourne le nombre de lignes affectées
- Retourne le nombre de lignes affectées lors de la dernière requête sql
- Retourne le nombre de lignes affectées ou retournées par la dernière requête.
- 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 oracle
- Retourne le nombre de lignes affectées par une modification ovrimos
- Retourne le nombre de lignes affectées par une requête.
- Retourne le nombre de lignes d'un résultat
- Retourne le nombre de lignes d'un résultat sqlite
- Retourne le nombre de lignes dans un résultat
- Retourne le nombre de lignes dans un résultat
- Retourne le nombre de lignes dans un résultat
- Retourne le nombre de lignes dans un résultat mssql
- Retourne le nombre de lignes dans un résultat sybase
- Retourne le nombre de lignes postgresql
- Retourne le nombre de lignes qui ont été modifiées par la dernière requête sqlite
- 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 dans une requête préparée ibase
- Retourne le nombre de paramètres pour la requête donnée
- Retourne le nombre de résultats de la dernière recherche
- Retourne le nombre de raisons d'erreurs
- Retourne le nombre de transactions dans la queue d'autorisation, ainsi que la liste des identifiants
- Retourne le numéro d'erreur
- Retourne le numéro d'erreur ^{xslt} courant
- Retourne le numéro d'erreur bzip2
- 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'ordre d'une carte
- Retourne le numéro d'une colonne
- Retourne le numéro de colonne ovrimos
- Retourne le numéro de frame courant
- 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
- Retourne le numéro de protocole associé à un nom de protocole
- Retourne le numéro de séquence de message pour un uid donné.
- Retourne le numéro de semaine dans l'année pour une date donnée
- 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 père d'un noeud dom xml
- Retourne le parent d'un objet
- Retourne le préfixe d'espace de nom d'un noeud
- 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 identifiant d'objet libre
- Retourne le prochain lot de lignes ms sql server
- Retourne le propriétaire d'un objet
- Retourne le résultat d'un scan
- Retourne le résultat de $\exp(\text{number}) - 1$, calculé de manière précise, même si val est proche de 0
- Retourne le résultat de $\log(1 + \text{number})$, calculé de manière précise, même si val est proche de zéro.
- Retourne le résultat de $\sqrt{\text{num1}*\text{num1} + \text{num2}*\text{num2}}$
- 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 segment de chaîne trouvé par une expression régulière multi-octets
- Retourne le service internet qui correspond au port et protocole.
- Retourne le sid du processus
- Retourne le sous ensemble interne (internal subset)
- Retourne le statut courant du serveur
- Retourne le statut de la chaîne
- Retourne le texte associé avec l'erreur générée lors de la dernière requête.
- Retourne le timestamp unix actuel avec microsecondes
- Retourne le timestamp unix actuel.
- 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 utilisé 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 de champs mysql
- Retourne le type de commande oci
- Retourne le type de document
- Retourne le type de données d'une colonne oracle
- Retourne le type de fichier
- Retourne le type de la colonne courante
- 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 ressource
- Retourne le type mime
- Retourne le type numérique d'une colonne ovrimos
- Retourne le type xmlrpc d'une valeur php. particulièrement pratique pour les types chaînes de type base64 et datetime
- 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 status de la connexion http
- Retourne les données de résultat
- Retourne les données du script
- Retourne les données enregistrées dans une colonne sous forme d'objet.
- 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 à propos d'un fichier ou d'un lien symbolique.
- 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 une socket
- Retourne les limites système
- Retourne les longueurs des colonnes de la ligne courante du 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 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 attributs d'une classe.
- Retourne plus de détails après une erreur
- 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 résultat oracle
- Retourne toutes les options supportées par le terminal, combinées ensemble par l'opérateur or
- Retourne toutes les valeurs de l'attribut sous la forme d'un tableau de chaînes
- Retourne un élément de la liste des arguments
- Retourne un analyseur destiné au traitement d'un message
- Retourne un caractère spécifique
- Retourne un champ d'un résultat
- Retourne un code d'erreur pour la dernière commande mysql
- 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 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, spécifie le nom de la note.
- Retourne un identifiant de sémaphore
- Retourne un identifiant de type de serveur ftp.
- Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha
- Retourne un message d'erreur
- Retourne un message d'erreur
- Retourne un nouveau curl multiple
- Retourne un nouveau pointeur à utiliser pour lier les pointeurs de références
- Retourne un objet contenant la structure de date
- Retourne un objet de classe hw_api
- 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 décrivant les classes agrégées
- 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 noms des fichiers qui sont inclus dans un script
- Retourne un tableau avec les résultat de la recherche
- Retourne un tableau contenant les tailles de clés acceptées par un algorithme
- Retourne un tableau contenant les tailles de clés supportée par un algorithme de chiffrement
- Retourne un tableau contenant toute la carte
- Retourne un tableau d'objets représentant les champs dans le résultat
- Retourne un tableau de message après recherche.
- Retourne un timestamp unix pour pâques, à minuit
- 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 utilisé
- Retourne une chaîne contenant les informations de version du serveur oracle
- Retourne une chaîne décrivant la dernière erreur
- Retourne une chaîne décrivant un message d'erreur
- Retourne une chaîne formatée
- Retourne une chaîne formattée
- Retourne une clé d'un tableau associatif
- Retourne une description de l'erreur
- Retourne une donnée d'une ligne de résultat oracle
- 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 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 objet
- 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'événement entre deux dates
- Retourne une liste d'événements qui ont une alarme prévue à une date
- Retourne une ressource sur une section d'un message mime
- Retourne une section extraite du corps d'un message
- Return first attribute
- Return first reference
- Return the status of the background communication process
- Returns ^{true} if child process is currently stopped
- Returns ^{true} if status code represents a termination due to a signal
- Returns the signal which caused the child to stop
- Returns the signal which caused the child to terminate
- Rotation de la transformation courante
- Runs the equivalent of the select() system call on the given arrays of streams with a timeout specified by tv_sec and tv_usec
- S
-
- Sélectionne la base de données ms sql
- Sélectionne la forme de départ
- Sélectionne la forme de fin
- Sélectionne la largeur et hauteur du champs
- 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 msq1
- Sélectionne une base de données msq1
- Sélectionne une base de données mysql
- Sélectionne une base de données par défaut 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
- S'identifie auprès du serveur hyperwave
- S'identifie sur un serveur cyrus imap
- Sauve dans un fichier
- Sauve l'état du terminal
- Sauve l'environnement courant
- Sauve l'environnement graphique courant
- 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 un fichier lob oracle
- Sauve un lob oracle
- Sauve une clé au format ascii dans un fichier
- Sauver un document fdf
- Scanne les données du fichier et extrait tous les fichiers encodés qui s'y trouvent. retourne un tableau contenant la liste des fichiers.

