Manuel PHP 4.2 – Version Française

http://www.nexen.net

Sommaire

Informations sur l'ouvrage 1
1.1 Les auteurs 1
1.2 L'éditeur 1
1.3 Note légale 1

2 Préface 2
2.1 A propos de ce manuel 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
3.1.3 La genèse du PHP 4
3.2 Une introduction à PHP 5
3.2.1 Le nécessaire 5
3.2.2 Votre première page PHP 5
3.2.3 Trucs pratiques 6
3.2.4 Utiliser un formulaire 8
3.2.5 Utiliser des codes anciens avec les nouvelles versions de PHP 8
3.2.6 Et après? 9
3.3 Installation 9
3.3.1 Considérations générales sur l'installation 9
3.3.2 Installation sous UNIX 10
3.3.3 Installation sous Linux/Unix 11
3.3.4 Installation sous HP−UX 11
3.3.5 Installation sous Solaris 13
3.3.6 Installations Unix/OpenBSD 13
3.3.7 Unix/Mac OS X installs 14
3.3.8 Liste complète des options de configuration 17
3.3.9 Installation sous Windows 26
3.3.10 Installation avec le serveur Apache 33
3.3.11 Installation avec les serveurs Apache 2.0 36
3.3.12 CGI/ Installation pour exécution en ligne de commande 39
3.3.13 Installation avec les serveurs fhttpd 40
3.3.14 Installation sur le serveur Caudium 40
3.3.15 Installation avec les serveurs IIS/PWS 41
3.3.16 Installation sous Netscape et iPlanet Enterprise Serveur 43
3.3.17 Installation pour les serveurs OmniHTTPd 47
3.3.18 Installation Oreilly Website Pro Server 48
3.3.19 Installation Xitami 48
3.3.20 Autres serveurs web 48
3.3.21 Des problèmes? 48
3.4 Configuration 49
3.4.1 Le fichier de configuration 49
3.5 Sécurité 59
3.5.1 Considérations générales 59
3.5.2 Binaires CGI 60
3.5.3 Module Apache 62
3.5.4 Sécurité des fichiers 63
3.5.5 Rapport d'erreur 64
3.5.6 Utilisation des variables HTTP 66
3.5.7 Données transmises par les internautes 67

Sommaire

3 Comment commencer
3.5.8 Masquer PHP....67
3.5.9 Etre à jour....68

4 Référence....69
4.1 La syntaxe de base....69
4.1.1 Le passage du HTML au PHP....69
4.1.2 Le séparateur d'instruction....70
4.1.3 Commentaires....70
4.2 Les types....71
4.2.1 Introduction....71
4.2.2 Booléens....71
4.2.3 Entiers....72
4.2.4 Les nombres à virgule flottante....74
4.2.5 Les chaînes de caractères....75
4.2.6 Les tableaux....81
4.2.7 Les objets....86
4.2.8 Ressources....87
4.2.9 La valeur NULL....87
4.2.10 Définition du type....87
4.3 Les variables....90
4.3.1 Essentiel....90
4.3.2 Variables prédéfinies....91
4.3.3 Missing title....93
4.3.4 Portée des variables....96
4.3.5 Les variables dynamiques....98
4.3.6 Variables externes à PHP....99
4.4 Les constantes....101
4.4.1 Syntaxe....101
4.4.2 Constantes prédéfinies....102
4.5 Les expressions....104
4.6 Les opérateurs....106
4.6.1 La précédence des opérateurs....106
4.6.2 Les opérateurs arithmétiques....107
4.6.3 Les opérateurs d'assignation....107
4.6.4 Opérateurs sur les bits....108
4.6.5 Opérateurs de comparaison....108
4.6.6 Opérateur de contrôle d'erreur....109
4.6.7 Opérateur d'exécutions....110
4.6.8 Opérateurs d'incrémentation/Décrémentation....110
4.6.9 Les opérateurs logiques....111
4.6.10 Opérateurs de chaînes....111
4.7 Les structures de contrôle....111
4.7.1 if....112
4.7.2 else....112
4.7.3 elseif....113
4.7.4 Syntaxe alternative....113
4.7.5 while....114
4.7.6 do..while....115
4.7.7 for....116
4.7.8 foreach....117
4.7.9 break....119

Sommaire

4 Référence
4.7.10 continue....120
4.7.11 switch....120
4.7.12 declare....122
4.7.13 require....124
4.7.14 include....125
4.7.15 require_once....126
4.7.16 include_once....128
4.8 Les fonctions....129
4.8.1 Les fonctions utilisateur....129
4.8.2 Les arguments de fonction....129
4.8.3 Les valeurs de retour....131
4.8.4 old_function....132
4.8.5 Fonctions−variable....132
4.9 Les classes et les objets....133
4.9.1 L class....133
4.9.2 extends....135
4.9.3 Constructor....136
4.9.4 O ::....138
4.9.5 parent....139
4.9.6 Sauvegarde d'objets − cas des sessions....139
4.9.7 L __sleep __wakeup....141
4.9.8 Références dans un constructeur....141
4.10 Les références....143
4.10.1 Qu'est ce qu'une référence?....143
4.10.2 Que font les références ?....143
4.10.3 Ce que les références ne sont pas....144
4.10.4 Passage par référence....145
4.10.5 Retourner des références....146
4.10.6 Détruire une référence....146
4.10.7 Repérer une référence....146

5 Caractéristiques....148
5.1 Authentification HTTP avec PHP....148
5.2 Cookies....149
5.3 Gestion des chargements de fichier....149
5.3.1 Chargements de fichiers par méthode POST....149
5.3.2 Erreurs classiques....151
5.3.3 Chargement multiples de fichiers....151
5.3.4 Chargement par méthode PUT....152
5.4 Utilisation des fichiers à distance....153
5.5 Gestion des connexions....154
5.6 Connexions persistantes aux bases de données....155
5.7 Safe mode....156
5.7.1 Functions désactivées par le Safe Mode....157
5.8 Utiliser PHP en ligne de commande....160

6 Index des fonctions....169
6.1 Apache....169
6.1.1 Introduction....169
6.1.2 Installation....169
6.1.3 Configuration à l'exécution....169

Sommaire

6 Index des fonctions

6.1.4 Types de ressources....169
6.1.5 Constantes Prédefinies....170
6.1.7 apache_lookup_uri() : Effectue une requête partielle pour l'URI spécifiée et renvoie toutes les informations....170
6.1.8 apache_note() : Affiche ou affecte le paramètre "apache request notes"....171
6.1.9 apache_request_headers() : Lit tous les entêtes HTTP de la requête....171
6.1.10 apache_response_headers() : Lit tous les entêtes de réponse HTTP....171
6.1.11 apache_setenv() : Modifie une variable de subprocess_env Apache....172
6.1.12 ascii2ebcdic() : Transforme une chaîne ASCII en EBCDIC....172
6.1.13 ebcdic2ascii() : Transforme une chaîne EBCDIC en ASCII....172
6.1.14 getallheaders() : Récupère toutes les en−têtes des requêtes HTTP....172
6.1.15 virtual() : Effectue une sous−requête Apache....173
6.2 Tableaux....174
6.2.1 Introduction....174
6.2.2 Pré−requis....174
6.2.3 Installation....174
6.2.4 Configuration à l'exécution....174
6.2.5 Types de ressources....174
6.2.6 Constantes Prédefinies....174
6.2.7 Voir aussi....175
6.2.9 array_chunk() : Sépare un tableau en tableaux de taille inférieure....176
6.2.10 array_count_values() : Compte le nombre de valeurs dans un tableau....177
6.2.11 array_diff() : Calcule la différence entre deux tableaux....177
6.2.12 array_fill() : Remplis un tableau avec une même valeur....178
6.2.13 array_filter() : Filtre les éléments d'un tableau....178
6.2.14 array_flip() : Remplace les clés par les valeurs, et les valeurs par les clés....179
6.2.15 array_intersect() : Calcule l'intersection de tableaux....180
6.2.16 array_key_exists() : Vérifie si une clé éxiste dans un tableau....180
6.2.17 array_keys() : Retourne toutes les clés d'un tableau....181
6.2.18 array_map() : Applique sur fonction sur des tableaux....182
6.2.19 array_merge_recursive() : Combine plusieurs tableaux ensembles, récursivement....184
6.2.20 array_merge() : Rassemble plusieurs tableaux....185
6.2.21 array_multisort() : Tri multi−dimensionnel....185
6.2.22 array_pad() : Complète un tableau jusqu'à la longueur spécifiée, avec une valeur....186
6.2.23 array_pop() : Dépile un élément de la fin d'un tableau....187
6.2.24 array_push() : Empile un ou plusieurs éléments à la fin d'un tableau....187
6.2.25 array_rand() : Prend une ou plusieurs valeurs, au hasard dans un tableau....188
6.2.26 array_reduce() : Réduit itérativement un tableau....188
6.2.27 array_reverse() : Renverse l'ordre des éléments d'un tableau....189
6.2.28 array_search() : Recherche dans un tableau la clé associée à une valeur....189
6.2.29 array_shift() : Dépile un élément au début d'un tableau....190
6.2.30 array_slice() : Extrait une portion de tableau....190
6.2.31 array_splice() : Efface et remplace une portion de tableau....191
6.2.32 array_sum() : Calcule la somme des valeurs du tableau....192
6.2.33 array_unique() : Dédoublonne un tableau....192
6.2.34 array_unshift() : Empile un ou plusieurs éléments au début d'un tableau....193
6.2.35 array_values() : Retourne les valeurs d'un tableau....194
6.2.36 array_walk() : Exécute une fonction sur chacun des membres d'un tableau....194
6.2.37 array() : Crée un tableau....195

Sommaire

6 Index des fonctions

6.2.38 arsort() : Trie un tableau en ordre inverse....196
6.2.39 asort() : Trie un tableau en ordre....197
6.2.40 compact() : Crée un tableau contenant les variables et leur valeur....198
6.2.41 count() : Compte le nombre d'éléments d'un tableau....198
6.2.42 current() : Transforme une variable en tableau....199
6.2.43 each() : Retourne chaque paire clé/valeur d'un tableau....199
6.2.44 end() : Positionne le pointeur de tableau en fin de tableau....200
6.2.45 extract() : Importe les variables dans la table des symboles....200
6.2.46 in_array() : Indique si une valeur appartient à un tableau....201
6.2.47 key() : Retourne une clé d'un tableau associatif....202
6.2.48 krsort() : Trie un tableau en sens inverse et suivant les clés....202
6.2.49 ksort() : Trie un tableau suivant les clés....203
6.2.50 list() : Transforme une liste de variables en tableau....203
6.2.51 natcasesort() : Tri d'un tableau avec l'algorithme à "ordre naturel" insensible à la casse....204
6.2.52 natsort() : Tri d'un tableau avec l'algorithme à "ordre naturel"....204
6.2.53 next() : Avance le pointeur interne d'un tableau....205
6.2.54 pos() : Retourne l'élément courant d'un tableau....205
6.2.55 prev() : Recule le pointeur courant de tableau....206
6.2.56 range() : Crée un tableau contenant un intervalle d'éléments....206
6.2.57 reset() : Remet le pointeur interne de tableau au début....206
6.2.58 rsort() : Trie en ordre inverse....206
6.2.59 shuffle() : Mélange les éléments d'un tableau....207
6.2.60 sizeof() : Retourne le nombre d'élément d'un tableau....207
6.2.61 sort() : Trie le tableau....207
6.2.62 uasort() : Trie d'un tableau en utilisant une fonction de comparaison définie par l'utilisateur....208
6.2.63 uksort() : Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur....208
6.2.64 usort() : Trie un tableau en utilisant une fonction de comparaison définie par l'utilisateur....209
6.3 Aspell....210
6.3.2 aspell_check() : Vérifie un mot [Obsolète]....210
6.3.3 aspell_new() : Charge un nouveau dictionnaire [Obsolète]....210
6.3.4 aspell_suggest() : Suggère l'orthographe d'un mot [Obsolète]....211
6.4 Nombres de grande taille....212
6.4.1 Introduction....212
6.4.2 Pré−requis....212
6.4.3 Installation....212
6.4.4 Configuration à l'exécution....212
6.4.5 Types de ressources....212
6.4.6 Constantes Prédefinies....212
6.4.8 bccomp() : Compare deux nombres de grande taille....213
6.4.9 bcdiv() : Divise deux nombres de grande taille....213
6.4.10 bcmod() : Retourne le reste d'une division entre nombre de grande taille....213
6.4.11 bcmul() : Multiplie deux nombres de grande taille....213
6.4.12 bcpow() : Elève un nombre à la puissance n−ième....214
6.4.13 bcscale() : Détermine le nombre de décimales par défaut....214
6.4.14 bcsqrt() : Renvoie la racine carrée d'un nombre de grande taille....214
6.4.15 bcsub() : Soustrait un nombre de grande taille à un autre....214
6.5 Compression Bzip2....215

Sommaire

6 Index des fonctions

6.5.1 Exemple de compression bzip2 215
6.5.3 bzcompress() : Compresse une chaîne avec bzip2 215
6.5.4 bzdecompress() : Décompresse une chaîne bzip2 216
6.5.5 bzerrno() : Retourne le numéro d'erreur bzip2 216
6.5.6 bzerror() : Retourne le numéro et le message d'erreur bzip2 dans un tableau 216
6.5.7 bzerrstr() : Retourne le message d'erreur bzip2 217
6.5.8 bzflush() : Force l'écriture de toutes les données compressées 217
6.5.9 bzopen() : Ouvre un fichier compressé avec bzip2 217
6.5.10 bzread() : Lecture binaire d'un fichier bzip2 218
6.5.11 bzwrite() : Ecriture binaire dans un fichier bzip2 218
6.6 Calendrier 219
6.6.1 Introduction 219
6.6.2 Installation 219
6.6.3 Configuration à l'exécution 219
6.6.4 Types de ressources 219
6.6.5 Constantes Prédefinies 219
6.6.7 cal_from_jd() : Convertit le nombre de jours Julien en un calendrier et retourne des détails 220
6.6.8 cal_info() : Retourne des détails sur un calendrier 220
6.6.9 cal_to_jd() : Convertit un calendrier en nombre de jours Julien 220
6.6.10 easter_date() : Retourne un timestamp UNIX pour Pâques, à minuit 221
6.6.11 easter_days() : Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée 221
6.6.12 FrenchToJD() : Convertit une date du calendrier français républicain en nombre de jours du calendrier Julien 222
6.6.13 GregorianToJD() : Convertit une date grégorienne en nombre de jours du calendrier Julien 222
6.6.14 JDDayOfWeek() : Retourne le numéro du jour de la semaine 223
6.6.15 JDMonthName() : Retourne le nom du mois 223
6.6.16 JDToFrench() : Convertit le nombre de jours du calendrier Julien en date du calendrier français républicain 223
6.6.17 JDToGregorian() : Convertit le nombre de jours du calendrier Julien en date grégorienne 223
6.6.18 JDToJewish() : Convertit le nombre de jours du calendrier Julien en date du calendrier juif 224
6.6.19 JDToJulian() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien 224
6.6.20 jdtounix() : Convertit un nombre de jours Julien en timestamp UNIX 224
6.6.21 JewishToJD() : Convertit une date du calendrier juif en nombre de jours du calendrier Julien 224
6.6.22 JulianToJD() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien 225
6.6.23 unixtojd() : Convertit un timestamp UNIX en nombre de jours Julien 225
6.7 Paiement CCVS 226
6.7.2 ccvs_auth() : Perform credit authorization test on a transaction 226
6.7.3 ccvs_command() : Performs a command which is peculiar to a single protocol, and thus is not available in the general CCVS API 227
6.7.4 ccvs_count() : Find out how many transactions of a given type are stored in the system 227
6.7.5 ccvs_delete() : Delete a transaction 227
6.7.6 ccvs_done() : Terminate CCVS engine and do cleanup work 227