- Scelle des données
- Scinde un code source en éléments de base
- Scinde une chaîne
- Scinde une chaîne en tableau avec une expression régulière multi-octets
- Scinde une chaîne en un tableau, grâce à une expression régulière
- Scinde une chaîne en un tableau, grâce à une expression régulière.
- Scrolle le contenu de la fenêtre vers le haut ou le bas, dans changer la position courante
- Se connecte à un serveur cyrus imap
- Se connecte à un serveur ldap
- Set the effective uid of the current process
- Signe du nombre gmp
- Signe les données
- Signe un certificat avec un autre certificat
- Signe un message s/mime
- Sinus
- Sinyus hyperbolique
- Souscrit à une boîte aux lettres
- Soustraction de 2 nombres gmp
- Soustrait un nombre de grande taille à un autre.
- Spécifie la base d'une session yaz
- Spécifie la nouvelle clé primaire d'une relation
- Spécifie la syntaxe de lecture des lignes
- Spécifie le fichier de clé de certificats si le serveur requiert la vérification des certificats du client
- Spécifie le jeu de caractères utilisé pour les messages du domaine domain
- Spécifie le nombre maximal de résultat à lire
- Spécifie le schema 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
- Stocke un message d'erreur
- Stoppe une base de données fbsql
- Substitute xincludes dans un objet domdocument
- Suggère l'orthographe d'un mot [obsolète]
- Suggère une orthographe
- 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 les anti-slash d'une chaîne c
- Supprime les balises html et ^{php} d'une chaîne
- Supprime les caractères invisibles de début de chaîne
- Supprime les espaces de fin de chaîne
- Supprime les espaces en début et fin de chaîne
- Supprime num éléments à la fin de la collection oracle
- Supprime tous les événements marqués pour l'effacement
- Supprime toutes les lignes d'une relation
- Supprime un attribut
- Supprime un attribut hyperwave
- Supprime un fils de la liste des noeuds fils
- Supprime un flag sur un message
- 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 collection oracle](#)
- [Supprime une couleur d'une image](#)
- [Supprime une fonction de callback sur une connexion cyrus imap](#)
- [Supprime une structure usersetup](#)
- [Supprime une usine à filtres des api de flôts](#)
- [Supprime une variable de la session](#)
- [Supprimer les anti-slash d'une chaîne](#)
- [Symbole de jacobi](#)
- [Symbole de legendre](#)
- [Synchronise avec le serveur postgresql](#)
- [Synchronise une base de données](#)
- [Synonyme de checkdnsrr](#)
- [Synonyme de getmxrr](#)
- [Synonyme de odbc_exec](#)
- T
- - [Télécharge un fichier depuis un serveur ftp](#)
 - [Télécharge un fichier via ftp dans un flôt local](#)
 - [Taille d'un document](#)
 - [Tangente](#)
 - [Tangente hyperbolique](#)
 - [Temporarily suspend monitoring](#)
 - [Terminate ccvs engine and do cleanup work](#)
 - [Terminate monitoring](#)
 - [Termine l'action courante](#)
 - [Termine la définition de symbole](#)
 - [Termine la définition du bouton 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 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 chiffrage](#)
 - [Termine un document](#)
 - [Termine un flôt imap](#)
 - [Termine un jeu de transferts curl](#)
 - [Termine un processus ouvert par proc_open\(\)](#)
 - [Termine une connexion mcve_conn](#)
 - [Termine une connexion postgresql](#)
 - [Termine une page](#)
 - [Termine une section de texte](#)

- Teste un module ouvert
- Teste la fin du fichier
- Teste le chiffage par blocs d'un algorithme
- Teste le chiffage 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
- Tourne la forme
- Tourne un objet en angle absolu
- Traite un fichier de configuration
- Transfère un jeu de résultats à partir de la dernière requête
- Transfer funds from the credit card holder to the merchant
- Transfer funds from the merchant to the credit card holder
- Transforme des données `xml` avec `xslt`
- Transforme les coordonnées
- Transforme les coordonnées d'une fenêtre
- 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 variable en tableau
- Translate l'origine du système de coordonnées
- Translate la transformation courante
- Tri d'un tableau avec l'algorithme à "ordre naturel"
- Tri d'un tableau avec l'algorithme à "ordre naturel" insensible à la casse
- Tri multi-dimensionnel
- Tri un résultat avec une fonction utilisateur
- Trie des messages
- Trie en ordre inverse
- Trie les entrées d'un résultat ldap
- Trie un tableau
- Trie un tableau en ordre
- Trie un tableau en ordre inverse
- Trie un tableau en sens inverse et suivant les clés
- Trie un tableau en utilisant une fonction de comparaison personnalisée
- Trie un tableau en utilisant une fonction de comparaison personnalisée
- Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur
- Trie un tableau suivant les clés
- Tronque un fichier
- Tronque un fichier
- Tronque une chaîne
- Trouve la position d'un caractère dans une chaîne
- 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 première occurrence d'un caractère dans une chaîne
- Trouve la première occurrence dans une chaîne
- Trouve la première occurrence dans une chaîne (insensible à la casse)
- Trouve le premier segment de chaîne
- Trouve un segment de chaîne ne contenant pas certains caractères
- Trouve une valeur

- Type de données d'un champ
- U
 -
 - Utilisation des ressources
 - Utilise la version de swf (??)
 - Utilise un analyseur xml à l'intérieur d'un objet.
 - Utilise une variable php pour la phase de définition, dans une commande select oracle
 - Utiliser ou non les fonctions d'antialias
- V
 -
 - Vérifie si le terminal peut changer de couleurs
 - Vérifie l'état de la connexion
 - Vérifie l'existence d'une constante
 - Vérifie l'usage d'un certificat
 - Vérifie la connexion mnogosearch avec document stocké
 - Vérifie la présence des touches de fonctions sur le clavier
 - Vérifie la signature d'un message s/mime
 - Vérifie le courrier de la boîte aux lettres courante.
 - Vérifie les capacités d'insertion et d'effacement
 - Vérifie les capacités d'insertion et d'effacement
 - 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 faite caractères blancs
 - Vérifie qu'une chaîne est imprimable
 - Vérifie qu'une chaîne est imprimable
 - Vérifie qu'une chaîne est un entier
 - Vérifie qu'une clé existe
 - Vérifie qu'une classe a été définie
 - Vérifie qu'une réponse est au format mvce
 - Vérifie que l'année est bissextile.
 - Vérifie que la méthode existe pour une classe.
 - Vérifie que le flôt imap est toujours actif
 - Vérifie que le pointeur n'est pas au bout du flôt
 - Vérifie que le que le terminal supporte les couleurs
 - Vérifie si la connexion postgresql est occupée
 - Vérifie si la transaction a réussi
 - Vérifie si le buffer de sortie est vide ou pas
 - Vérifie si plusieurs résultats sont disponibles
 - Vérifie si un attribut est éditabile
 - Vérifie si un attribut est spécifié
 - Vérifie si un attribut existe
 - Vérifie si un fichier existe
 - Vérifie si un jeu de caractère 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 une assertion est fausse
 - Vérifie si une clé existe dans un tableau