Sommaire

6 Index des fonctions

6.7.7 ccvs_init() : Initialize CCVS for use....227
6.7.8 ccvs_lookup() : Look up an item of a particular type in the database #....228
6.7.9 ccvs_new() : Create a new, blank transaction....228
6.7.10 ccvs_report() : Return the status of the background communication process....228
6.7.11 ccvs_return() : Transfer funds from the merchant to the credit card holder....228
6.7.12 ccvs_reverse() : Perform a full reversal on an already−processed authorization....228
6.7.13 ccvs_sale() : Transfer funds from the credit card holder to the merchant....229
6.7.14 ccvs_status() : Check the status of an invoice....229
6.7.15 ccvs_textvalue() : Get text return value for previous function call....229
6.7.16 ccvs_void() : Perform a full reversal on a completed transaction....229
6.8 Support COM pour Windows....230
6.8.1 Introduction....230
6.8.2 Pré−requis....230
6.8.3 Installation....230
6.8.4 Configuration à l'exécution....230
6.8.5 Constantes Prédefinies....230
6.8.6 Voir aussi....231
6.8.8 VARIANT() : classe VARIANT....232
6.8.9 com_addref() : Incrémente le compteur de références....232
6.8.10 com_get() : Lit la valeur d'un propriété d'un composant COM....232
6.8.11 com_invoke() : Appelle une méthode d'un composant....232
6.8.12 com_isenum() : Récupère un IEnumVariant....232
6.8.13 com_load_typelib() : Charge une librairie de type (Typelib)....232
6.8.14 com_load() : Crée une référence sur un composant COM....233
6.8.15 com_propget() : Lit la valeur d'un propriété d'un composant COM....233
6.8.16 com_propput() : Modifie une propriété d'un composant COM....233
6.8.17 com_propset() : Modifie une propriété d'un composant COM....233
6.8.18 com_release() : Décrémente le compteur de références....233
6.8.19 com_set() : Modifie une propriété d'un composant COM....233
6.9 Classes et Objets....234
6.9.1 Introduction....234
6.9.3 call_user_method() : Appelle une méthode utilisateur d'un objet....236
6.9.4 class_exists() : Vérifie qu'une classe a été définie....236
6.9.5 get_class_methods() : Retourne les noms des méthodes d'une classe....237
6.9.6 get_class_vars() : Retourne les valeurs par défaut des attributs d'une classe....237
6.9.7 get_class() : Retourne la classe d'un objet....238
6.9.8 get_declared_classes() : Liste toutes les classes définies....238
6.9.9 get_object_vars() : Retourne un tableau associatif des propriétés d'un objet....239
6.9.10 get_parent_class() : Retourne le nom de la classe d'un objet....239
6.9.11 is_a() : Retourne TRUE si un objet a pour parent une classe donnée....240
6.9.12 is_subclass_of() : Détermine si un objet est une sous−classe....240
6.9.13 method_exists() : Vérifie que la méthode existe pour une classe....240
6.10 ClibPDF....241
6.10.1 Introduction....241
6.10.2 Voir aussi....243
6.10.4 cpdf_add_outline() : Ajoute un signet à la page courante....244
6.10.5 cpdf_arc() : Dessine un arc de cercle....245
6.10.6 cpdf_begin_text() : Démarre une section de texte....245
6.10.7 cpdf_circle() : Dessine un cercle....245

Sommaire

6 Index des fonctions

6.10.8 cpdf_clip() : Aligne les dessins sur le chemin courant....246
6.10.9 cpdf_close() : Ferme un fichier PDF....246
6.10.10 cpdf_closepath_fill_stroke() : Remplit le chemin, dessine le bord et ferme le chemin....246
6.10.11 cpdf_closepath_stroke() : Ferme le fichier et dessine une ligne le long du chemin....246
6.10.12 cpdf_closepath() : Ferme le chemin....246
6.10.13 cpdf_continue_text() : Imprime le texte à la ligne suivante....247
6.10.14 cpdf_curveto() : Dessine une courbe....247
6.10.15 cpdf_end_text() : Termine une section de texte....247
6.10.16 cpdf_fill_stroke() : Remplit le chemin, et dessine le bord....247
6.10.17 cpdf_fill() : Remplit le chemin courant....248
6.10.18 cpdf_finalize_page() : Termine une page....248
6.10.19 cpdf_finalize() : Termine un document....248
6.10.20 cpdf_global_set_document_limits() : Fixe les limites d'un document PDF....248
6.10.21 cpdf_import_jpeg() : Ouvre une image JPEG....249
6.10.22 cpdf_lineto() : Dessine une ligne....249
6.10.23 cpdf_moveto() : Fixe le point courant....249
6.10.24 cpdf_newpath() : Commence un nouveau chemin....249
6.10.25 cpdf_open() : Ouvre un nouveau document PDF....250
6.10.26 cpdf_restore() : Restaure un environnement....250
6.10.27 cpdf_rlineto() : Dessine une ligne, relativement....250
6.10.28 cpdf_rmoveto() : Fixe le point courant relativement....251
6.10.29 cpdf_rotate() : Effectue une rotation....251
6.10.30 cpdf_save_to_file() : Ecrit un document PDF dans un fichier....251
6.10.31 cpdf_save() : Sauve l'environnement courant....251
6.10.32 cpdf_scale() : Modifie l'échelle....252
6.10.33 cpdf_set_char_spacing() : Fixe l'espacement des caractères....252
6.10.34 cpdf_set_creator() : Fixe le créateur d'un document PDF....252
6.10.35 cpdf_set_current_page() : Fixe la page courante....252
6.10.36 cpdf_set_font() : Sélectionne la police courante et sa taille....252
6.10.37 cpdf_set_horiz_scaling() : Fixe l'échelle horizontale du texte....253
6.10.38 cpdf_set_keywords() : Fixe les mot clés d'un document PDF....253
6.10.39 cpdf_set_leading() : Fixe la distance entre deux lignes....253
6.10.40 cpdf_set_page_animation() : Fixe l'animation de la transition entre les pages....253
6.10.41 cpdf_set_subject() : Fixe le sujet d'un document PDF....254
6.10.42 cpdf_set_text_matrix() : Fixe la matrice du texte....254
6.10.43 cpdf_set_text_pos() : Fixe la position du texte....254
6.10.44 cpdf_set_text_rendering() : Détermine le rendu du texte....254
6.10.45 cpdf_set_text_rise() : Fixe l'élévation du texte....254
6.10.46 cpdf_set_title() : Fixe le titre d'un document PDF....255
6.10.47 cpdf_set_word_spacing() : Fixe l'espacement des mots....255
6.10.48 cpdf_setdash() : Fixe le motif de pointillé....255
6.10.49 cpdf_setflat() : Fixe la platitude (flatness)....255
6.10.50 cpdf_setgray_fill() : Modifie le niveau de gris comme couleur de remplissage....255
6.10.51 cpdf_setgray_stroke() : Choisit un niveau de gris comme couleur de dessin....255
6.10.52 cpdf_setgray() : Modifie un niveau de gris comme couleur de dessin et de remplissage....256

Sommaire

6 Index des fonctions

6.10.53 cpdf_setlinecap() : Fixe le paramètre linecap....256
6.10.54 cpdf_setlinejoin() : Fixe le paramètre linejoin....256
6.10.55 cpdf_setlinewidth() : Fixe la largeur de ligne....256
6.10.56 cpdf_setmiterlimit() : Fixe le paramètre miter limit....256
6.10.57 cpdf_setrgbcolor_fill() : Choisit une couleur rgb comme couleur de remplissage....257
6.10.58 cpdf_setrgbcolor_stroke() : Choisit une couleur rgb comme couleur de dessin....257
6.10.59 cpdf_setrgbcolor() : Choisit une couleur rgb comme couleur de dessin et de remplissage....257
6.10.60 cpdf_show_xy() : Affiche un texte à une position....257
6.10.61 cpdf_show() : Imprime un texte à la position courante....258
6.10.62 cpdf_stringwidth() : Retourne la taille de la chaîne....258
6.10.63 cpdf_stroke() : Dessine une ligne le long du chemin....258
6.10.64 cpdf_text() : Imprime un texte avec des options....258
6.10.65 cpdf_translate() : Modifie l'origine du système de coordonnées....259
6.11 Crack....260
6.11.1 Introduction....260
6.11.2 Pré−requis....260
6.11.3 Installation....260
6.11.4 Configuration à l'exécution....260
6.11.5 Types de ressources....260
6.11.6 Constantes Prédefinies....260
6.11.7 Exemples....260
6.11.9 crack_closedict() : Referme le dictionnaire CrackLib....261
6.11.10 crack_getlastmessage() : Retourne le message de diagnostic....261
6.11.11 crack_opendict() : Ouvre un nouveau dictionnaire CrackLib dictionary....262
6.12 CURL....263
6.12.1 Introduction....263
6.12.2 Pré−requis....263
6.12.3 Installation....263
6.12.4 Constantes Prédefinies....263
6.12.5 Exemples....266
6.12.7 curl_errno() : Retourne un entier représentant le dernier message d'erreur....267
6.12.8 curl_error() : Retourne une chaîne contenant le dernier message d'erreur....267
6.12.9 curl_exec() : Exécute une session CURL....267
6.12.10 curl_getinfo() : Lit des informations concernant un transfert particulier....267
6.12.11 curl_init() : Initialise une session CURL....267
6.12.12 curl_setopt() : Modifie une option de transfert CURL....268
6.12.13 curl_version() : Retourne la version courante de CURL....270
6.13 Paiement Cybercash....271
6.13.2 cybercash_base64_encode() :....271
6.13.3 cybercash_decr() :....271
6.13.4 cybercash_encr() :....271
6.14 CyberMUT : Crédit Mutuel....272
6.14.2 cybermut_creerreponsecm() : Génère un accusé de réception de confirmation de paiement....272
6.14.3 cybermut_testmac() : Vérifie le message de confirmation....272
6.15 Cyrus IMAP....274
6.15.1 Introduction....274
6.15.2 Constantes Prédefinies....274

Sommaire

6 Index des fonctions

6.15.4 cyrus_bind() : Ajoute une fonction de callback sur une connexion Cyrus IMAP....274
6.15.5 cyrus_close() : Ferme la connexion à un serveur Cyrus IMAP....274
6.15.6 cyrus_connect() : Se connecte à un serveur Cyrus IMAP....275
6.15.7 cyrus_query() : Envoye une requête à un serveur Cyrus IMAP....275
6.15.8 cyrus_unbind() : Supprime une fonction de callback sur une connexion Cyrus IMAP....275
6.16 Caractères....276
6.16.2 ctype_alpha() : Vérifie qu'un caractère est alphabétique....276
6.16.3 ctype_cntrl() : Vérifie qu'un caractère est un caractère de contrôle....276
6.16.4 ctype_digit() : Vérifie qu'un caractère est numérique....276
6.16.5 ctype_graph() : Vérifie qu'un caractère est imprimable (sauf " ", espace)....277
6.16.6 ctype_lower() : Vérifie qu'un caractère est en minuscule....277
6.16.7 ctype_print() : Vérifie qu'un caractère est imprimable....277
6.16.8 ctype_punct() : Vérifie qu'un caractère est imprimable, sans être ni un espace, ni un caractère alpha−numérique....277
6.16.9 ctype_space() : Vérifie qu'un caractère est caractère blanc (espace, tabulation...)....277
6.16.10 ctype_upper() : Vérifie qu'un caractère est en majuscule....277
6.16.11 ctype_xdigit() : Vérifie qu'un caractère représente un nombre héxadécimal....277
6.17 DBA....278
6.17.2 dba_delete() : Efface une entrée....279
6.17.3 dba_exists() : Vérifie qu'une clé existe....280
6.17.4 dba_fetch() : Lit les données liées à une clé....280
6.17.5 dba_firstkey() : Lit la première clé....280
6.17.6 dba_handlers() : Liste les gestionnaires dba disponibles....280
6.17.7 dba_insert() : Insère une entrée....281
6.17.8 dba_list() : Liste tous les fichiers de bases de données ouverts....281
6.17.9 dba_nextkey() : Lit la clé suivante....281
6.17.10 dba_open() : Ouvre une base de données....281
6.17.11 dba_optimize() : Optimise une base....282
6.17.12 dba_popen() : Ouvre une connexion persistante à une base de données....282
6.17.13 dba_replace() : Remplace ou insère une entrée....282
6.17.14 dba_sync() : Synchronise une base de données....283
6.18 Dates et heures....284
6.18.1 checkdate() : Valide une date/heure....284
6.18.3 getdate() : Retourne la date/heure....284
6.18.4 gettimeofday() : Retourne l'heure actuelle....285
6.18.5 gmdate() : Formate une date/heure GMT/CUT....285
6.18.6 gmmktime() : Retourne le timestamp UNIX d'une date GMT....285
6.18.7 gmstrftime() : Formate une date/heure GMT/CUT en fonction des paramétrages locaux....286
6.18.8 localtime() : Lit l'heure locale....286
6.18.9 microtime() : Retourne le timestamp UNIX actuel avec microsecondes....286
6.18.10 mktime() : Retourne le timestamp UNIX d'une date....287
6.18.11 strftime() : Formate une date/heure locale avec les options locales....288
6.18.12 strtotime() : Transforme un texte anglais en timestamp....289
6.18.13 time() : Retourne le timestamp UNIX actuel....290
6.19 dBase....291
6.19.2 dbase_close() : Ferme une base dBase....291

Sommaire

6 Index des fonctions

6.19.3 dbase_create() : Crée une base de données dBase....291
6.19.4 dbase_delete_record() : Efface un enregistrement dans une base dBase....292
6.19.5 dbase_get_record_with_names() : Lit un enregistrement dans une base, sous la forme d'un tableau associatif....292
6.19.6 dbase_get_record() : Lit un enregistrement dans une base dBase....293
6.19.7 dbase_numfields() : Compte le nombre de champs d'une base dBase....293
6.19.8 dbase_numrecords() : Compter le nombre d'enregistrements dans une base dBase....293
6.19.9 dbase_open() : Ouverture d'une base dBase....293
6.19.10 dbase_pack() : Compacte une base dBase....294
6.19.11 dbase_replace_record() : Remplace un enregistrement dans une base dBase....294
6.20 DBM....295
6.20.2 dbmclose() : Ferme une base de données dbm....295
6.20.3 dbmdelete() : Efface une valeur....295
6.20.4 dbmexists() : Indique si une valeur existe....296
6.20.5 dbmfetch() : Lit une valeur....296
6.20.6 dbmfirstkey() : Lit la première clé....296
6.20.7 dbminsert() : Insère une valeur....296
6.20.8 dbmnextkey() : Lit la clé suivante....296
6.20.9 dbmopen() : Ouvre une base de données dbm....297
6.20.10 dbmreplace() : Remplace une valeur....297
6.21 dbx....298
6.21.1 Introduction....298
6.21.2 Pré−requis....298
6.21.3 Installation....298
6.21.4 Configuration à l'exécution....298
6.21.5 Types de ressources....299
6.21.6 Constantes Prédefinies....299
6.21.8 dbx_compare() : Compare deux lignes afin de les trier....300
6.21.9 dbx_connect() : Ouvre une connexion à une base de données....300
6.21.10 dbx_error() : Rapporte le message d'erreur du dernier appel de fonction....302
6.21.11 dbx_escape_string() : Protège une chaîne de caractères pour l'utiliser dans une requête....302
6.21.12 dbx_query() : Envoie une requête et lit tous les résultats....303
6.21.13 dbx_sort() : Tri un résultat avec une fonction utilisateur....305
6.22 DB++....306
6.22.1 Introduction....306
6.22.2 Pré−requis....306
6.22.3 Installation....306
6.22.4 Configuration à l'exécution....306
6.22.5 Types de ressources....306
6.22.6 Constantes Prédefinies....307
6.22.8 dbplus_aql() : Exécute une requête AQL....309
6.22.9 dbplus_chdir() : Lit/modifie le chemin courant de base....309
6.22.10 dbplus_close() : Ferme une relation....310
6.22.11 dbplus_curr() : Lit la ligne courante dans une relation....310
6.22.12 dbplus_errcode() : Get error string for given errorcode or last error....310
6.22.13 dbplus_errno() : Lit le code d'erreur généré par la dernière opération....311
6.22.14 dbplus_find() : Ajoute une contrainte sur une relation....311
6.22.15 dbplus_first() : Lit la première ligne d'une relation....311

Sommaire

6 Index des fonctions

6.22.16 dbplus_flush() : Ecrit toutes les modifications apportées à une relation....312
6.22.17 dbplus_freealllocks() : Libère tous les verrous posés par le client....312
6.22.18 dbplus_freelock() : Lève un verrou en écriture sur une ligne....313
6.22.19 dbplus_freerlocks() : Lève tous les verrous sur les lignes d'une relation....313
6.22.20 dbplus_getlock() : Pose un verrou sur une ligne....313
6.22.21 dbplus_getunique() : Lit un identifiant unique pour une relation....314
6.22.22 dbplus_info() : Encore non implémentée....314
6.22.23 dbplus_last() : Lit la dernière ligne d'une relation....314
6.22.24 dbplus_lockrel() : Pose un verrou en écriture sur une relation....315
6.22.25 dbplus_next() : Lit la ligne suivante dans la relation....315
6.22.26 dbplus_open() : Ouvre un fichier de relation....315
6.22.27 dbplus_prev() : Lit la ligne précédente dans la relation....316
6.22.28 dbplus_rchperm() : Modifie les droits sur la relation....316
6.22.29 dbplus_rcreate() : Crée une nouvelle relation DB++....316
6.22.30 dbplus_rcrtexact() : Crée une copie exacte mais vide d'une relation....317
6.22.31 dbplus_rcrtlike() : Crée une copie exacte d'une relation, sans les indexes....317
6.22.32 dbplus_resolve() : Résoud les informations d'hôte pour une relation....317
6.22.33 dbplus_restorepos() : Pas encore implémenté....318
6.22.34 dbplus_rkeys() : Spécifie la nouvelle clé primaire d'une relation....318
6.22.35 dbplus_ropen() : Ouvre un fichier de relation local....318
6.22.36 dbplus_rquery() : Performe une requête AQL locale....318
6.22.37 dbplus_rrename() : Renomme une relation....319
6.22.38 dbplus_rsecindex() : Crée un nouvel index secondaire pour une relation....319
6.22.39 dbplus_runlink() : Détruit une relation....319
6.22.40 dbplus_rzap() : Supprime toutes les lignes d'une relation....320
6.22.41 dbplus_savepos() : Pas encore implémentée....320
6.22.42 dbplus_setindex() : Pas encore implémentée....320
6.22.43 dbplus_setindexbynumber() : Pas encore implémentée....320
6.22.44 dbplus_sql() : Exécute une requête SQL....321
6.22.45 dbplus_tcl() : Exécute du code TCL coté serveur....321
6.22.46 dbplus_tremove() : Supprime la ligne courante et retourne la nouvelle ligne courante....321
6.22.47 dbplus_undo() : Pas encore implémentée....322
6.22.48 dbplus_undoprepare() : Pas encore implémentée....322
6.22.49 dbplus_unlockrel() : Libère un verrou en écriture posé sur une relation....322
6.22.50 dbplus_unselect() : Retire une contrainte d'une relation....322
6.22.51 dbplus_update() : Modifie une ligne dans une relation....323
6.22.52 dbplus_xlockrel() : Pose un verrou exclusif sur une relation....323
6.22.53 dbplus_xunlockrel() : Lève un verrou exclusif posé sur une relation....323
6.23 Direct IO....324
6.23.1 Fonctions d'entrée/sortie directes....324
6.23.2 Installation....324
6.23.4 dio_fcntl() : Exécute une fonction fichier de langage C....324
6.23.5 dio_open() : Ovre un nouveau fichier....325
6.23.6 dio_read() : Lit des octets dans un fichier....325
6.23.7 dio_seek() : Déplace le pointeur interne de fichier....325
6.23.8 dio_stat() : Lit des informations sur le fichier....326
6.23.9 dio_tcsetattr() : Modifie les attributs du terminal et le taux de baud du port série....326
6.23.10 dio_truncate() : Tronque le fichier....327
6.23.11 dio_write() : Ecrit des données dans le fichier....327

Sommaire

6 Index des fonctions

6.24 Accès aux dossiers....328
6.24.2 chroot() : Change la racine....328
6.24.3 closedir() : Ferme le pointeur sur le dossier....328
6.24.4 dir() : Classe dossier....328
6.24.5 getcwd() : Retourne le dossier de travail....329
6.24.6 opendir() : Ouvre un dossier, et récupère un pointeur dessus....329
6.24.7 readdir() : Lit une entrée du dossier....329
6.24.8 rewinddir() : Retourne à la première entrée du dossier....330
6.25 DOM XML....331
6.25.2 domxml_attributes() : Retourne les attributs d'un noeud....333
6.25.3 domxml_children() : Retourne les fils d'un noeud....333
6.25.4 domxml_dumpmem() : Ecrit le document XML interne dans une chaîne....333
6.25.5 domxml_get_attribute() : Retourne un attribut d'un noeud....334
6.25.6 domxml_new_child() : Ajoute un nouveau fils....334
6.25.7 domxml_new_xmldoc() : Crée un document XML vide....334
6.25.8 domxml_root() : Retourne l'élément racine....335
6.25.9 domxml_set_attribute() : Modifie un attribut....335
6.25.10 xmldoc() : Crée un objet DOM pour un document XML....336
6.25.11 xmldocfile() : Crée un objet DOM à partir d'un fichier XML....336
6.25.12 xmltree() : Crée un arbre d'objet PHP, à partir d'un document XML....336
6.25.13 xpath_eval() : Evalue une expression xpath....336
6.25.14 xpath_new_context() : Crée un nouveau contexte xpath....337
6.26 .NET....338
6.27 Gestion des erreurs....339
6.27.1 Introduction....339
6.27.2 Pré−requis....339
6.27.3 Installation....339
6.27.4 Configuration à l'exécution....339
6.27.5 Constantes Prédefinies....341
6.27.6 Exemples....343
6.27.7 Voir aussi....344
6.27.9 error_reporting() : Fixe le niveau de rapport d'erreurs PHP....345
6.27.10 restore_error_handler() : Réactive l'ancienne fonction de gestion des erreurs....346
6.27.11 set_error_handler() : Choisi une fonction utilisateur comme gestionnaire d'erreurs....346
6.27.12 trigger_error() : Déclenche une erreur utilisateur....348
6.27.13 user_error() : Génère un message d'erreur utilisateur....348
6.28 FrontBase....349
6.28.1 Introduction....349
6.28.2 Pré−requis....349
6.28.3 Installation....349
6.28.4 Configuration à l'exécution....349
6.28.5 Types de ressources....350
6.28.6 Constantes Prédefinies....350
6.28.7 fbsql_affected_rows() : Lit le nombre de ligne affectées par la dernière requête....351
6.28.8 fbsql_autocommit() : Active ou désactive la validation automatique....352
6.28.9 fbsql_change_user() : Change le nom d'utilisateur de la session active....352
6.28.10 fbsql_close() : Ferme la connexion FrontBase....352
6.28.11 fbsql_commit() : Valide une transaction....353

Sommaire

6 Index des fonctions

6.28.12 fbsql_connect() : Ouvre une connexion à un serveur FrontBase....353
6.28.13 fbsql_create_blob() : Crée un BLOB....354
6.28.14 fbsql_create_clob() : Crée un CLOB....354
6.28.15 fbsql_create_db() : Crée une base de données....355
6.28.16 fbsql_data_seek() : Déplace le pointeur interne de résultat....355
6.28.17 fbsql_database_password() : Modifie/lit le mot de passe dans une base FBSQL....356
6.28.18 fbsql_db_query() : Envoie une requête à la base FrontBase....356
6.28.19 fbsql_db_status() : Lit le statut courant d'une base de données....356
6.28.20 fbsql_drop_db() : Supprime une base de données FrontBase....357
6.28.21 fbsql_errno() : Retourne le code d'erreur FrontBase....357
6.28.22 fbsql_error() : Retourne le message d'erreur FrontBase....358
6.28.23 fbsql_fetch_array() : Lit toute une ligne de résultat dans un tableau....358
6.28.24 fbsql_fetch_assoc() : Lit toute une ligne de résultat dans un tableau associatif....359
6.28.25 fbsql_fetch_field() : Lit des informations sur une colonne dans un résultat, et retourne un objet....359
6.28.26 fbsql_fetch_lengths() : Lit la taille de chaque colonne d'un résultat....360
6.28.27 fbsql_fetch_object() : Lit une ligne de résultat sous forme d'objet....361
6.28.28 fbsql_fetch_row() : Lit une ligne de résultat sous forme de tableau numérique....361
6.28.29 fbsql_field_flags() : Lit les options associé à une colonne de résultat....362
6.28.30 fbsql_field_len() : Retourne la taille d'un champs....362
6.28.31 fbsql_field_name() : Lit le nom d'un champs....362
6.28.32 fbsql_field_seek() : Déplace le pointeur de résultat....363
6.28.33 fbsql_field_table() : Lit le nom de la table d'origine d'un champs....363
6.28.34 fbsql_field_type() : Lit le type d'une colonne....363
6.28.35 fbsql_free_result() : Libère le résultat de la mémoire....364
6.28.36 fbsql_insert_id() : Lit le dernier identifiant généré par une requête INSERT....364
6.28.37 fbsql_list_dbs() : Liste les bases de données....364
6.28.38 fbsql_list_fields() : Liste les champs d'un résultat FrontBase....365
6.28.39 fbsql_list_tables() : Liste les tables dans une base de données FrontBase....365
6.28.40 fbsql_next_result() : Déplace le pointeur interne vers le résultat suivant....366
6.28.41 fbsql_num_fields() : Lit le nombre de champs dans un résultat....366
6.28.42 fbsql_num_rows() : Lit le nombre de lignes dans un résultat....366
6.28.43 fbsql_pconnect() : Ouvre une connexion persistante à un serveur FrontBase....367
6.28.44 fbsql_query() : Exécute une requête sur un serveur FrontBase....367
6.28.45 fbsql_read_blob() : Lit un BLOB dans une base de données....368
6.28.46 fbsql_read_clob() : Lit un CLOB dans une base de données....369
6.28.47 fbsql_result() : Lit des données dans un résultat....369
6.28.48 fbsql_rollback() : Annule une transaction....370
6.28.49 fbsql_select_db() : Sélectionne une base de données FrontBase....370
6.28.50 fbsql_set_lob_mode() : Modifie le mode de lecture des LOB....371
6.28.51 fbsql_start_db() : Démarre une base de données....371
6.28.52 fbsql_stop_db() : Stoppe une base de données....371
6.28.53 fbsql_tablename() : Lit le nom de la table d'un champs....372
6.28.54 fbsql_warnings() : Active ou désactive les alertes FrontBase....372
6.29 FilePro....373
6.29.1 filepro_fieldcount() : Retourne le nombre de champs dans une base filePro....373
6.29.3 filepro_fieldtype() : Retourne le type d'un champs....373

Sommaire

6 Index des fonctions

6.29.4 filepro_fieldwidth() : Retourne la taille d'un champs....373
6.29.5 filepro_retrieve() : Retourne la valeur d'un champs....373
6.29.6 filepro_rowcount() : Retourne le nombre de champs dans une base filePro....374
6.29.7 filepro() : Lit et vérifie un fichier....374
6.30 Système de fichiers....375
6.30.1 Introduction....375
6.30.2 Pré−requis....375
6.30.3 Installation....375
6.30.4 Configuration à l'exécution....375
6.30.5 Types de ressources....376
6.30.6 Constantes Prédefinies....376
6.30.7 Voir aussi....376
6.30.9 chgrp() : Change le groupe possesseur du fichier....378
6.30.10 chmod() : Change le mode du fichier....378
6.30.11 chown() : Change le groupe propriétaire du fichier....379
6.30.12 clearstatcache() : Efface le cache de stat....379
6.30.13 copy() : Copie un fichier....379
6.30.14 delete() : Effacer....380
6.30.15 dirname() : Renvoie le nom du dossier....380
6.30.16 disk_free_space() : Renvoie l'espace disque disponible dans le répertoire....380
6.30.17 disk_total_space() : Retourne la taille d'un dossier....381
6.30.18 diskfreespace() : Alias de disk_free_space....381
6.30.19 fclose() : Ferme un fichier....381
6.30.20 feof() : Teste la fin du fichier....381
6.30.21 fflush() : Envoi tout le contenu généré dans un fichier....382
6.30.22 fgetc() : Renvoie le caractère que pointe le pointeur du fichier....382
6.30.23 fgetcsv() : Renvoie la ligne courante et cherche les champs CSV....382
6.30.24 fgets() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier....383
6.30.25 fgetss() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier et élimine les balises HTML....383
6.30.26 file_exists() : Vérifie si un fichier existe....384
6.30.27 file_get_contents() : Lit tout un fichier dans une chaîne....384
6.30.28 file() : Lit le fichier et renvoie le résultat dans un tableau....384
6.30.29 fileatime() : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois....385
6.30.30 filectime() : Renvoie l'heure à laquelle l'inode a été accédé pour la dernière fois....385
6.30.31 filegroup() : Lire le nom du groupe....386
6.30.32 fileinode() : Renvoie le numéro d'inode du fichier....386
6.30.33 filemtime() : Renvoie la date de dernière modification du fichier....386
6.30.34 fileowner() : Renvoie le nom du propriétaire du fichier....387
6.30.35 fileperms() : Renvoie les permissions affectées au fichier....387
6.30.36 filesize() : Renvoie la taille du fichier....387
6.30.37 filetype() : Retourne le type de fichier....387
6.30.38 flock() : Verrouille le fichier....388
6.30.39 fnmatch() : Repère un fichier à partir d'un masque....388
6.30.40 fopen() : Ouverture d'un fichier ou d'une URL....389
6.30.41 fpassthru() : Affiche la partie du fichier située après le pointeur du fichier....390
6.30.42 fputs() : Ecrit dans un fichier....390