- Vérifie si une clé privée correspon à un cert
- Vérifie si une transaction a été complétée
- 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
- Va à la prochaine image du sprite
- Valeur absolue
- Valeur absolue gmp
- Valide la transaction courante
- Valide la transaction sesam en cours
- Valide le mot de passe d'un utilisateur pour un domaine. retourne ^{true}/_{false}
- Valide les transactions oracle en cours
- Valide une date grégorienne
- Valide une date.
- Valide une heure.
- 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
- Verifie le mot de passe
- Verrouille le fichier
- Verrouille un objet
- Verrouille une session
- Version insensible à la casse de ^{str} replace
- Version plurielle de dcgettext
- Version plurielle de dgettext
- Version plurielle de gettext
- Vide le buffer d'entrée du clavier
- Vide les buffers après détection d'un caractère de signal
- Vide les buffers dans le flôt
- Vide les buffers de sortie

12.3 Index des exemples

- A
- - ♦ Exemple avec abs
 - ♦ Accéder aux données du formulaire
 - ♦ Accès aux membres d'une classe
 - ♦ Accès aux valeurs rgb
 - ♦ Accs aux données de formulaire fdf
 - ♦ Activer l'extension bzip2 sous windows
 - ♦ Exemple avec addaction.php">swfbutton->addaction
 - ♦ exemple avec addslashes et charlist (2)
 - ♦ Exemple avec addfill.php">swfshape->addfill
 - ♦ Addition de tableaux
 - ♦ Affectation d'une colonne de type "champs multiple"

- ◆ [Affichage d'adresse ip](#)
- ◆ [Affichage de texte avec cpdf_begin_text](#)
- ◆ [Affichage de texte avec cpdf_end_text](#)
- ◆ [Affichage des fichiers inclus et requis](#)
- ◆ [Affichage sous la forme d'une table html](#)
- ◆ [Affiche la liste des attributs d'une entrée](#)
- ◆ [Afficher des données issues d'un formulaire](#)
- ◆ [Afficher des informations d'exécution.](#)
- ◆ [Afficher le contenu d'une base avec dbplus](#)
- ◆ [Afficher les informations d'entête d'une base dbase](#)
- ◆ [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.](#)
- ◆ [Ajouter du code javascript à un document fdf](#)
- ◆ [Ajouter un élément dans un tableau associatif](#)
- ◆ [Ajouter un élément dans un tableau indexé](#)
- ◆ [Ajouter un noeud fils avec dom xml \(exemple 3\)](#)
- ◆ [Ajouter un noeud fils dom xml](#)
- ◆ [Ajouter une mise en relief](#)
- ◆ [Alternatives à scandir](#)
- ◆ [Analyser le résultat de socket_select](#)
- ◆ [Annulation d'une transaction sesam](#)
- ◆ [Exemple avec apache_request_headers](#)
- ◆ [Appeler des fonctions définies par l'utilisateur.](#)
- ◆ [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_fill](#)
- ◆ [Exemple avec array_filter](#)
- ◆ [Exemple avec array_flip : collision](#)
- ◆ [Exemple avec array_intersect](#)
- ◆ [Exemple avec array_intersect_assoc](#)
- ◆ [Exemple avec array_key_exists](#)
- ◆ [Exemple avec array_merge](#)
- ◆ [Exemple avec array_merge_recursive](#)
- ◆ [Exemple avec array_pad](#)
- ◆ [Exemple avec array_pop](#)
- ◆ [Exemple avec array_push](#)
- ◆ [Exemple avec array_rand](#)
- ◆ [Exemple avec array_reduce](#)
- ◆ [Exemple avec array_reverse](#)
- ◆ [Exemple avec array_shift](#)
- ◆ [Exemple avec array_sum](#)
- ◆ [array_unique et les types de valeurs](#)
- ◆ [array_values pour les utilisateurs php 3](#)
- ◆ [Exemple avec array_walk](#)

- ♦ [Exemple avec arsort](#)
- ♦ [Exemple avec asort](#)
- ♦ [Exemple avec aspell_check](#)
- ♦ [Exemple avec aspell_check_raw](#)
- ♦ [Exemple avec aspell_new](#)
- ♦ [Exemple avec aspell_suggest](#)
- ♦ [Association d'objets](#)
- ♦ [Attaque d'un serveur de base de données \(mssql server\) – 2](#)
- ♦ [Attention aux erreurs de fractions](#)
- **B**
- - ♦ [Exemple avec base64_decode](#)
 - ♦ [base64_decode example](#)
 - ♦ [Exemple avec base64_encode](#)
 - ♦ [Exemple avec base_convert](#)
 - ♦ [Exemple avec basename](#)
 - ♦ [Exemple avec bcadd](#)
 - ♦ [Exemple avec bcpowmod](#)
 - ♦ [Exemple avec bcsb](#)
 - ♦ [Exemple avec bzcompress](#)
 - ♦ [Exemple avec bzdecompress](#)
 - ♦ [Exemple avec bzerror](#)
 - ♦ [Exemple avec bzopen](#)
 - ♦ [Exemple avec bzread](#)
- **C**
- - ♦ [Calcul du hash et enregistrement d'un utilisateur de mailing liste](#)
 - ♦ [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](#)
 - ♦ [Exemple avec ceil](#)
 - ♦ [Exemple avec chr](#)
 - ♦ [Exemple avec chunk_split](#)
 - ♦ [Classer un tableau multi-dimensionnel](#)
 - ♦ [code source et résultat dans phpinfo .](#)
 - ♦ [Collection](#)
 - ♦ [Collection](#)
 - ♦ [Colorisation d'url](#)
 - ♦ [Com event sink example](#)
 - ♦ [Commande utilisables avec mysql_info](#)
 - ♦ [Comment gérer les valeurs retournées par dbx_fetch_row](#)
 - ♦ [Comment retourner un flôt depuis une fonction](#)
 - ♦ [Comment utiliser le paramètre mode de chmod](#)
 - ♦ [Commentaires emboîtés](#)
 - ♦ [Exemple avec compact](#)
 - ♦ [Compactage d'une chaîne](#)
 - ♦ [Comparaison d'objets composés en php 4](#)
 - ♦ [Compilation cgi avec php \(unix\)](#)
 - ♦ [Compter le nombre de hits d'un utilisateur sur une page](#)
 - ♦ [Configuration ccl](#)
 - ♦ [Configuration de php.ini pour les utilisateurs de sjis](#)
 - ♦ [configuration de l'utilisateur d'exécution d'apache](#)
 - ♦ [configuration des gestionnaires de scheme pour xslt](#)
 - ♦ [Configuration interbase](#)

- ♦ [Configuration isapi de sambar](#)
- ♦ [Configuration par défaut des paramètres](#)
- ♦ [Configuration pour utiliser simultanément php 3 et 4](#)
- ♦ [Configuration sesam](#)
- ♦ [Configuration windws de magic.mime](#)
- ♦ [Configurer php avec apache 2.0 en module](#)
- ♦ [Connexion à un serveur informix](#)
- ♦ [connexion à un serveur ovrimos sql server et préparation d'une requête](#)
- ♦ [Connexion à une base sesam](#)
- ♦ [Connexion anonyme à un serveur ldap](#)
- ♦ [connexion au serveur ovrimos sql server et sélection d'une table système](#)
- ♦ [Constructeur de classe en php 4](#)
- ♦ [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](#)
- ♦ [Conversion html en texte](#)
- ♦ [Exemple avec copy](#)
- ♦ [Création d'un document pdf en mémoire](#)
- ♦ [Création d'un entête de document html](#)
- ♦ [Création d'un entête de document html avec dump-file.php">domdocument->dump_file](#)
- ♦ [Création d'un entête de document html avec dump-mem.php">domdocument->dump_mem](#)
- ♦ [Création d'un entête de document html avec html-dump-mem.php">domdocument->html_dump_mem](#)
- ♦ [création d'un fichier flash simple, basé sur une entrée de l'utilisateur, et sauvegarde dans une base.](#)
- ♦ [création d'un flux d'image gd, et affichage](#)
- ♦ [Création d'un nombre gmp](#)
- ♦ [Création d'un objet](#)
- ♦ [Création d'un tableau de tableaux](#)
- ♦ [Création d'une base dbase](#)
- ♦ [création d'une image gd et affichage de cette image](#)
- ♦ [Création de variables dans différents environnements](#)
- ♦ [Créer un attribut avec dom xml](#)
- ♦ [Créer un fichier gzip](#)
- ♦ [Créer un nouveau bloc de mémoire partagée shmop](#)
- ♦ [Creation of a long.](#)
- ♦ [Critères de tri](#)
- ♦ [Exemple avec crypt](#)
- ♦ [Création d'une image png avec php](#)
- ♦ [Exemple avec current](#)
- **D**
- ♦ [Déclaration interne de zend_function_entry.](#)
- ♦ [Déclaration interne de zend_module_entry.](#)
- ♦ [Décoder des entités html](#)
- ♦ [Déconnexion d'une base sesam](#)
- ♦ [Définir une constante](#)
- ♦ [Définition d'une constante](#)
- ♦ [Définition du type zval.](#)
- ♦ [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 de corruption de variables](#)
- ♦ [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](#)
- ♦ [Demarrer les règles de réécriture d'apache pour hyperwave](#)
- ♦ [Dernier jour du mois](#)
- ♦ [Dessin d'un cercle avec imagearc](#)
- ♦ [Destruction d'une session avec \\$_session](#)
- ♦ [Exemple avec dir](#)
- ♦ [Exemple avec dirname](#)
- ♦ [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](#)
- **E**
- ♦ [Exemple avec easter_days](#)
- ♦ [Ecrire des données dans un fichier compressé](#)
- ♦ [Ecrire un bloc de mémoire partagée](#)
- ♦ [Effacement d'un bloc de mémoire partagée](#)
- ♦ [Effacer une ressource existante](#)
- ♦ [Exemple avec empty](#)
- ♦ [Encryption d'une valeur avec triplades sous 2.4.x en mode ecb](#)
- ♦ [Exemple avec end](#)
- ♦ [Enregistrement d'un fichier téléchargé](#)
- ♦ [enregistrer une variable avec register_globals activé](#)
- ♦ [Enumerer tous les messages d'erreur ldap](#)
- ♦ [Envoi de mail complexe avec mail](#)
- ♦ [Equivalent de variables variables](#)
- ♦ [Exemple avec ereg](#)
- ♦ [Exemple avec ereg_replace](#)
- ♦ [Exemple avec eregi](#)
- ♦ [Exemple avec escapeshellarg](#)
- ♦ [Exemple avec escapeshellcmd](#)
- ♦ [exemple avec eval – concaténation de texte](#)
- ♦ [Evaluer un document fdf](#)
- ♦ [Exécute une requête postgresql asynchrone](#)
- ♦ [Exempe avec constant](#)
- ♦ [Exempe avec mb_convert_case](#)
- ♦ [Exemple](#)
- ♦ [Exemple bzwrite](#)
- ♦ [Exemple fbsql_num_rows](#)
- ♦ [Exemple ibase_connect](#)
- ♦ [Exemple imap_mime_header_decode](#)
- ♦ [Exemple imap_status](#)
- ♦ [Exemple msql_tablename](#)

- ◆ [Exemple mysql_data_seek](#)
- ◆ [Exemple mysql_db_name](#)
- ◆ [exemple mysql_num_rows](#)
- ◆ [Exemple mysql_stat](#)
- ◆ [Exemple pspell_add_to_personal](#)
- ◆ [Exemple restore_include_path](#)
- ◆ [Exemple setline.php">swfshape->setline](#)
- ◆ [Exemple setname.php">swfdisplayitem->setname](#)
- ◆ [Exemple str_shuffle](#)
- ◆ [Exemple étendu avec mysqli_fetch_assoc](#)
- ◆ [Exemple au format de date iso 8601:1988](#)
- ◆ [Exemple awt](#)
- ◆ [Exemple complet avec identification ldap](#)
- ◆ [Exemple complet de vérification de mot de passe avec ldap](#)
- ◆ [Exemple cracklib](#)
- ◆ [Exemple d'écriture dans un fichier avec fwrite](#)
- ◆ [exemple d'envoi de page compressée avec ob_gzhandler](#)
- ◆ [Exemple d'installation de php sous openbsd avec ports](#)
- ◆ [Exemple d'introduction](#)
- ◆ [Exemple d'ordre nis](#)
- ◆ [Exemple d'udf sqlite : taille maximale de chaîne](#)
- ◆ [Exemple d'utilisation d'une fonction php dans sqlite](#)
- ◆ [Exemple d'utilisation de max](#)
- ◆ [Exemple d'utilisation de miniature avec exif_thumbnail](#)
- ◆ [Exemple d'utilisation de zip](#)
- ◆ [Exemple de mysql_fetch_array avec mysql_both](#)
- ◆ [Exemple de swfsprite](#)
- ◆ [Exemple de bufferisation de sortie](#)
- ◆ [Exemple de chaîne heredoc](#)
- ◆ [Exemple de comparaison d'objets en php 5](#)
- ◆ [Exemple de configuration apache](#)
- ◆ [Exemple de configuration ccl](#)
- ◆ [Exemple de connexion à un serveur ldap ssl](#)
- ◆ [Exemple de contrôle de processus](#)
- ◆ [exemple de création de base mysql avec mysql_create_db](#)
- ◆ [Exemple de flôt qui affiche la page de 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 \(gracieusement offert par vic@zymsys.com \)](#)
- ◆ [exemple de gestion d'erreur lors de la création d'image \(gracieusement offert par vic@zymsys.com \)](#)
- ◆ [Exemple de gestion de sortie avec fonction utilisateur](#)
- ◆ [exemple de gestion des erreurs durant la création d'image \(gracieusement offert par vic@zymsys.com \)](#)
- ◆ [exemple de gestion des erreurs durant la création d'une image wbmp \(gracieusement proposé par vic@zymsys.com\)](#)
- ◆ [Exemple de l'extension zlib](#)
- ◆ [Exemple de lecture de ligne ovrmos](#)
- ◆ [Exemple de maître nis](#)
- ◆ [Exemple de message du débogueur](#)
- ◆ [Exemple de modification d'option odbc](#)
- ◆ [Exemple de modification de niveau d'erreur](#)
- ◆ [Exemple de nombres à virgule flottante](#)