Sommaire

6 Index des fonctions

6.30.43 fread() : Lecture du fichier en mode binaire....391
6.30.44 fscanf() : Analyse un fichier en fonction d'un format....391
6.30.45 fseek() : Modifie le pointeur de fichier....392
6.30.46 fstat() : Lit les informations sur un fichier à partir d'un pointeur de fichier....392
6.30.47 ftell() : Renvoie la position du pointeur du fichier....393
6.30.48 ftruncate() : Tronque un fichier....393
6.30.49 fwrite() : Ecriture du fichier en mode binaire....393
6.30.50 glob() : Recheche des chemins qui vérifie un masque....394
6.30.51 is_dir() : Indique si le nom de fichier est un dossier....394
6.30.52 is_executable() : Indique si le fichier est exécutable....394
6.30.53 is_file() : Indique si le fichier est un véritable fichier....395
6.30.54 is_link() : Indique si le fichier est un lien symbolique....395
6.30.55 is_readable() : Indique un fichier est autorisé en lecture....395
6.30.56 is_uploaded_file() : Indique si le fichier a été téléchargé par HTTP POST....396
6.30.57 is_writable() : Indique si un fichier est autorisé en écriture....396
6.30.58 is_writeable() : Indique si un fichier est autorisé en écriture....396
6.30.59 link() : Crée un lien....397
6.30.60 linkinfo() : Renvoie les informations à propos d'un lien....397
6.30.61 lstat() : Renvoie les informations à propos d'un fichier ou d'un lien symbolique....397
6.30.62 mkdir() : Crée un dossier....398
6.30.63 move_uploaded_file() : Déplace un fichier téléchargé....398
6.30.64 parse_ini_file() : Traite un fichier de configuration....399
6.30.65 pathinfo() : Retourne des informations sur un chemin système....400
6.30.66 pclose() : Ferme un processus de pointeur de fichier....400
6.30.67 popen() : Crée un processus de pointeur de fichier....400
6.30.68 readfile() : Affiche un fichier....401
6.30.69 readlink() : Renvoie le nom du fichier vers lequel pointe un lien symbolique....401
6.30.70 realpath() : Retourne le chemin canonique absolu....402
6.30.71 rename() : Renomme un fichier....402
6.30.72 rewind() : Replace le pointeur de fichier au début....402
6.30.73 rmdir() : Efface un dossier....402
6.30.74 set_file_buffer() : Fixe la bufferisation de fichier....403
6.30.75 stat() : Renvoie les informations à propos d'un fichier....403
6.30.76 stream_get_meta_data() : Retrieves header/meta data from streams/file pointers....404
6.30.77 stream_register_wrapper() : Enregistre une enveloppe URL, implémenté comme classe PHP....405
6.30.78 stream_set_blocking() : Configure le mode bloquant d'un flôt....408
6.30.79 stream_set_timeout() : Configure la durée d'expiration d'un flôt....408
6.30.80 stream_set_write_buffer() : Configure la bufferisation de fichier pour un flôt....409
6.30.81 symlink() : Crée un lien symbolique....410
6.30.82 tempnam() : Crée un fichier avec un nom unique....410
6.30.83 tmpfile() : Crée un fichier temporaire....410
6.30.84 touch() : Affecte une nouvelle date de modification à un fichier....411
6.30.85 umask() : Change le "umask" courant....411
6.30.86 unlink() : Efface un fichier....411
6.31 Forms Data Format....412
6.31.1 Introduction....412
6.31.2 Pré−requis....412

Sommaire

6 Index des fonctions

6.31.3 Installation....412
6.31.4 Configuration à l'exécution....413
6.31.5 Types de ressources....413
6.31.6 Constantes Prédefinies....413
6.31.7 Exemples....413
6.31.9 fdf_create() : Crée un nouveau document FDF....415
6.31.10 fdf_get_file() : Lit la valeur de la clé /F....415
6.31.11 fdf_get_status() : Lit la valeur de la clé /STATUS....415
6.31.12 fdf_get_value() : Mot la valeur d'un champs....415
6.31.13 fdf_next_field_name() : Lit le nom du champs suivant....416
6.31.14 fdf_open() : Ouvre un document FDF....416
6.31.15 fdf_save() : Sauver un document FDF....416
6.31.16 fdf_set_ap() : Fixe l'apparence d'un champs....417
6.31.17 fdf_set_encoding() : Modifie l'encodage des caractères....417
6.31.18 fdf_set_file() : Fixe la valeur de la clé /F....417
6.31.19 fdf_set_flags() : Modifie une option d'un champs....417
6.31.20 fdf_set_javascript_action() : Modifie l'action javascript d'un champs....417
6.31.21 fdf_set_opt() : Modifie une option d'un champs....418
6.31.22 fdf_set_status() : Fixe la valeur de la clé /STATUS....418
6.31.23 fdf_set_submit_form_action() : Modifie l'action d'un formulaire....418
6.31.24 fdf_set_value() : Fixe la valeur d'un champs....418
6.32 FriBiDi....419
6.32.1 Introduction....419
6.32.2 Pré−requis....419
6.32.3 Installation....419
6.32.4 Configuration à l'exécution....419
6.32.5 Types de ressources....419
6.32.6 Constantes Prédefinies....419
6.33 FTP....420
6.33.1 Introduction....420
6.33.2 Pré−requis....420
6.33.3 Installation....420
6.33.4 Configuration à l'exécution....420
6.33.5 Types de ressources....420
6.33.6 Constantes Prédefinies....420
6.33.7 Exemples....421
6.33.9 ftp_chdir() : Change le dossier courant....422
6.33.10 ftp_close() : Ferme une connexion FTP....422
6.33.11 ftp_connect() : Ouvre une connexion FTP....422
6.33.12 ftp_delete() : Efface un fichier sur un serveur FTP....423
6.33.13 ftp_exec() : Exécute un programme sur le serveur FTP....423
6.33.14 ftp_fget() : Télécharge un fichier depuis un serveur FTP et le sauve dans un fichier déjà ouvert....423
6.33.15 ftp_fput() : Charge un fichier ouvert sur un serveur FTP....423
6.33.16 ftp_get_option() : Lit différenctes options pour la connexion FTP courante....423
6.33.17 ftp_get() : Télécharge un fichier depuis un serveur FTP....424
6.33.18 ftp_login() : Authentification d'une connexion FTP....424
6.33.19 ftp_mdtm() : Retourne la date de dernière modification d'un fichier sur un serveur FTP....424
6.33.20 ftp_mkdir() : Crée un dossier....425
6.33.21 ftp_nb_continue() : Reprend le téléchargement d'un fichier (non bloquant)....425

Sommaire

6 Index des fonctions

6.33.22 ftp_nb_fget() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant) 425
6.33.23 ftp_nb_fput() : Ecrit un fichier sur un serveur FTP, et le lit depuis un fichier (non bloquant) 425
6.33.24 ftp_nb_get() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant) 426
6.33.25 ftp_nb_put() : Envoie un fichier sur le serveur FTP (non−bloquant) 427
6.33.26 ftp_nlist() : Retourne la liste des fichiers dans un dossier 428
6.33.27 ftp_pasv() : Active ou désactive le mode passif 428
6.33.28 ftp_put() : Charge un fichier sur un serveur FTP 428
6.33.29 ftp_pwd() : Retourne le nom du dossier courant 429
6.33.30 ftp_quit() : Ferme une connexion FTP 429
6.33.31 ftp_rawlist() : Fait une liste détaillée de fichiers dans un dossier 429
6.33.32 ftp_rename() : Renomme un fichier sur un serveur FTP 429
6.33.33 ftp_rmdir() : Efface un dossier 429
6.33.34 ftp_set_option() : Modifie les options de la connexion FTP 429
6.33.35 ftp_site() : Envoie la commande SITE au serveur 430
6.33.36 ftp_size() : Retourne la taille d'un fichier 430
6.33.37 ftp_systype() : Retourne un identifiant de type de serveur FTP 430
6.34 Fonctions 432
6.34.2 call_user_func() : Appelle une fonction utilisateur 432
6.34.3 create_function() : Crée une fonction anonyme (style lambda) 432
6.34.4 func_get_arg() : Retourne un élément de la liste des arguments 434
6.34.5 func_get_args() : Retourne les arguments d'une fonction sous forme de tableau 435
6.34.6 func_num_args() : Retourne le nombre d'arguments passé à la fonction 435
6.34.7 function_exists() : Indique si une fonction est définie 436
6.34.8 get_defined_functions() : Liste toutes les fonctions définies 436
6.34.9 register_shutdown_function() : Enregistre une fonction pour exécution à l'extinction 437
6.34.10 register_tick_function() : Enregistre une fonction exécutée à chaque tick 437
6.34.11 unregister_tick_function() : Annule la fonction exécutée à chaque tick 437
6.35 Gettext (GNU) 439
6.35.2 dcgettext() : Remplace le domaine lors d'une recherche 439
6.35.3 dgettext() : Remplace le domaine courant 439
6.35.4 gettext() : Recherche un message dans le domaine courant 439
6.35.5 textdomain() : Fixe le domaine par défaut 440
6.36 GMP 441
6.36.1 Introduction 441
6.36.2 Pré−requis 441
6.36.3 Installation 441
6.36.4 Configuration à l'exécution 441
6.36.5 Types de ressources 441
6.36.6 Constantes Prédefinies 442
6.36.7 Exemples 442
6.36.9 gmp_add() : Addition de 2 nombres GMP 443
6.36.10 gmp_and() : ET logique 443
6.36.11 gmp_clrbit() : Annule un bit 443
6.36.12 gmp_cmp() : Compare des nombres GMP 443
6.36.13 gmp_com() : Calcule le complémentaire d'un nombre 444
6.36.14 gmp_div_q() : Divisions de 2 nombres GMP 444

Sommaire

6 Index des fonctions

6.36.15 gmp_div_qr() : Divise deux nombres GMP....444
6.36.16 gmp_div_r() : Reste de la division de deux nombres GMP....444
6.36.17 gmp_div() : Divise deux nombres GMP....445
6.36.18 gmp_divexact() : Division exacte de nombres GMP....445
6.36.19 gmp_fact() : Factorielle GMP....445
6.36.20 gmp_gcd() : PGCD....445
6.36.21 gmp_gcdext() : PGCD étendu....445
6.36.22 gmp_hamdist() : Distance de Hamming....446
6.36.23 gmp_init() : Crée un nombre GMP....446
6.36.24 gmp_intval() : Convertit un nombre GMP en entier....446
6.36.25 gmp_invert() : Inverse modulo....447
6.36.26 gmp_jacobi() : Symbole de Jacobi....447
6.36.27 gmp_legendre() : Symbole de Legendre....447
6.36.28 gmp_mod() : Modulo GMP....447
6.36.29 gmp_mul() : Multiplication de 2 nombres GMP....447
6.36.30 gmp_neg() : Opposé de nombre GMP....447
6.36.31 gmp_or() : OU logique....447
6.36.32 gmp_perfect_square() : Carré parfait GMP....448
6.36.33 gmp_popcount() : Compte de population....448
6.36.34 gmp_pow() : Puissance....448
6.36.35 gmp_powm() : Puissance et modulo....448
6.36.36 gmp_prob_prime() : Nombre GMP probablement premier....448
6.36.37 gmp_random() : Nombre GMP aléatoire....449
6.36.38 gmp_scan0() : Recherche 0....449
6.36.39 gmp_scan1() : Recherche 1....449
6.36.40 gmp_setbit() : Modifie un bit....449
6.36.41 gmp_sign() : Signe du nombre GMP....449
6.36.42 gmp_sqrt() : Racine carrée GMP....449
6.36.43 gmp_sqrtrm() : Racine carrée avec reste GMP....450
6.36.44 gmp_strval() : Convertit un nombre GMP en chaîne....450
6.36.45 gmp_sub() : Soustraction de 2 nombres GMP....450
6.36.46 gmp_xor() : OU exclusif logique....450
6.37 HTTP....451
6.37.2 headers_sent() : Indique si les entêtes HTTP ont déjà été envoyés....451
6.37.3 setcookie() : Envoie un cookie....451
6.38 Hyperwave....453
6.38.1 Introduction....453
6.38.2 Pré−requis....455
6.38.3 Installation....455
6.38.4 Intégration avec Apache....455
6.38.5 Configuration à l'exécution....456
6.38.6 Types de ressources....457
6.38.7 Constantes Prédefinies....457
6.38.8 A faire....457
6.38.10 hw_Children() : Liste des object ids des objets fils....458
6.38.11 hw_ChildrenObj() : Liste des object records des objets fils....458
6.38.12 hw_Close() : Ferme la connexion Hyperwave....459
6.38.13 hw_Connect() : Ouvre une connexion Hyperwave....459
6.38.14 hw_Cp() : Copie des objets....459
6.38.15 hw_Deleteobject() : Efface des objets....459
6.38.16 hw_DocByAnchor() : Identifiant d'objet de l'objet dans l'ancrage....460

Sommaire

6 Index des fonctions

6.38.17 hw_DocByAnchorObj() : Attributs de l'objet dans l'ancrage....460
6.38.18 hw_Document_Attributes() : Object record de hw_document....460
6.38.19 hw_Document_BodyTag() : Balise de corps d'un document....460
6.38.20 hw_Document_Content() : Contenu d'un document....460
6.38.21 hw_Document_SetContent() : Modifie/remplace le contenu d'un document....461
6.38.22 hw_Document_Size() : Taille d'un document....461
6.38.23 hw_EditText() : Retourne un document texte....461
6.38.24 hw_Error() : Retourne le code d'erreur....462
6.38.25 hw_ErrorMsg() : Retourne un message d'erreur....462
6.38.26 hw_Free_Document() : Détruit un document....462
6.38.27 hw_GetAnchors() : Identifiants des ancrages d'un document....462
6.38.28 hw_GetAnchorsObj() : Attributs des ancrages d'un document....462
6.38.29 hw_GetAndLock() : Retourne les attributs, et verrouille l'objet....462
6.38.30 hw_GetChildColl() : Identifiant d'objet des groupes fils....463
6.38.31 hw_GetChildCollObj() : object records d'un groupe d'enfants....463
6.38.32 hw_GetChildDocColl() : ids des documents fils d'un groupe....463
6.38.33 hw_GetChildDocCollObj() : Attributs des documents fils d'un groupe....463
6.38.34 hw_GetObject() : Attributs d'un objet....463
6.38.35 hw_GetObjectByQuery() : Recherche un objet....464
6.38.36 hw_GetObjectByQueryColl() : Recherche un objet dans un groupe....464
6.38.37 hw_GetObjectByQueryCollObj() : Recherche un objet dans un groupe....465
6.38.38 hw_GetObjectByQueryObj() : Recherche un objet....465
6.38.39 hw_GetParents() : Identifiant d'objet des parents....465
6.38.40 hw_GetParentsObj() : Attributs des parents....465
6.38.41 hw_GetRemote() : Retourne un document distant....466
6.38.42 hw_GetRemoteChildren() : Retourne les fils d'un document distant....466
6.38.43 hw_GetSrcByDestObj() : Retourne les ancrages qui pointent sur un objet....466
6.38.44 hw_GetText() : Retourne un document texte....467
6.38.45 hw_Username() : Nom de l'utilisateur actuellement identifié....467
6.38.46 hw_Identify() : Identifie un utilisateur....467
6.38.47 hw_InCollections() : Vérifie qu'un identifiant d'objet est dans un groupe....468
6.38.48 hw_Info() : Informations à propos d'une connexion....468
6.38.49 hw_InsColl() : Insère un groupe....468
6.38.50 hw_InsDoc() : Insère un document....468
6.38.51 hw_InsertDocument() : Insère un document dans un groupe....468
6.38.52 hw_InsertObject() : Insère un object record....469
6.38.53 hw_mapid() : Représente un id globale en un id virtuel local....469
6.38.54 hw_Modifyobject() : Modifie les attributs d'objet record....469
6.38.55 hw_Mv() : Déplace un objet....471
6.38.56 hw_New_Document() : Crée un nouveau document....472
6.38.57 hw_Objrec2Array() : Convertit les attributs d'un objet en tableau....472
6.38.58 hw_Output_Document() : Affiche hw_document....472
6.38.59 hw_pConnect() : Crée une connexion persistante....472
6.38.60 hw_PipeDocument() : Retourne un document....473
6.38.61 hw_Root() : Object id de la racine....473
6.38.62 hw_Unlock() : Déverrouille un objet....473
6.38.63 hw_Who() : Liste des utilisateurs actuellement identifiés....473
6.39 Hyperwave API....474
6.39.1 Introduction....474
6.39.2 Pré−requis....474

Sommaire

6 Index des fonctions

6.39.3 Installation....474
6.39.4 intégration dans Apache....474
6.39.5 Configuration à l'exécution....474
6.39.6 Types de ressources....475
6.39.7 Constantes Prédefinies....475
6.39.8 Classes....475
6.39.10 hw_api_attribute->langdepvalue() : Returns value for a given language....477
6.39.11 hw_api_attribute->value() : Returns value of the attribute....477
6.39.12 hw_api_attribute->values() : Returns all values of the attribute....477
6.39.13 hw_api_attribute() : Creates instance of class hw_api_attribute....477
6.39.14 hw_api->checkin() : Checks in an object....477
6.39.15 hw_api->checkout() : Checks out an object....478
6.39.16 hw_api->children() : Returns children of an object....478
6.39.17 hw_api_content->mimetype() : Returns mimetype....478
6.39.18 hw_api_content->read() : Read content....479
6.39.19 hw_api->content() : Returns content of an object....479
6.39.20 hw_api->copy() : Copies physically....479
6.39.21 hw_api->dbstat() : Returns statistics about database server....479
6.39.22 hw_api->dcstat() : Returns statistics about document cache server....479
6.39.23 hw_api->dstanchors() : Returns a list of all destination anchors....480
6.39.24 hw_api->dstofsrcanchors() : Returns destination of a source anchor....480
6.39.25 hw_api_error->count() : Returns number of reasons....480
6.39.26 hw_api_error->reason() : Returns reason of error....480
6.39.27 hw_api->find() : Search for objects....480
6.39.28 hw_api->ftstat() : Returns statistics about fulltext server....481
6.39.29 hwapi_hgcsp() : Returns object of class hw_api....481
6.39.30 hw_api->hwstat() : Returns statistics about Hyperwave server....481
6.39.31 hw_api->identify() : Log into Hyperwave Server....481
6.39.32 hw_api->info() : Returns information about server configuration....481
6.39.33 hw_api->insert() : Inserts a new object....482
6.39.34 hw_api->insertanchor() : Inserts a new object of type anchor....482
6.39.35 hw_api->insertcollection() : Inserts a new object of type collection....482
6.39.36 hw_api->insertdocument() : Inserts a new object of type document....483
6.39.37 hw_api->link() : Creates a link to an object....483
6.39.38 hw_api->lock() : Locks an object....483
6.39.39 hw_api->move() : Moves object between collections....483
6.39.40 hw_api_content() : Create new instance of class hw_api_content....484
6.39.41 hw_api_object->assign() : Clones object....484
6.39.42 hw_api_object->attreditable() : Checks whether an attribute is editable....484
6.39.43 hw_api_object->count() : Returns number of attributes....484
6.39.44 hw_api_object->insert() : Inserts new attribute....484
6.39.45 hw_api_object() : Creates a new instance of class hw_api_object....484
6.39.46 hw_api_object->remove() : Removes attribute....484
6.39.47 hw_api_object->title() : Returns the title attribute....485
6.39.48 hw_api_object->value() : Returns value of attribute....485
6.39.49 hw_api->object() : Retrieve attribute information....485
6.39.50 hw_api->objectbyanchor() : Returns the object an anchor belongs to....486
6.39.51 hw_api->parents() : Returns parents of an object....486
6.39.52 hw_api_reason->description() : Returns description of reason....486
6.39.53 hw_api_reason->type() : Returns type of reason....487
6.39.54 hw_api->remove() : Delete an object....487

Sommaire

6 Index des fonctions

6.39.55 hw_api−>replace() : Replaces an object....487
6.39.56 hw_api−>setcommitedversion() : Commits version other than last version....487
6.39.57 hw_api−>srcanchors() : Returns a list of all source anchors....488
6.39.58 hw_api−>srcsofdst() : Returns source of a destination object....488
6.39.59 hw_api−>unlock() : Unlocks a locked object....488
6.39.60 hw_api−>user() : Returns the own user object....488
6.39.61 hw_api−>userlist() : Returns a list of all logged in users....489
6.40 Iconv....490
6.40.1 Introduction....490
6.40.2 Pré−requis....490
6.40.3 Installation....490
6.40.4 Configuration à l'exécution....490
6.40.5 Types de ressources....490
6.40.6 Constantes Prédefinies....490
6.40.7 Voir aussi....490
6.40.9 iconv_set_encoding() : Modifie le jeu courant de caractères courant....491
6.40.10 iconv() : Convertit une chaîne dans un jeu de caractères....491
6.40.11 ob_iconv_handler() : Gestionnaire de sortie pour maitriser le jeu de caractères de sortie....491
6.41 Images....493
6.41.1 Introduction....493
6.41.2 Pré−requis....493
6.41.3 Installation....493
6.41.4 Configuration à l'exécution....494
6.41.5 Types de ressources....495
6.41.6 Constantes Prédefinies....495
6.41.7 Exemples....495
6.41.9 image2wbmp() : Crée une image WBMP....497
6.41.10 ImageAlphaBlending() : Modifie le mode de blending d'une image....498
6.41.11 ImageArc() : Dessine une ellipse partielle....498
6.41.12 ImageChar() : Dessine un caractère horizontalement....498
6.41.13 ImageCharUp() : Dessine un caractère verticalement....498
6.41.14 ImageColorAllocate() : Alloue une couleur pour une image....499
6.41.15 ImageColorAt() : Retourne l'index de la couleur d'un pixel donné....499
6.41.16 ImageColorClosest() : Retourne l'index de la couleur la plus proche d'une couleur donnée....499
6.41.17 ImageColorClosestAlpha() : Retourne la couleur la plus proche, en tenant compte du canal alpha....500
6.41.18 ImageColorDeAllocate() : Désalloue une couleur pour une image....500
6.41.19 ImageColorExact() : Retourne l'index de la couleur donnée....500
6.41.20 ImageColorExactAlpha() : Retourne l'index d'une couleur avec son canal alpha....500
6.41.21 ImageColorResolve() : Retourne l'index de la couleur donnée, ou la plus proche possible....501
6.41.22 ImageColorResolveAlpha() : Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha....501
6.41.23 ImageColorSet() : Change la couleur dans une palette à l'index donné....501
6.41.24 ImageColorsForIndex() : Retourne la couleur associée à un index....502
6.41.25 ImageColorsTotal() : Calcule le nombre de couleurs d'une palette....502
6.41.26 ImageColorTransparent() : Définit la couleur transparente....502
6.41.27 ImageCopy() : Copie une partie d'une image....502

Sommaire

6 Index des fonctions

6.41.28 ImageCopyMerge() : Copie et fusionne une partie d'une image....502
6.41.29 ImageCopyMergeGray() : Copie et fusionne une partie d'une image en niveaux de gris....503
6.41.30 ImageCopyResampled() : Copie, redimensionne, rééchantillonne une image....503
6.41.31 ImageCopyResized() : Copie et redimensionne une partie d'une image....504
6.41.32 ImageCreate() : Crée une nouvelle image à palette....504
6.41.33 imagecreatefromgif() : Crée une nouvelle image à partir d'un fichier ou d'une URL....504
6.41.34 ImageCreateFromJPEG() : Crée une nouvelle image JPEG à partir d'un fichier ou d'une URL....505
6.41.35 ImageCreateFromPNG() : Crée une nouvelle image PNG à partir d'un fichier ou d'une URL....505
6.41.36 ImageCreateFromString() : Crée une image à partir d'une chaîne....506
6.41.37 ImageCreateFromWBMP() : Crée une image depuis un fichier WBMP....506
6.41.38 ImageCreateFromXBM() : Crée une image à partir d'un fichier XBM....507
6.41.39 ImageCreateFromXPM() : Crée une image à partir d'un fichier XPM....507
6.41.40 ImageCreateTrueColor() : Crée une nouvelle image en vraies couleurs....507
6.41.41 ImageDashedLine() : Dessine une ligne pointillée....507
6.41.42 ImageDestroy() : détruit une image....507
6.41.43 ImageEllipse() : Dessine une ellipse....508
6.41.44 ImageFill() : Remplit....508
6.41.45 imagefilledarc() : Dessine une ellipse partielle et la remplit....508
6.41.46 ImageFilledEllipse() : Dessine une ellipse pleine....509
6.41.47 ImageFilledPolygon() : Dessine un polygone rempli....509
6.41.48 ImageFilledRectangle() : Dessine un rectangle rempli....509
6.41.49 ImageFillToBorder() : Remplit avec une région avec une couleur spécifique....509
6.41.50 ImageFontHeight() : Retourne la hauteur de la police....509
6.41.51 ImageFontWidth() : Retourne la largeur de la police....510
6.41.52 ImageGammaCorrect() : Applique une correction gamma à l'image....510
6.41.53 ImageGif() : Envoie une image GIF vers un navigateur ou un fichier....510
6.41.54 ImageInterlace() : Active ou désactive l'entrelacement....511
6.41.55 ImageJPEG() : Envoie une image JPEG vers un navigateur ou un fichier....511
6.41.56 ImageLine() : Dessine une ligne....512
6.41.57 ImageLoadFont() : Charge une nouvelle police....512
6.41.58 ImagePaletteCopy() : Copie la palette d'une image à l'autre....513
6.41.59 ImagePNG() : Envoie une image PNG vers un navigateur ou un fichier....513
6.41.60 ImagePolygon() : Dessine un polygone....513
6.41.61 ImagePSBBox() : Retourne le rectangle entourant un texte et dessiné avec une police PostScript Type1....514
6.41.62 imagepscopyfont() : Fait une copie d'une police chargée pour modification ultérieure....514
6.41.63 ImagePSEncodeFont() : Change le codage vectoriel d'un caractère dans une police....515
6.41.64 ImagePsExtendFont() : Etend ou condense une police de caractères....515
6.41.65 ImagePSFreeFont() : Libère la mémoire occupée par une police PostScript Type 1....515
6.41.66 ImagePSLoadFont() : Charge une police PostScript Type 1 depuis un fichier....515
6.41.67 ImagePsSlantFont() : Incline une police de caractères....516

Sommaire

6 Index des fonctions

6.41.68 ImagePSText() : Dessine un texte sur une image avec une police PostScript Type1....516
6.41.69 ImageRectangle() : Dessine un rectangle....517
6.41.70 imagesetbrush() : Modifie la brosse pour le dessin des lignes....517
6.41.71 ImageSetPixel() : Dessine un pixel....517
6.41.72 ImageSetThickness() : Modifie l'épaisseur d'un trait....518
6.41.73 ImageSetTile() : Modifie l'image utilisée pour le carrelage....518
6.41.74 ImageString() : Dessine une chaîne horizontale....518
6.41.75 ImageStringUp() : Dessine une chaîne verticale....519
6.41.76 ImageSX() : Retourne la largeur d'une image....519
6.41.77 ImageSY() : Retourne la hauteur de l'image....519
6.41.78 ImageTrueColorToPalette() : Convertit une image en vraies couleurs en image à palette....519
6.41.79 ImageTTFBBox() : Retourne le rectangle entourant un texte et dessiné avec une police TrueType....520
6.41.80 ImageTTFText() : Dessine un texte avec une police TrueType....520
6.41.81 ImageTypes() : Retourne les types d'images supportés par la version courante de PHP....521
6.41.82 ImageWBMP() : Affiche une image WBMP....522
6.41.83 iptcparse() : Analyse un bloc binaire IPTC http://www.iptc.org/ et recherche les balises simples....522
6.41.84 JPEG2WBMP() : Convertit une image JPEG en image WBMP....522
6.41.85 PNG2WBMP() : Convertit une image PNG en image WBMP....523
6.41.86 read_exif_data() : Lit les en−têtes EXIF d'une image JPEG....523
6.42 IMAP....524
6.42.1 Introduction....524
6.42.2 Pré−requis....524
6.42.3 Installation....524
6.42.4 Configuration à l'exécution....525
6.42.5 Types de ressources....525
6.42.6 Constantes Prédefinies....525
6.42.7 Voir aussi....526
6.42.9 imap_alerts() : Retourne toutes les alertes....528
6.42.10 imap_append() : Ajoute une chaîne dans une boîte aux lettres....528
6.42.11 imap_base64() : Décode un texte encodé en BASE64....528
6.42.12 imap_binary() : Convertit une chaîne à 8 bits en une chaîne à base64....529
6.42.13 imap_body() : Lit le corps d'un message....529
6.42.14 imap_check() : Vérifie le courrier de la boîte aux lettres courante....529
6.42.15 imap_clearflag_full() : Supprime un flag sur un message....530
6.42.16 imap_close() : Termine un flot IMAP....530
6.42.17 imap_createmailbox() : Crée une nouvelle boîte aux lettres....530
6.42.18 imap_delete() : Marque le fichier pour l'effacement, dans la boîte aux lettres courante....531
6.42.19 imap_deletemailbox() : Efface une boîte aux lettres....531
6.42.20 imap_errors() : Retourne toutes les erreurs....532
6.42.21 imap_expunge() : Efface tous les messages marqués pour l'effacement....532
6.42.22 imap_fetch_overview() : Lit un sommaire des en−têtes de messages....532
6.42.23 imap_fetchbody() : Retourne une section extraite du corps d'un message....533
6.42.24 imap_fetchheader() : Retourne l'en−tête d'un message....533
6.42.25 imap_fetchstructure() : Lit la structure d'un message....534
6.42.26 imap_get_quota() : Lit les quotas des boîtes aux lettres....535