- ♦ [Exemple de présentation de l'extension mysql](#)
- ♦ [Exemple de recherche nis](#)
- ♦ [Exemple de script avec mbstring](#)
- ♦ [Exemple de session : page2.php](#)
- ♦ [Exemple de transaction payflow pro](#)
- ♦ [Exemple de variable fonction](#)
- ♦ [Exemple ftp](#)
- ♦ [Exemple getusage](#)
- ♦ [Exemple mysql_close](#)
- ♦ [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 recode_file](#)
- ♦ [Exemple simple avec recode_string](#)
- ♦ [Exemple swf](#)
- ♦ [Exemples avec array_slice](#)
- ♦ [Exemples avec array_splice](#)
- ♦ [Exemples avec array_unshift](#)
- ♦ [Exemples avec easter_date](#)
- ♦ [Exemples avec error_log](#)
- ♦ [Exemples avec min](#)
- ♦ [Exemples avec next](#)
- ♦ [Exemples avec next](#)
- ♦ [Exemples avec pg_connect](#)
- ♦ [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 d'utilisation de str_word_count](#)
- ♦ [Exemples de chaînes](#)
- ♦ [Exemples de conversions](#)
- ♦ [Exemples de masques invalides](#)
- ♦ [Exemples de requêtes](#)
- ♦ [Exemples de vérification de type de variable](#)
- ♦ [Exemples liés à str_split](#)
- ♦ [Exemple avec exif_imagetype](#)
- ♦ [Exemple avec exif_read_data](#)
- ♦ [Exemple avec exit](#)
- ♦ [Exemple avec explode](#)
- ♦ [Expressions régulières](#)
- ♦ [Exemple avec extension_loaded](#)
- ♦ [Exemple avec extract](#)
- ♦ [Extrait du php.ini](#)

• 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](#)
- ♦ [Fermer une connexion informix](#)
- ♦ [Fermeture d'un bloc de mémoire partagée](#)
- ♦ [Exemple avec fgetc](#)
- ♦ [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 filemtime](#)
- ♦ [Filtre rot13 pour les flôts](#)
- ♦ [Exemple avec floatval](#)
- ♦ [Exemple avec floor](#)
- ♦ [Exemple avec fmod](#)
- ♦ [Fonction php qui scanne les titres sur un serveur yaz](#)
- ♦ [Fonctions à nombre d'arguments variable](#)
- ♦ [Fonctions calendrier](#)
- ♦ [Fonctions d'objets : test_script.php](#)
- ♦ [Fonctions dans une autre fonctions](#)
- ♦ [Exemple avec fopen sous windows](#)
- ♦ [Format des questionnaires avec xslt_set_sax_handler](#)
- ♦ [Format monétaire avec sprintf](#)
- ♦ [Formatage avec date](#)
- ♦ [Exemple avec ftp_chdir](#)
- ♦ [Exemple avec ftp_connect](#)
- ♦ [Exemple avec ftp_get_option](#)
- ♦ [Exemple avec ftp_login](#)
- ♦ [Exemple avec ftp_nlist](#)
- ♦ [Exemple avec ftp_put](#)
- ♦ [Exemple avec ftp_set_option](#)
- ♦ [Exemple avec func_get_arg](#)
- ♦ [Exemple avec func_get_args](#)
- ♦ [Exemple avec func_num_args](#)

- ♦ [Exemple avec fonction_exists](#)
- **G**
- ♦ [Générer un journal de suivi](#)
- ♦ [Genérer 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 messages non traités de sybase_set_message_handler](#)
- ♦ [Gestionnaire d'exception java](#)
- ♦ [Exemple avec get_browser](#)
- ♦ [Exemple avec get_class](#)
- ♦ [Exemple avec get_class_methods](#)
- ♦ [Exemple avec get_class_vars](#)
- ♦ [Exemple avec get_defined_constants](#)
- ♦ [Exemple avec get_defined_functions](#)
- ♦ [Exemple avec get_defined_vars](#)
- ♦ [Exemple avec get_extension_funcs](#)
- ♦ [Exemple avec get_include_path](#)
- ♦ [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 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 gmdate](#)
- ♦ [Exemple avec gmstrftime](#)
- ♦ [Exemple avec gzopen](#)
- ♦ [Exemple avec gzread](#)
- **H**
- ♦ [Héritage de classes \(2\)](#)
- ♦ [Exemple avec headers_sent](#)
- ♦ [Exemple avec hexdec](#)
- ♦ [Exemple avec htmlentities](#)
- ♦ [Exemple avec htmlspecialchars](#)
- **I**
- ♦ [Exemple avec ibase_blob_get](#)
- ♦ [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_set_encoding](#)

- ◆ [Identification http avec nom d'utilisateur/mot de passe forcé](#)
- ◆ [Exemple avec ifx_errormsg](#)
- ◆ [Exemple avec ifx_fetch_row](#)
- ◆ [Exemple avec ifx_fieldproperties](#)
- ◆ [Image_type_to_mime_type \(file\)](#)
- ◆ [Exemple avec imagecolorallocate](#)
- ◆ [Exemple avec imagecolordeallocate](#)
- ◆ [Exemple avec imagepng](#)
- ◆ [Exemple avec imagesetstyle](#)
- ◆ [Exemple avec imagesx](#)
- ◆ [Exemple avec imagesy](#)
- ◆ [Exemple avec imagettftext](#)
- ◆ [Exemple avec imagetypes](#)
- ◆ [Exemple avec imap_append](#)
- ◆ [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_rfc822_parse_adrlist](#)
- ◆ [Exemple avec imap_rfc822_write_address](#)
- ◆ [Exemple avec imap_set_quota](#)
- ◆ [Exemple avec imap_setflag_full](#)
- ◆ [implémentation de array_keys pour les utilisateurs de php 3](#)
- ◆ [Implémentation en php du nombre variable d'arguments avec fsockopen\(\)](#)
- ◆ [Exemple avec implode](#)
- ◆ [Exemple avec import_request_variables](#)
- ◆ [In an url](#)
- ◆ [Exemple avec in_array et deux tableaux en paramètres](#)
- ◆ [include et return](#)
- ◆ [include_once est insensible à la casse sous windows](#)
- ◆ [Inclure une image php en pdf](#)
- ◆ [Indexation automatique de tableau](#)
- ◆ [Exemple avec ingres_fetch_row](#)
- ◆ [Exemple avec ingres_query](#)
- ◆ [Exemple avec ini_get](#)
- ◆ [Exemple avec ini_get_all](#)
- ◆ [initialiser une session curl et récupération d'une page web](#)
- ◆ [Insertion à grande vitesse dans une table](#)
- ◆ [Insertion de valeurs dans la table "catalogue" ifx](#)
- ◆ [Installation de yaz](#)
- ◆ [Installation sur mac os x server](#)
- ◆ [Installer php comme module apache](#)
- ◆ [Instruction alternative if \(\) elseif et else](#)
- ◆ [Instruction break](#)
- ◆ [Instruction continue](#)
- ◆ [Instruction d'installation \(module partagé apache 2\)](#)
- ◆ [Instruction declare](#)
- ◆ [Instruction elseif \(\)](#)

- ♦ [Instruction for\(\) alternative](#)
- ♦ [Instruction foreach\(\) et while\(\) \(3\)](#)
- ♦ [Instruction if \(\) else](#)
- ♦ [Instruction if \(\) et bloc](#)
- ♦ [Instruction switch\(\) alternative](#)
- ♦ [Instruction while \(\) et break](#)
- ♦ [Instructions d'installation avec apache \(version module\)](#)
- ♦ [Instructions d'installation caudium](#)
- ♦ [Introduction à la mémoire partagée](#)
- ♦ [Inverser une chaîne avec strev](#)
- ♦ [Exemple avec inversion de la table de traduction des caractères en entités html](#)
- ♦ [Exemple avec ircg_fetch_error_msg](#)
- ♦ [Exemple avec is_bool](#)
- ♦ [Exemple avec is_callable](#)
- ♦ [Exemple avec is_scalar](#)
- ♦ [Exemple avec isset et un tableau](#)
- J
- ♦ [Exemple avec jdtojewish](#)
- ♦ [Jeu de caractères qui fonctionnent avec php](#)
- K
- ♦ [Exemple avec krsort](#)
- ♦ [Exemple avec ksort](#)
- L
- ♦ [Exemple avec la fonction stream_get_filters](#)
- ♦ [La valeur null](#)
- ♦ [Exemple avec le domaine par défaut](#)
- ♦ [Le fichier config.m4 par défaut](#)
- ♦ [Exemple avec le paramètre convmap](#)
- ♦ [Le séparateur d'instructions](#)
- ♦ [Lecture d'un fichier ligne par ligne](#)
- ♦ [Lecture d'un objet postgres](#)
- ♦ [Les arguments à valeur par défaut doivent être en premier : valide](#)
- ♦ [Exemple avec les sockets : client tcp/ip simple](#)
- ♦ [Les variables sont locales à la fonction \(2\)](#)
- ♦ [Les variables statiques et la récursivité](#)
- ♦ [Les variables statiques et les références \(2\)](#)
- ♦ [Les variables super globales](#)
- ♦ [Lire des informations dbm depuis la ligne de commande](#)
- ♦ [Lire des informations sur les champs postgresql](#)
- ♦ [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 méta-données d'une table postgresql](#)
- ♦ [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 les noeuds sous dom xml](#)
- ♦ [liste tous les fichiers du dossier courant, sauf "." et ".."](#)
- ♦ [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
- - ♦ Méthode avancée
 - ♦ Manipuler des éléments de tableau
 - ♦ Manipuler des éléments de tableau
 - ♦ 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_output_handler
 - ♦ Exemple avec mb_preferred_mime_string
 - ♦ Exemple avec mb_strimwidth
 - ♦ Exemple avec mb_strtolower
 - ♦ Exemple avec mb_strtoupper
 - ♦ Exemple avec mb_substitute_character
 - ♦ Exemple avec mb_substr_count
 - ♦ Exemple avec mdecrypt_create_iv
 - ♦ Exemple avec mdecrypt_enc_get_algorithms_name
 - ♦ Exemple avec mdecrypt_enc_get_modes_name
 - ♦ Exemple avec mdecrypt_enc_get_supported_key_sizes
 - ♦ Exemple avec mdecrypt_encrypt
 - ♦ Exemple avec mdecrypt_get_block_size
 - ♦ Exemple avec mdecrypt_get_block_size
 - ♦ Exemple avec mdecrypt_get_block_size
 - ♦ Exemple avec mdecrypt_get_cipher_name
 - ♦ Exemple avec mdecrypt_list_algorithms
 - ♦ Exemple avec mdecrypt_list_modes
 - ♦ Exemple avec md5
 - ♦ Exemple avec mdecrypt_generic
 - ♦ Exemple avec memory_get_usage
 - ♦ Exemple avec mhash_get_hash_name
 - ♦ Exemple avec microtime
 - ♦ Migration depuis 2.0: concaténation de chaînes
 - ♦ Migration depuis 2.0: valeur retournées, nouvelle façon
 - ♦ Migration des variables globales
 - ♦ Migration: nouvelle syntaxe if..endif
 - ♦ Migration: nouvelle syntaxe while..endwhile
 - ♦ Mise en italique d'un mot dans un texte
 - ♦ Exemple avec mkdir
 - ♦ Modification d'une variable d'environnement
 - ♦ Modifier le taux de baud du port série
 - ♦ Modifier les paramètres de configuration sesam
 - ♦ Exemple avec money_format
 - ♦ Exemple avec mysql_query

- ♦ [Exemple avec mssql_next_result](#)
- ♦ [Exemple avec mssql_select_db](#)
- ♦ [Exemple avec mt_srand](#)
- ♦ [Exemple avec multicolor.php">swfdisplayitem->multicolor](#)
- ♦ [Exemple avec mysql_client_encoding](#)
- ♦ [Exemple avec mysql_connect](#)
- ♦ [Exemple avec mysql_errno](#)
- ♦ [Exemple avec mysql_error](#)
- ♦ [Exemple avec mysql_escape_string](#)
- ♦ [Exemple avec mysql_fetch_assoc](#)
- ♦ [Exemple avec mysql_fetch_field](#)
- ♦ [Exemple avec mysql_fetch_object](#)
- ♦ [Exemple avec mysql_field_name](#)
- ♦ [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_fields](#)
- ♦ [Exemple avec mysql_list_processes](#)
- ♦ [Exemple avec mysql_list_tables](#)
- ♦ [Exemple avec mysql_query \(2\)](#)
- ♦ [Exemple avec mysql_real_escape_string](#)
- ♦ [Exemple avec mysql_result](#)
- ♦ [Exemple avec mysql_tablename](#)
- ♦ [Exemple avec mysql_thread_id](#)
- ♦ [mysqli_fetch_array avec mysqli_both](#)
- N
- - ♦ [Exemple avec natsort](#)
 - ♦ [Exemple avec ncurses_getmouse](#)
 - ♦ [Exemple avec ncurses_mousemask](#)
 - ♦ [Nom de champs et type sql.](#)
 - ♦ [Nom du catalogue de messages sesam](#)
 - ♦ [Nombre de lignes affectées](#)
 - ♦ [Nombre variable d'argument sous forme de tableau](#)
 - ♦ [Notre premier script php : bonjour.php](#)
 - ♦ [Nouveau comportement de strtok](#)
 - ♦ [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_iconv_handler](#)
 - ♦ [Exemple avec ocibindbyname](#)
 - ♦ [Exemple avec ocicolumnname](#)
 - ♦ [Exemple avec ocicolumnsize](#)
 - ♦ [Exemple avec ocicolumntype](#)
 - ♦ [Ocicommit](#)
 - ♦ [Exemple avec ocidefinebyname](#)
 - ♦ [Exemple avec ocifetchinto](#)
 - ♦ [Exemple avec ocifetchstatement](#)