Sommaire

6 Index des fonctions

6.42.27 imap_getmailboxes() : Liste les boîtes aux lettres, et retourne le détail pour chacune....536
6.42.28 imap_getsubscribed() : Liste toutes les boîtes aux lettres souscrites....537
6.42.29 imap_header() : Lit l'en−tête d'un message....537
6.42.30 imap_headerinfo() : Lit l'en−tête du message....537
6.42.31 imap_headers() : Retourne les en−têtes de tous les messages d'une boîte aux lettres....538
6.42.32 imap_last_error() : Retourne la dernière erreur (si elle existe) qui est survenu lors de la dernière requête....538
6.42.33 imap_listmailbox() : Liste les boîtes aux lettres....539
6.42.34 imap_listsubscribed() : Liste les boîtes aux lettres souscrites....539
6.42.35 imap_mail_compose() : Crée un message MIME....539
6.42.36 imap_mail_copy() : Copie les messages spécifiés dans une boîte aux lettres....540
6.42.37 imap_mail_move() : Déplace les messages spécifiés dans une boîte aux lettres....540
6.42.38 imap_mail() : Envoie un message mail....540
6.42.39 imap_mailboxmsginfo() : Lit les informations à propos de la boîte aux lettres courante....541
6.42.40 imap_mime_header_decode() : Décode les éléments MIME d'une en−tête....541
6.42.41 imap_msgno() : Retourne le numéro de séquence de message pour un UID donné....542
6.42.42 imap_num_msg() : Retourne le nombre de message dans la boîte aux lettres courante....542
6.42.43 imap_num_recent() : Retourne le nombre de messages récents dans la boîte aux lettres courante....542
6.42.44 imap_open() : Ouvre un flot IMAP vers une boîte aux lettres....542
6.42.45 imap_ping() : Vérifie que le flot IMAP est toujours actif....544
6.42.46 imap_qprint() : Convertit une chaîne à guillemets en une chaîne à 8 bits....544
6.42.47 imap_renamemailbox() : Renomme une boîte aux lettres....545
6.42.48 imap_reopen() : Ouvre un flot IMAP vers une nouvelle boîte aux lettres....545
6.42.49 imap_rfc822_parse_adrlist() : Analyse une chaîne d'adresse....545
6.42.50 imap_rfc822_parse_headers() : Analyse une en−tête mail....546
6.42.51 imap_rfc822_write_address() : Retourne une adresse email proprement formatée....546
6.42.52 imap_scanmailbox() : Lit la liste des boîtes aux lettres, et y recherche une chaîne....546
6.42.53 imap_search() : Retourne un tableau de message après recherche....546
6.42.54 imap_set_quota() : Modifie le quota d'une boîte aux lettres....547
6.42.55 imap_setflag_full() : Positionne un flag sur un message....548
6.42.56 imap_sort() : Trie des messages....548
6.42.57 imap_status() : Retourne les informations de statut sur une boîte aux lettres autres que la boîte courante....549
6.42.58 imap_subscribe() : Souscrit à une boîte aux lettres....550
6.42.59 imap_uid() : Retourne l'UID d'un message....550
6.42.60 imap_undelete() : Enlève la marque d'effacement d'un message....550
6.42.61 imap_unsubscribe() : Termine la souscription à une boîte aux lettres....550
6.42.62 imap_utf7_decode() : Décode une chaîne modifiée UTF−7....551
6.42.63 imap_utf7_encode() : Convertit des données 8bit en texte UTF−7....551
6.42.64 imap_utf8() : Convertit du texte en UTF8....551
6.43 Informix....552

Sommaire

6 Index des fonctions

6.43.1 Introduction....552
6.43.2 Pré−requis....552
6.43.3 Installation....552
6.43.4 Configuration à l'exécution....552
6.43.5 Types de ressources....555
6.43.6 Constantes Prédefinies....555
6.43.8 ifx_blobinfile_mode() : Choisit le mode par défaut des objets BLOB pour toutes les requêtes SELECT....556
6.43.9 ifx_byteasvarchar() : Choisit le mode par défaut des objets BYTE....556
6.43.10 ifx_close() : Ferme une connexion à un serveur Informix....557
6.43.11 ifx_connect() : Ouvre une connexion à un serveur Informix....557
6.43.12 ifx_copy_blob() : Duplique un objet BLOB....558
6.43.13 ifx_create_blob() : Crée un objet BLOB....558
6.43.14 ifx_create_char() : Crée un objet char....558
6.43.15 ifx_do() : Exécute une requête SQL déjà préparée....558
6.43.16 ifx_error() : Retourne le code d'erreur de la dernière requête Informix....558
6.43.17 ifx_errormsg() : Retourne le message d'erreur de la dernière requête Informix....559
6.43.18 ifx_fetch_row() : Retourne une ligne sous la forme d'un tableau énuméré....559
6.43.19 ifx_fieldproperties() : Liste les propriétés des champs SQL....560
6.43.20 ifx_fieldtypes() : Liste les champs Informix SQL....561
6.43.21 ifx_free_blob() : Supprime un objet BLOB....561
6.43.22 ifx_free_char() : Supprime un objet char....562
6.43.23 ifx_free_result() : Libère les ressources prises par un résultat....562
6.43.24 ifx_get_blob() : Retourne le contenu d'un objet BLOB....562
6.43.25 ifx_get_char() : Retourne le contenu d'un objet char....562
6.43.26 ifx_getsqlca() : Retourne le contenu de la variable sqlca.sqlerrd[0..5] après une requête....562
6.43.27 ifx_htmltbl_result() : Lit toutes les lignes d'un tableau, et la met sous la forme d'un tableau HTML....563
6.43.28 ifx_nullformat() : Modifie le mode par défaut de lecture des valeurs....563
6.43.29 ifx_num_fields() : Retourne le nombre de colonnes dans une requête....564
6.43.30 ifx_num_rows() : Compte le nombre de ligne déjà lues dans un résultat....564
6.43.31 ifx_pconnect() : Ouvre une connexion persistante à un serveur Informix....564
6.43.32 ifx_prepare() : Prépare une requête SQL pour l'exécution....564
6.43.33 ifx_query() : Envoie une requête Informix....565
6.43.34 ifx_textasvarchar() : Choisit le mode par défaut des objets text....566
6.43.35 ifx_update_blob() : Modifie le contenu d'un objet BLOB....567
6.43.36 ifx_update_char() : Modifie le contenu d'un objet char....567
6.43.37 ifxus_close_slob() : Ferme un objet SLOB....567
6.43.38 ifxus_create_slob() : Crée un objet SLOB et l'ouvre....567
6.43.39 ifx_free_slob() : Supprime un objet SLOB....567
6.43.40 ifxus_open_slob() : Ouvre un objet SLOB....568
6.43.41 ifxus_read_slob() : Lit n bytes d'un objet SLOB....568
6.43.42 ifxus_seek_slob() : Fixe le fichier courant, ou la position courante....568
6.43.43 ifxus_tell_slob() : Retourne le fichier courant, ou la position courante....568
6.43.44 ifxus_write_slob() : Ecrit une chaîne dans un objet SLOB....568
6.44 InterBase....569
6.44.1 Introduction....569
6.44.2 Pré−requis....569
6.44.3 Installation....569

Sommaire

6 Index des fonctions

6.44.4 Configuration à l'exécution ... 569
6.44.5 Types de ressources ... 569
6.44.6 Constantes Prédefinies ... 570
6.44.8 ibase_commit() : Valide une transaction ... 570
6.44.9 ibase_connect() : Ouvre une connexion à une base de données Interbase ... 570
6.44.10 ibase_errmsg() : Retourne un message d'erreur ... 571
6.44.11 ibase_execute() : Exécute une requête préparée ... 572
6.44.12 ibase_fetch_object() : Lit une ligne dans une base Interbase dans un objet ... 572
6.44.13 ibase_fetch_row() : Lit une ligne dans une base Interbase ... 572
6.44.14 ibase_field_info() : Lit les informations sur un champs ... 573
6.44.15 ibase_free_query() : Libère la mémoire réservée par une requête préparée ... 573
6.44.16 ibase_free_result() : Libère un résultat ... 573
6.44.17 ibase_num_fields() : Retourne le nombre de colonnes dans un résultat ... 573
6.44.18 ibase_pconnect() : Ouvre une connexion persistante à une base de données Interbase ... 574
6.44.19 ibase_prepare() : Prépare une requête pour lier les paramètres et l'éxécuter ultérieurement ... 574
6.44.20 ibase_query() : Exécute une requête sur une base Interbase ... 575
6.44.21 ibase_rollback() : Annule une transaction ... 575
6.44.22 ibase_timefmt() : Fixe le format de date pour les prochaines requêtes ... 575
6.44.23 ibase_trans() : Prépare une transaction ... 576
6.45 Ingres II ... 577
6.45.1 Introduction ... 577
6.45.2 Pré−requis ... 577
6.45.3 Installation ... 577
6.45.4 Configuration à l'exécution ... 577
6.45.5 Types de ressources ... 578
6.45.6 Constantes Prédefinies ... 578
6.45.8 ingres_close() : Ferme une connexion à un serveur Ingres ... 578
6.45.9 ingres_commit() : Valide une transaction ... 579
6.45.10 ingres_connect() : Ouvre une connexion à un serveur Ingres ... 579
6.45.11 ingres_fetch_array() : Récupère une ligne de résultat dans un tableau ... 580
6.45.12 ingres_fetch_object() : Récupère une ligne de résultat dans un objet ... 581
6.45.13 ingres_fetch_row() : Récupère une ligne de résultat dans un tableau énuméré ... 581
6.45.14 ingres_field_length() : Retourne la taille d'un champ ... 582
6.45.15 ingres_field_name() : Retourne le nom d'un champ dans le résultat d'une requête ... 582
6.45.16 ingres_field_nullable() : Teste si un champ est annulable ... 582
6.45.17 ingres_field_precision() : Retourne la précision d'un champ ... 583
6.45.18 ingres_field_scale() : Retourne l'échelle d'un champ ... 583
6.45.19 ingres_field_type() : Retourne le type d'un champ dans le résultat d'une requête ... 583
6.45.20 ingres_num_fields() : Retourne le nombre de champs renvoyés par la dernière requête ... 584
6.45.21 ingres_num_rows() : Retourne le nombre de lignes affectées ou retournées par la dernière requête ... 584
6.45.22 ingres_pconnect() : Ouvre une connexion persistante à un serveur Ingres ... 584
6.45.23 ingres_query() : Envoie une requête SQL à un serveur Ingres II ... 585

Sommaire

6 Index des fonctions

6.45.24 ingres_rollback() : Annule une transaction....586
6.46 IRC....587
6.46.2 ircg_disconnect() : Ferme la connexion avec un serveur....587
6.46.3 ircg_html_encode() : Prépare l'affichage pour le HTML....587
6.46.4 ircg_ignore_add() : Ajoute un utilisateur sur la liste des utilisateurs indésirables....587
6.46.5 ircg_ignore_del() : Supprime un utilisateur de la liste des utilisateurs indésirables....588
6.46.6 ircg_is_conn_alive() : Vérifie l'état de la connexion....588
6.46.7 ircg_join() : Rejoint un canal IRC....588
6.46.8 ircg_kick() : Expulse un utilisateur d'un canal....588
6.46.9 ircg_lookup_format_messages() : Sélectionne un format d'affichage pour les messages IRC....588
6.46.10 ircg_msg() : Envoie un message à un canal ou un utilisateur....589
6.46.11 ircg_nick() : Change de nom sur le serveur....589
6.46.12 ircg_notice() : Envoie une note (notice) à un utilisateur....589
6.46.13 ircg_part() : Quitte le canal....589
6.46.14 ircg_pconnect() : Connecte à un serveur IRC....589
6.46.15 ircg_register_format_messages() : Enregistre un nouveau format d'affichage des messages IRC....590
6.46.16 ircg_set_current() : Prépare la connexion courante pour l'affichage....591
6.46.17 ircg_topic() : Modifie le sujet (topic) d'un canal....591
6.46.18 ircg_whois() : Requiert les informations sur un utilisateur....591
6.47 Java....592
6.47.1 Introduction....592
6.47.2 Pré−requis....592
6.47.3 Installation....592
6.47.4 Configuration à l'exécution....592
6.47.5 Types de ressources....592
6.47.6 Constantes Prédefinies....593
6.47.7 Exemples....593
6.47.8 Java Servlet SAPI....594
6.47.10 java_last_exception_get() : Lit la dernière exception Java....595
6.48 LDAP....596
6.48.1 Introduction....596
6.48.2 Pré−requis....597
6.48.3 Installation....597
6.48.4 Configuration à l'exécution....597
6.48.5 Types de ressources....597
6.48.6 Constantes Prédefinies....597
6.48.7 Exemple complet de code LDAP....598
6.48.8 Utiliser les fonctions LDAP de PHP....598
6.48.10 ldap_add() : Ajoute une entrée dans un dossier LDAP....600
6.48.11 ldap_bind() : Bind to LDAP directory....600
6.48.12 ldap_close() : Ferme une connexion au serveur LDAP....601
6.48.13 ldap_compare() : Compare une entrée avec des valeurs d'attributs....601
6.48.14 ldap_connect() : Se connecte à un serveur LDAP....602
6.48.15 ldap_count_entries() : Compte le nombre d'entrées après une recherche....602
6.48.16 ldap_delete() : Efface une entrée dans un dossier....603
6.48.17 ldap_dn2ufn() : Convert it un DN en format UFN (User Friendly Naming)....603
6.48.18 ldap_err2str() : Convertit un numéro d'erreur LDAP en message d'erreur....603