- ♦ [Exemple avec ocilogon](#)
- ♦ [Exemple avec ocinewdescriptor](#)
- ♦ [Exemple avec ocinlogon](#)
- ♦ [Exemple avec ocinumcols](#)
- ♦ [Exemple avec ocirowcount](#)
- ♦ [Exemple avec ociserverversion](#)
- ♦ [Exemple avec ocistatementtype](#)
- ♦ [Exemple avec odbc_fetch_into en php 4.2.0](#)
- ♦ [Exemple avec odbc_result](#)
- ♦ [Opérations arithmétiques et caractères](#)
- ♦ [Opérations arithmétiques sur un caractère](#)
- ♦ [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](#)
- ♦ [Exemple avec ora_bind](#)
- ♦ [Exemple avec ora_fetch_into](#)
- ♦ [Exemple avec ora_logon](#)
- ♦ [Exemple avec ord](#)
- ♦ [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 ovrmos_connect](#)
- ♦ [Exemple avec ovrmos_result_all](#)
- **P**
- ♦ [Parcourir la liste des hash](#)
- ♦ [Exemple avec parse_ini_file](#)
- ♦ [Exemple avec parse_str](#)
- ♦ [Exemple avec parse_url](#)
- ♦ [parsemolddb.php : analyse molddb.xml et crée un tableau d'objets moléculaires](#)
- ♦ [Passage de valeur par référence, ponctuellement](#)
- ♦ [Passer des données fdf à un second formulaire](#)
- ♦ [Passer du mode php au mode html et vice-versa](#)
- ♦ [passer en revue \\$_post 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_findfont](#)
- ♦ [Exemple avec pdf_setcolor : couleurs cmyk](#)
- ♦ [Personnalisation de la page de crédits php](#)
- ♦ [Petit exemple avec bzip2](#)
- ♦ [Exemple avec pfprow_process_raw](#)
- ♦ [Exemple avec pg_affected_rows](#)

- ♦ [Exemple avec pg_close](#)
- ♦ [Exemple avec pg_connection_busy](#)
- ♦ [Exemple avec pg_connection_reset](#)
- ♦ [Exemple avec pg_connection_status](#)
- ♦ [Exemple avec pg_dbname](#)
- ♦ [Exemple avec pg_delete](#)
- ♦ [Exemple avec pg_fetch_all](#)
- ♦ [Exemple avec pg_fetch_array](#)
- ♦ [Exemple avec pg_fetch_assoc](#)
- ♦ [Exemple avec pg_field_is_null](#)
- ♦ [Exemple avec pg_get_notify](#)
- ♦ [Exemple avec pg_get_pid](#)
- ♦ [Exemple avec pg_insert](#)
- ♦ [Exemple avec pg_ping](#)
- ♦ [Exemple avec pg_select](#)
- ♦ [Exemple avec pg_update](#)
- ♦ [Exemple avec php_logo_guid](#)
- ♦ [Exemple avec php_sapi_name](#)
- ♦ [Exemple avec phpinfo](#)
- ♦ [Exemple avec phpversion](#)
- ♦ [Exemple avec pi](#)
- ♦ [Exemple avec posix_getgid](#)
- ♦ [Postgres retourne une ligne](#)
- ♦ [Pré remplir un formulaire pdf](#)
- ♦ [Présentation de dbm](#)
- ♦ [Exemple avec preg_grep](#)
- ♦ [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_elipse](#)
- ♦ [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](#)
- ♦ [Exemple avec proc_open](#)
- ♦ [Produit une liste de tous les services d'une société avec ldap_list](#)
- ♦ [Profiler une section de code php](#)
- ♦ [Exemple avec pspell_add_to_personal](#)
- ♦ [Exemple avec pspell_add_to_personal](#)
- ♦ [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 avec pow](#)
- R
- ♦ [Récupération du noeud racine avec document-element.php">domdocument->document_element](#)
- ♦ [Références et constructeurs](#)
- ♦ [Résolution d'une équation diophantine linéaire](#)
- ♦ [Résultat de get_meta_tags](#)
- ♦ [résultat de set_error_handler et trigger_error](#)
- ♦ [Exemple avec range](#)
- ♦ [Exemple avec rawurlencode](#)
- ♦ [Exemple avec readline](#)
- ♦ [Exemple avec realpath](#)
- ♦ [Recherche avec ldap](#)
- ♦ [Recherche ldap](#)
- ♦ [Recherche les couples de balises html \(gourmand\)](#)
- ♦ [Recherche parallèle utilisant yaz](#)
- ♦ [Exemple avec register_tick_function](#)
- ♦ [Reinstaller apache avec pthread](#)
- ♦ [Exemple avec rename](#)
- ♦ [reprise du téléchargement à la position 100 avec ftp_nb_get](#)
- ♦ [Requête de mise à jour](#)
- ♦ [Requête sql update](#)
- ♦ [require_once est insensible à la casse sur windows](#)
- ♦ [Resuming an upload with ftp_nb_put](#)
- ♦ [Retourner une référence d'une fonction](#)
- ♦ [Exemple avec rotateto.php">swfdisplayitem->rotateto](#)
- ♦ [Exemple avec round](#)
- ♦ [Exemple avec rsort](#)
- ♦ [Exemple avec rtrim](#)
- S
- ♦ [Sauver le résultat d'une transformation xslt dans un fichier](#)
- ♦ [Sauver le résultat d'une transformation xslt dans une chaîne](#)
- ♦ [Sauver un fichier par méthode put en php 4](#)
- ♦ [Sauver/restaurer l'environnement cpdf](#)
- ♦ [Scinde une chaîne et capture les offsets](#)
- ♦ [Exemple avec serialize](#)
- ♦ [Exemple avec sesam_affected_rows](#)
- ♦ [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](#)
- ♦ [Set server controls](#)
- ♦ [Exemple avec set_include_path](#)
- ♦ [Exemple avec setcookie](#)
- ♦ [Exemple avec setlocale pour windows](#)
- ♦ [Exemple avec settype](#)
- ♦ [Exemple avec sha1](#)
- ♦ [Exemple avec shuffle](#)
- ♦ [Exemple avec sin](#)
- ♦ [Exemple avec snmp_get_quick_print](#)
- ♦ [Exemple avec snmp_set_quick_print](#)
- ♦ [Exemple avec snmpget](#)
- ♦ [Exemple avec socket_sendto](#)
- ♦ [Exemple avec socket_strerror](#)
- ♦ [Exemple avec sort](#)
- ♦ [spécifier une catégorie avec un format hiérarchisé](#)
- ♦ [Exemple avec split](#)
- ♦ [sprintf : format de monétaire](#)
- ♦ [Exemple avec sql_regcase](#)
- ♦ [sqlite_array_query traditionnel](#)
- ♦ [Exemple avec sqlite_fetch_single](#)
- ♦ [Exemple avec sqlite_open](#)
- ♦ [Exemple avec sqrt](#)
- ♦ [Exemple avec srand](#)
- ♦ [Exemple avec sscanf : utilisation des paramètres optionnels](#)
- ♦ [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 strcasecmp](#)
- ♦ [Exemple avec stream_context_create](#)
- ♦ [Exemple avec stream_copy_to_stream](#)
- ♦ [Exemple avec stream_get_transports](#)
- ♦ [Exemple avec stream_set_timeout](#)
- ♦ [Exemple avec stream_set_write_buffer](#)
- ♦ [Exemple avec strip_tags](#)
- ♦ [Exemple avec stripslashes](#)
- ♦ [Exemple avec stristr](#)
- ♦ [Exemple avec strlen](#)
- ♦ [Exemple avec strnatcmp](#)
- ♦ [Exemple avec strpos](#)
- ♦ [Exemple avec strchr](#)
- ♦ [Exemple avec strrpos](#)
- ♦ [Exemple avec strspn](#)
- ♦ [Exemple avec strstr](#)
- ♦ [Exemple avec strtolower](#)
- ♦ [Exemple avec strtoupper](#)
- ♦ [Exemple avec strtr avec deux arguments](#)
- ♦ [Structure d'instruction exécutable](#)
- ♦ [Structure d'un entier](#)