Sommaire

6 Index des fonctions

6.48.19 ldap_errno() : Retourne le numéro d'erreur LDAP de la dernière commande exécutée....604
6.48.20 ldap_error() : Retourne le message LDAP de la dernière commande LDAP....604
6.48.21 ldap_explode_dn() : Sépare les différents composants d'un DN....605
6.48.22 ldap_first_attribute() : Return first attribute....605
6.48.23 ldap_first_entry() : Retourne la première entrée....605
6.48.24 ldap_first_reference() : Return first reference....605
6.48.25 ldap_free_result() : Free result memory....606
6.48.26 ldap_get_attributes() : Lit les attributs d'une entrée....606
6.48.27 ldap_get_dn() : Lit le DN d'une entrée....607
6.48.28 ldap_get_entries() : Lit toutes les entrées du résultat....607
6.48.29 ldap_get_option() : Lit/écrit la valeur courante d'une option....608
6.48.30 ldap_get_values_len() : Lit toutes les valeurs binaires d'une entrée....608
6.48.31 ldap_get_values() : Lit toutes les valeurs d'une entrée....609
6.48.32 ldap_list() : Recherche dans un niveau....609
6.48.33 ldap_mod_add() : Ajoute un attribut à l'entrée courante....610
6.48.34 ldap_mod_del() : Efface un attribut à l'entrée courante....610
6.48.35 ldap_mod_replace() : Remplace un attribut dans l'entrée courante....611
6.48.36 ldap_modify() : Modifie une entrée LDAP....611
6.48.37 ldap_next_attribute() : Lit l'attribut suivant....611
6.48.38 ldap_next_entry() : Lit la prochaine entrée....611
6.48.39 ldap_next_reference() : Lit la référence suivante....612
6.48.40 ldap_parse_reference() : Extrait les informations d'une référence d'entrée....612
6.48.41 ldap_parse_result() : Extrait des informations d'un résultat....612
6.48.42 ldap_read() : Lit une entrée....612
6.48.43 ldap_rename() : Modifie le nom d'une entrée....613
6.48.44 ldap_search() : Recherche sur le serveur LDAP....613
6.48.45 ldap_set_option() : Modifie la valeur d'une option LDAP....615
6.48.46 ldap_set_rebind_proc() : Configure une fonction de callback pour refaire des liaisons, lors de recherche de référants....616
6.48.47 ldap_sort() : Trie les entrées d'un résultat LDAP....616
6.48.48 ldap_start_tls() : Démarre TLS....616
6.48.49 ldap_t61_to_8859() : Convertit les caractères t6 en caractères 8859....617
6.48.50 ldap_unbind() : Déconnecte d'un serveur LDAP....617
6.49 Mail....618
6.49.1 Introduction....618
6.49.2 Pré−requis....618
6.49.3 Installation....618
6.49.4 Configuration à l'exécution....618
6.49.5 Types de ressources....619
6.49.6 Constantes Prédefinies....619
6.49.8 mail() : Envoie de mail....619
6.50 Traitement de mail....622
6.50.1 Introduction....622
6.50.3 mailparse_msg_create() : Retourne un analyseur destiné au traitement d'un message....622
6.50.4 mailparse_msg_extract_part_file() : Extrait et décode une section de message....622
6.50.5 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

Sommaire

6 Index des fonctions

sortie standard....623
6.50.6 mailparse_msg_free() : Détruit un analyseur créé par mailparse_msg_crea()....623
6.50.7 mailparse_msg_get_part_data() : Retourne un tableau associatif avec des informations sur le message....623
6.50.8 mailparse_msg_get_part() : Retourne une ressource sur une section d'un message MIME....624
6.50.9 mailparse_msg_get_structure() : Retourne un tableau avec les différentes sections MIME du message....624
6.50.10 mailparse_msg_parse_file() : Analyse un fichier et retourne une ressource représentant la structure du fichier....624
6.50.11 mailparse_msg_parse() : Analyse incrémentalement des données dans un buffer....625
6.50.12 mailparse_rfc822_parse_addresses() : Analyse les adresses et retourne un tableau contenant ces données....625
6.50.13 mailparse_stream_encode() : Lit les données dans un fichier, applique l'encodage et envoie le résultat à destfp....625
6.50.14 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....625
6.51 Mathématiques....627
6.51.1 Introduction....627
6.51.2 Pré−requis....627
6.51.3 Installation....627
6.51.4 Configuration à l'exécution....627
6.51.5 Types de ressources....627
6.51.6 Constantes Prédefinies....627
6.51.8 Acos() : arc cosinus....629
6.51.9 acosh() : Arc cosinus hyperbolique....629
6.51.10 Asin() : arc sinus....629
6.51.11 asinh() : Arc sinus hyperbolique....629
6.51.12 Atan() : arc tangent....630
6.51.13 Atan2() : arc tangent de deux variables....630
6.51.14 atanh() : Arc tangeant hyperbolique....630
6.51.15 base_convert() : Convertit un nombre entre des bases arbitraires....630
6.51.16 BinDec() : Convertit de binaire en décimal....631
6.51.17 Ceil() : Arrondit au nombre supérieur....631
6.51.18 Cos() : cosinus....631
6.51.19 cosh() : Cosinus hyperbolic....632
6.51.20 DecBin() : Convertit de décimal en binaire....632
6.51.21 DecHex() : Convertit de décimal en hexadécimal....632
6.51.22 DecOct() : Convertit de décimal en octal....632
6.51.23 deg2rad() : Convertit un nombre de degrés en radians....632
6.51.24 Exp() : exponentielle....633
6.51.25 expm1() : Retourne le résultat de exp(number) − 1, calculé de manière précise, même si val est proche de 0....633
6.51.26 Floor() : Arrondi à l'entier inférieur....633
6.51.27 fmod() : Retourne le reste de la division....633
6.51.28 getrandmax() : Plus grande valeur aléatoire possible....634
6.51.29 hexdec() : Convertit de hexadécimal en décimal....634
6.51.30 hypot() : Retourne le résultat de sqrt(num1*num1 + num2*num2)....634

Sommaire

6 Index des fonctions

6.51.31 is_finite() : Indique si un nombre est fini....635
6.51.32 is_infinite() :....635
6.51.33 is_nan() : Indique si une valeur est un nombre....635
6.51.34 lcg_value() : Générateur de congruence combinée linéaire....635
6.51.35 Log() : Logarithme naturel (népérien)....635
6.51.36 Log10() : logarithme en base 10....636
6.51.37 log1p() : Retourne le résultat de log(1 + number), calculé de manière précise, même si val est proche de zéro....636
6.51.38 max() : La plus grande valeur....636
6.51.39 min() : La plus petite valeur....636
6.51.40 mt_getrandmax() : La plus grand valeur aléatoire possible....637
6.51.41 mt_rand() : Génère une meilleure valeur aléatoire....637
6.51.42 mt_srand() : Initialise une meilleure valeur aléatoire....637
6.51.43 OctDec() : Convertit d'octal en décimal....638
6.51.44 pi() : Retourne la valeur de pi....638
6.51.45 pow() : Puissance....638
6.51.46 rad2deg() : Convertit de radians en degrés....639
6.51.47 rand() : Génère une valeur aléatoire....639
6.51.48 round() : Arrondi....639
6.51.49 Sin() : Sinus....640
6.51.50 sinh() : Sinyus hyperbolique....640
6.51.51 Sqrt() : Racine carrée....640
6.51.52 srand() : Initialise le générateur de nombres aléatoires....640
6.51.53 Tan() : Tangente....641
6.51.54 tanh() : Tangente hyperbolique....641
6.52 Chaînes de caractères multi−octets....642
6.52.1 Introduction....642
6.52.2 Installation....643
6.52.3 Configuration à l'exécution....643
6.52.4 Types de ressources....646
6.52.5 Constantes Prédefinies....646
6.52.6 Entrées/Sorties HTTP....646
6.52.7 Jeux de caractères supportés....647
6.52.8 Exploitation des chaînes multi−octets en PHP....648
6.52.9 Cas des caractères japonais....648
6.52.10 Références....649
6.52.12 mb_convert_kana() : Convertit entre les différents "kana"....650
6.52.13 mb_convert_variables() : Convertit l'encodage de variables....650
6.52.14 mb_decode_mimeheader() : Décode une en−tête MIME....651
6.52.15 mb_decode_numericentity() : Décode les entités HTML en caractères....651
6.52.16 mb_detect_encoding() : Détecte un encodage....652
6.52.17 mb_detect_order() : Lit/modifie l'ordre de détection des encodages....652
6.52.18 mb_encode_mimeheader() : Encode une chaîne pour une en−tête MIME....653
6.52.19 mb_encode_numericentity() : Encode des entités HTML....653
6.52.20 mb_http_input() : Détecte le type d'encodage d'un caractère HTTP....654
6.52.21 mb_http_output() : Lit/modifie l'encodage d'affichage....655
6.52.22 mb_internal_encoding() : Lit/modifie l'encodage interne....655
6.52.23 mb_language() : Lit/modifie le langage courant....655
6.52.24 mb_output_handler() : Fonction de traitement des affichages web....656
6.52.25 mb_parse_str() : Analyse les données HTTP GET/POST/COOKIE et assigne les variables globales....656

Sommaire

6 Index des fonctions

6.52.26 mb_preferred_mime_name() : Détecte l'encodage MIME....657
6.52.27 mb_send_mail() : Envoie un mail encodé ISO−2022−JP (mail japonais)....657
6.52.28 mb_strcut() : Coupe une partie de chaîne....658
6.52.29 mb_strimwidth() : Tronque une chaîne....658
6.52.30 mb_strlen() : Retourne la taille d'une chaîne....658
6.52.31 mb_strpos() : Repère la première occurence d'un caractère dans une chaîne....659
6.52.32 mb_strrpos() : Repère la dernière occurence d'un caractère dans une chaîne....659
6.52.33 mb_strwidth() : Retourne la largeur d'une chaîne....660
6.52.34 mb_substitute_character() : Lit/modifie les caractères de substitution....660
6.52.35 mb_substr() : Lit une sous−chaîne....661
6.53 MCAL....662
6.53.1 Introduction....662
6.53.2 Pré−requis....662
6.53.3 Installation....662
6.53.4 Configuration à l'exécution....662
6.53.5 Types de ressources....662
6.53.6 Constantes Prédefinies....663
6.53.8 mcal_close() : Ferme une connexion MCAL....664
6.53.9 mcal_create_calendar() : Crée un nouveau calendrier....664
6.53.10 mcal_date_compare() : Compare deux dates....665
6.53.11 mcal_date_valid() : Valide une date....665
6.53.12 mcal_day_of_week() : Le jour de la semaine....665
6.53.13 mcal_day_of_year() : Le jour de l'année....665
6.53.14 mcal_days_in_month() : Retourne le nombre de jour d'un mois....665
6.53.15 mcal_delete_calendar() : Efface un calendrier....665
6.53.16 mcal_delete_event() : Efface un événement dans un calendrier MCAL....666
6.53.17 mcal_event_add_attribute() : Ajoute un attribut et une valeur à la structure globale....666
6.53.18 mcal_event_init() : Initialise la structure globale d'un flot....666
6.53.19 mcal_event_set_alarm() : Fixe l'alarme de la structure globale....666
6.53.20 mcal_event_set_category() : Fixe la catégorie de la structure globale....666
6.53.21 mcal_event_set_class() : Fixe la classe de la structure globale....667
6.53.22 mcal_event_set_description() : Fixe la description de la structure globale....667
6.53.23 mcal_event_set_end() : Fixe la date de fin de la structure globale....667
6.53.24 mcal_event_set_recur_daily() : Fixe la récurrence quotidienne....667
6.53.25 mcal_event_set_recur_monthly_mday() : Fixe la récurrence....667
6.53.26 mcal_event_set_recur_monthly_wday() : Fixe la récurrence mensuelle....668
6.53.27 mcal_event_set_recur_none() : Supprime la récurrence de la structure globale....668
6.53.28 mcal_event_set_recur_weekly() : Fixe la récurrence hebdomadaire....668
6.53.29 mcal_event_set_recur_yearly() : Fixe la récurrence annuelle....668
6.53.30 mcal_event_set_start() : Fixe les dates de début et de fin de la structure globale....668
6.53.31 mcal_event_set_title() : Fixe le titre de la structure globale....669
6.53.32 mcal_expunge() : Supprime tous les événements marqués pour l'effacement....669
6.53.33 mcal_fetch_current_stream_event() : Retourne un objet contenant la structure de date pour le flot courant....669
6.53.34 mcal_fetch_event() : Recherche un événement dans le calendrier....670

Sommaire

6 Index des fonctions

6.53.35 mcal_is_leap_year() : Vérifie que l'année est bissextile....671
6.53.36 mcal_list_alarms() : Retourne une liste d'événements qui ont une alarme prévue à une date....671
6.53.37 mcal_list_events() : Retourne une liste d'événement entre deux dates....671
6.53.38 mcal_next_recurrence() : Retourne la prochaine occurrence d'un événement....671
6.53.39 mcal_open() : Ouvre une connexion MCAL....672
6.53.40 mcal_popen() : Ouvre une connexion persistante MCAL....672
6.53.41 mcal_rename_calendar() : Renomme un calendrier....672
6.53.42 mcal_reopen() : Réouvre une connexion MCAL....672
6.53.43 mcal_snooze() : Eteint l'alarme d'un événement....672
6.53.44 mcal_store_event() : Modifie un événement dans un calendrier MCAL....673
6.53.45 mcal_time_valid() : Valide une heure....673
6.53.46 mcal_week_of_year() : Retourne le numéro de semaine dans l'année pour une date donnée....673
6.54 Chiffrage mcrypt....674
6.54.1 Introduction....674
6.54.2 Pré−requis....674
6.54.3 Installation....674
6.54.4 Configuration à l'exécution....674
6.54.5 Types de ressources....674
6.54.6 Constantes Prédefinies....674
6.54.7 Mcrypt ciphers....675
6.54.8 Exemples....676
6.54.10 mcrypt_cfb() : Chiffre/déchiffre des données en mode CFB....678
6.54.11 mcrypt_create_iv() : Crée un vecteur d'initialisation à partir d'une source aléatoire....678
6.54.12 mcrypt_decrypt() : Déchiffre un texte....679
6.54.13 mcrypt_ecb() : Chiffre/déchiffre des données en mode ECB....679
6.54.14 mcrypt_enc_get_algorithms_name() : Retourne le nom de l'algorithme....680
6.54.15 mcrypt_enc_get_block_size() : Retourne la taille de blocs d'un algorithme....680
6.54.16 mcrypt_enc_get_iv_size() : Retourne la taille du VI d'un algorithme....680
6.54.17 mcrypt_enc_get_key_size() : Retourne la taille maximale de la clé pour un mode....680
6.54.18 mcrypt_enc_get_modes_name() : Retourne le nom du mode....680
6.54.19 mcrypt_enc_get_supported_key_sizes() : Retourne un tableau contenant les tailles de clés acceptées par un algorithme....680
6.54.20 mcrypt_enc_is_block_algorithm_mode() : Teste le chiffrage par blocs d'un mode....681
6.54.21 mcrypt_enc_is_block_algorithm() : Teste le chiffrage par blocs d'un algorithme....681
6.54.22 mcrypt_enc_is_block_mode() : Teste si le mode retourne les données par blocs....681
6.54.23 mcrypt_enc_self_test() : Teste un module ouvert....681
6.54.24 mcrypt_encrypt() : Chiffre un texte....681
6.54.25 mcrypt_generic_deinit() : Prépare le module pour le déchargement....682
6.54.26 mcrypt_generic_end() : Termine un chiffrage....682
6.54.27 mcrypt_generic_init() : Initialise tous les buffers nécessaires....683
6.54.28 mcrypt_generic() : Chiffre....683
6.54.29 mcrypt_get_block_size() : Retourne la taille de blocs d'un chiffrement....683
6.54.30 mcrypt_get_cipher_name() : Lit le nom du chiffrement utilisé....683