- ♦ [Structure du tableau retourné par ldap_get_entries](#)
- ♦ [Exemple avec substr_count](#)
- ♦ [Exemple avec substr_replace](#)
- ♦ [Suppression d'un attribut](#)
- ♦ [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 swfmovie->streammp3](#)
- ♦ [Exemple avec swfshape](#)
- ♦ [Exemple avec swftext](#)
- ♦ [Exemple avec sybase_affected_rows](#)
- ♦ [Exemple avec sybase_connect](#)
- ♦ [sybase_fetch_object](#) retourne l'objet foo
- ♦ [Exemple avec sybase_unbuffered_query](#)
- ♦ [Syntaxe complexe de chaîne](#)
- T
 - ♦ [Télécharger plusieurs fichiers simultanément](#)
 - ♦ [Tableau d'index commençant à 1](#)
 - ♦ [Tableaux multi-dimensionnels, et récursifs](#)
 - ♦ [Exemple avec tempnam](#)
 - ♦ [Test d'existence d'un fichier](#)
 - ♦ [Test des paramètres passés par référence.](#)
 - ♦ [Exemple avec tmpfile](#)
 - ♦ [Exemple avec token_get_all](#)
 - ♦ [Exemple avec token_name](#)
 - ♦ [Exemple avec touch](#)
 - ♦ [Transtypage](#)
 - ♦ [Transtypage en chaîne](#)
 - ♦ [Transtypage xml -> html](#)
 - ♦ [tri avec usort sur un objet](#)
 - ♦ [Exemple avec trigger_error](#)
 - ♦ [Exemple avec trim](#)
 - ♦ [Trouver le document d'un noeud avec dom xml](#)
 - ♦ [Types mysql](#)
- U
 - ♦ [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](#)
 - ♦ [Un exemple simple avec stripslashes](#)
 - ♦ [Un exemple simple de liste retournée par php_ini_scanned_files](#)
 - ♦ [Un fichier test pour first_module.so.](#)
 - ♦ [un moyen pratique pour remplacer opendir par glob](#)
 - ♦ [Une classe de flôt pour accéder aux variables globales](#)
 - ♦ [Une extension simple](#)
 - ♦ [Une navigation de fiches plus sécuritaire](#)
 - ♦ [Exemple avec uniqid](#)
 - ♦ [Exemple avec unpack](#)

- ♦ [Exemple avec unserialize](#)
- ♦ [Exemple avec unset avec une variable globale](#)
- ♦ [Exemple avec urldecode](#)
- ♦ [exemple avec urlencode et htmlentities](#)
- ♦ [Using http_build_query with an object](#)
- ♦ [Using mcrypt_module_open in encryption](#)
- ♦ [Utilisateurs.txt](#)
- ♦ [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 ftp_raw pour se logger manuellement](#)
- ♦ [Utilisation de gd_info](#)
- ♦ [utilisation de header pour générer un fichier de type pdf ou d'un autre type](#)
- ♦ [Utilisation de mysqli_change_user](#)
- ♦ [Utilisation de mysqli_character_set_name](#)
- ♦ [Utilisation de mysqli_data_seek](#)
- ♦ [Utilisation de mysqli_execute](#)
- ♦ [Utilisation de syslog](#)
- ♦ [Utilisation de xslt_process pour transformer une variable xml avec une autre variable xsl et placer le résultat dans une variable](#)
- ♦ [Utilisation de connexions udp](#)
- ♦ [Utilisation de curl et php pour récupérer une page](#)
- ♦ [Utilisation de fonctions anonymes comme fonction de callback](#)
- ♦ [Utilisation de l'enregistrement de messages xslt](#)
- ♦ [Utilisation de la fonction mysqli_autocommit](#)
- ♦ [Utilisation de la fonction mysqli_connect](#)
- ♦ [Utilisation de la fonction mysqli_error](#)
- ♦ [Utilisation de paquets incrémentaux](#)
- ♦ [Utilisation de procédures stockées](#)
- ♦ [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 objets de grande taille \(large objects\)](#)
- ♦ [Utilisation des tableaux avec setcookie](#)
- ♦ [Utilisation du paramètre return de print_r](#)
- ♦ [Utiliser dbase_numfields](#)
- ♦ [Utiliser le type html pour les extensions php](#)
- ♦ [Utiliser un mot de passe et md5](#)
- ♦ [Utiliser un ref cursor issue d'une commande select](#)
- ♦ [Utiliser un serveur de socket udp](#)
- ♦ [Utiliser une valeur négative de length](#)
- V
- - ♦ [Vérification gettext](#)
 - ♦ [Vérification de la version du protocole avec ldap_get_option](#)
 - ♦ [Vérification de noms de fichiers sécurisée](#)
 - ♦ [Vérification des erreurs avec strtotime](#)
 - ♦ [vérifier la présence de constantes avec defined](#)
 - ♦ [Vérifier le nom d'une couleur avec un masque shell](#)
 - ♦ [Validation de téléchargement de fichiers](#)
 - ♦ [Valider une transaction sesam](#)
 - ♦ [Exemple avec var_dump](#)
 - ♦ [Exemple avec var_export](#)
 - ♦ [Variables de formulaires complexes](#)

- ♦ Exemple avec version_compare
- W
- ♦ wddx_serialize_vars
- ♦ Exemple avec while ()
- ♦ Windows include_path
- ♦ Exemple avec wordwrap avec césure
- X
- ♦ Xmltest2.xml
- Y
- ♦ Exemple avec yp_first
- ♦ Exemple avec yp_next
- Z
- ♦ Exemple avec zend_version