Sommaire

6 Index des fonctions

6.54.31 mcrypt_get_iv_size() : Retourne la taille du VI utilisé par un couple chiffrement/mode....684
6.54.32 mcrypt_get_key_size() : Retourne la taille de la clé d'un chiffrement....684
6.54.33 mcrypt_list_algorithms() : Liste tous les algorithmes de chiffrement supportés....685
6.54.34 mcrypt_list_modes() : Liste tous les modes de chiffrement supportés....685
6.54.35 mcrypt_module_close() : Décharge le module de chiffrement....685
6.54.36 mcrypt_module_get_algo_block_size() : Retourne la taille de blocs d'un algorithme....686
6.54.37 mcrypt_module_get_algo_key_size() : Retourne la taille maximale de clé....686
6.54.38 mcrypt_module_get_supported_key_sizes() : Retourne un tableau contenant les tailles de clés supportée par un algorithme de chiffrement....686
6.54.39 mcrypt_module_is_block_algorithm_mode() : Indique si un mode fonctionne par blocs....686
6.54.40 mcrypt_module_is_block_algorithm() : Indique si un algorithme fonctionne par blocs....687
6.54.41 mcrypt_module_is_block_mode() : Indique si un mode travaille par blocs....687
6.54.42 mcrypt_module_open() : Ouvre le module de l'algorithme et le mode à utiliser....687
6.54.43 mcrypt_module_self_test() : Teste un mode....688
6.54.44 mcrypt_ofb() : Chiffre/déchiffre des données en mode OFB....688
6.54.45 mdecrypt_generic() : Déchiffre....688
6.55 MCVE (Paiement)....690
6.55.1 Introduction....690
6.55.2 Installation....690
6.55.3 Voir aussi....690
6.55.5 mcve_adduserarg() : Ajoute une valeur dans une structure de configuration d'utilisateur....692
6.55.6 mcve_bt() : Lit le total des paiements non finalisés en mode batch....692
6.55.7 mcve_checkstatus() : Vérifie si une transaction a été complétée....692
6.55.8 mcve_chkpwd() : Verifie le mot de passe....692
6.55.9 mcve_chngpwd() : Modifie le mot de passe administrateur....693
6.55.10 mcve_completeauthorizations() : Retourne le nombre de transactions dans la queue d'autorisation, ainsi que la liste des identifiants....693
6.55.11 mcve_connect() : Etablit la connexion avec le système MCVE....693
6.55.12 mcve_connectionerror() : Lit un message représentant la raison de l'échec de la connexion....693
6.55.13 mcve_deleteresponse() : Efface une transaction spécifiée dans une structure MCVE_CONN....693
6.55.14 mcve_deletetrans() : Efface une transaction spécifiée dans une structure MCVE_CONN....694
6.55.15 mcve_deleteusersetup() : Supprime une structure usersetup....694
6.55.16 mcve_deluser() : Efface un compte d'utilisateur MCVE....694
6.55.17 mcve_destroyconn() : Termine une connexion MCVE_CONN....694
6.55.18 mcve_destroyengine() : Libère la mémoire associée à la connectivité IP/SSL....694
6.55.19 mcve_disableuser() : Inactive un utilisateur MCVE actif....695
6.55.20 mcve_edituser() : Modifie la structure usersetup d'un utilisateur....695
6.55.21 mcve_enableuser() : Active un utilisateur MCVE inactif....695
6.55.22 mcve_force() : Envoie une commande FORCE à MCVE. (généralement une autorisation par téléphone)....695

Sommaire

6 Index des fonctions

6.55.23 mcve_getcell() : Lit une cellule spécifique dans une réponse au format CSV, en utilisant le nom de colonne....695
6.55.24 mcve_getcellbynum() : Lit une cellule spécifique dans une réponse au format CSV, en utilisant le numéro de colonne....696
6.55.25 mcve_getcommadelimited() : Lit une ligne brute, au format CSV retournée par MCVE....696
6.55.26 mcve_getheader() : Lit le nom d'une colonne dans une réponse au format CSV....696
6.55.27 mcve_getuserarg() : Lit une valeur dans une structure usersetup....696
6.55.28 mcve_getuserparam() : Lit un paramètre de réponse utilisateur....696
6.55.29 mcve_gft() : Audite MCVE pour les transactions échouées....697
6.55.30 mcve_gl() : Audite MCVE pour les transactions finalisées....697
6.55.31 mcve_gut() : Audite MCVE pour les transactions non conclues....697
6.55.32 mcve_initconn() : Crée et initialise une structure MCVE_CONN....697
6.55.33 mcve_initengine() : Prépare le client pour la connexion IP/SSL....697
6.55.34 mcve_initusersetup() : Initialise une structure pour stocker des données utilisateurs....698
6.55.35 mcve_iscommadelimited() : Vérifie qu'une réponse est au format MVCE....698
6.55.36 mcve_liststats() : Liste les statistiques de tous les utilisateurs MCVE....698
6.55.37 mcve_listusers() : Liste de tous les utilisateurs du système MCVE....698
6.55.38 mcve_maxconntimeout() : La durée maximum de tentative de connexion à MCVE....698
6.55.39 mcve_monitor() : Communique avec MCVE (envoie et reçoit des données) sans blocage....699
6.55.40 mcve_numcolumns() : Nombre de colonnes retournées dans une ligne au format CSV....699
6.55.41 mcve_numrows() : Nombre de lignes retournées dans une réponse au format CSV....699
6.55.42 mcve_override() : Envoie une commande OVERRIDE à MCVE....699
6.55.43 mcve_parsecommadelimited() : Analyse une au format CSV pour que mcve_getcell et consorts fonctionnent....699
6.55.44 mcve_ping() : Envoie une requête de ping à MCVE....700
6.55.45 mcve_preauth() : Envoie une commande PREAUTHORIZATION à MCVE....700
6.55.46 mcve_preauthcompletion() : Complète une PREAUTHORIZATION et la prépare pour la finalisation....700
6.55.47 mcve_qc() : Audite MCVE pour une liste de transactions dans la queue de sortie....700
6.55.48 mcve_responseparam() : Lit un paramètre de réponse personnalisé....700
6.55.49 mcve_return() : Emet une commande RETURN ou CREDIT vers MCVE....701
6.55.50 mcve_returncode() : Lit le code retourné exact de la transaction....701
6.55.51 mcve_returnstatus() : Vérifie si la transaction a réussi....701
6.55.52 mcve_sale() : Envoie une commande de vente (SALE) à MCVE....701
6.55.53 mcve_setblocking() : Configure la connexion en mode bloquant/non bloquant....701
6.55.54 mcve_setdropfile() : Configure la connexion en mode Drop−File....702
6.55.55 mcve_setip() : Configure la connexion en mode IP....702
6.55.56 mcve_setssl() : Configure la connexion en mode SSL....702
6.55.57 mcve_settimeout() : Configure le temps maximum d'une transaction (par transaction)....702
6.55.58 mcve_settle() : Emet une résolution pour faire un dépôt batch....702
6.55.59 mcve_text_avs() : Lit une représentation textuelle de return_avs....703

Sommaire

6 Index des fonctions

6.55.60 mcve_text_code() : Lit une représentation textuelle de eturn_code....703
6.55.61 mcve_text_cv() : Lit une représentation textuelle de return_cv....703
6.55.62 mcve_transactionauth() : Lit le numéro d'autorisation de la transaction (alpha−numérique)....703
6.55.63 mcve_transactionavs() : Lit l'adresse de vérification (Address Verification) dans le statut retourné....703
6.55.64 mcve_transactionbatch() : Lit le numéro de batch associé à la transaction....704
6.55.65 mcve_transactioncv() : Lit le statut de CVC2/CVV2/CID....704
6.55.66 mcve_transactionid() : Lit l'identifiant système unique d'une transaction....704
6.55.67 mcve_transactionitem() : Lit le numéro d'ITEM d'une transaction dans un batch....704
6.55.68 mcve_transactionssent() : Vérifie si le buffer de sortie est vide ou pas....704
6.55.69 mcve_transactiontext() : Lit le message retourné par MCVE ou par l'institution de traitement....705
6.55.70 mcve_transinqueue() : Nombre de transactions dans la queue client....705
6.55.71 mcve_transnew() : Démarre une nouvelle transaction....705
6.55.72 mcve_transparam() : Ajoute un paramètre à une transaction....705
6.55.73 mcve_transsend() : Finalise et envoie une transaction....705
6.55.74 mcve_ub() : Lit une liste des paiements en attente en mode batch....706
6.55.75 mcve_uwait() : Attend x microsecondes....706
6.55.76 mcve_verifyconnection() : Configure MCVE pour qu'il pingue ou pas la connexion, afin d'en vérifier l'existence....706
6.55.77 mcve_verifysslcert() : Configure MCVE pour qu'il vérifie ou pas le certificat SSL....706
6.55.78 mcve_void() : Annule (rend VOID) une transaction dans la queue d'attente....706
6.56 Hash....707
6.56.1 Introduction....707
6.56.2 Pré−requis....707
6.56.3 Installation....707
6.56.4 Configuration à l'exécution....707
6.56.5 Types de ressources....707
6.56.6 Constantes Prédefinies....707
6.56.7 Exemples....708
6.56.9 mhash_get_block_size() : Retourne la taille de bloc du hash....708
6.56.10 mhash_get_hash_name() : Retourne le nom du hash....708
6.56.11 mhash_keygen_s2k() : Génère une clé....709
6.56.12 mhash() : Calcule un hash....709
6.57 Mimetype....710
6.57.1 Introduction....710
6.57.2 Pré−requis....710
6.57.3 Installation....710
6.57.4 Configuration à l'exécution....710
6.57.5 Types de ressources....710
6.57.6 Constantes Prédefinies....710
6.58 Microsoft SQL Server....711
6.58.1 Introduction....711
6.58.2 Pré−requis....711
6.58.3 Installation....711
6.58.4 Configuration à l'exécution....711
6.58.5 Types de ressources....712

Sommaire

6 Index des fonctions

6.58.6 Constantes Prédefinies 712
6.58.8 mssql_connect() : Ouvre une connexion à un serveur MS SQL server 712
6.58.9 mssql_data_seek() : Déplace le pointeur interne de ligne 713
6.58.10 mssql_fetch_array() : Lit une ligne dans un tableau 713
6.58.11 mssql_fetch_field() : Lit les inforamtions sur le champs 713
6.58.12 mssql_fetch_object() : Retourne une ligne sous la forme d'un objet 714
6.58.13 mssql_fetch_row() : Lit une ligne comme un tableau 714
6.58.14 mssql_field_length() : Lit la longueur d'un champs 715
6.58.15 mssql_field_name() : Lit le nom d'un champs 715
6.58.16 mssql_field_seek() : Fixe l'offset du pointeur de champs 715
6.58.17 mssql_field_type() : Lit le nom d'un champs 715
6.58.18 mssql_free_result() : Libère la mémoire 715
6.58.19 mssql_get_last_message() : Retourne le dernier message d'erreur du serveur (min_message_severity?) 715
6.58.20 mssql_min_error_severity() : Fixe le niveau de sévérité des erreurs 715
6.58.21 mssql_min_message_severity() : Fixe le niveau de sévérité des messages d'erreurs 716
6.58.22 fbsql_next_result() : Déplace le pointeur interne vers le résultat suivant 716
6.58.23 mssql_num_fields() : Retourne le nombre de champs dans un résultat 716
6.58.24 mssql_num_rows() : Retourne le nombre de lignes dans un résultat 716
6.58.25 mssql_pconnect() : Ouvre une connexion persistante à un serveur MS SQL 717
6.58.26 mssql_query() : Envoie une requête SQL 717
6.58.27 mssql_result() : Lit les données d'un résultat 717
6.58.28 mssql_select_db() : Sélectionne la base de données MS SQL 718
6.59 Ming pour Flash 719
6.59.1 Introduction 719
6.59.2 Pré−requis 719
6.59.3 Installation 719
6.59.4 Configuration à l'exécution 720
6.59.5 Types de ressources 720
6.59.6 Constantes Prédefinies 720
6.59.7 Classes Prédefinies 721
6.59.9 SWFBitmap−>getHeight() : Retourne la hauteur d'une bitmap 723
6.59.10 SWFBitmap−>getWidth() : Retourne la largeur d'une bitmap 723
6.59.11 SWFBitmap() : Crée un objet bitmap 723
6.59.12 SWFbutton−>addAction() : Ajoute une action au bouton 725
6.59.13 SWFbutton−>addShape() : Ajoute une forme à un bouton 725
6.59.14 SWFbutton−>setAction() : Assigne l'action du bouton 725
6.59.15 SWFbutton−>setDown() : Alias de addShape(shape, SWFBUTTON_DOWN) 725
6.59.16 SWFbutton−>setHit() : Alias de addShape(shape, SWFBUTTON_HIT) 726
6.59.17 SWFbutton−>setOver() : Alias de addShape(shape, SWFBUTTON_OVER) 726
6.59.18 SWFbutton−>setUp() : Alias de SWFbutton−>addShape(shape, SWFBUTTON_UP) 726
6.59.19 SWFbutton() : Crée un nouveau bouton 726
6.59.20 SWFDisplayItem−>addColor() : Ajoute une couleur à une transoformation 728
6.59.21 SWFDisplayItem−>move() : Déplace un objet en coordonnées relatives 728
6.59.22 SWFDisplayItem−>moveTo() : Déplace un objet en coordonnées globales 729
6.59.23 SWFDisplayItem−>multColor() : Multiplie la couleur de transformation 729

Sommaire

6 Index des fonctions

6.59.24 SWFDisplayItem−>remove() : Supprime un objet d'une animation....730
6.59.25 SWFDisplayItem−>Rotate() : Fait tourner une forme relativement....730
6.59.26 SWFDisplayItem−>rotateTo() : Tourne un objet en angle absolu....730
6.59.27 SWFDisplayItem−>scale() : Etire un objet relativement....731
6.59.28 SWFDisplayItem−>scaleTo() : Etire un objet en coordonnées globales....732
6.59.29 SWFDisplayItem−>setDepth() : Modifie la place en profondeur (z−order)....732
6.59.30 SWFDisplayItem−>setName() : Nomme un objet....732
6.59.31 SWFDisplayItem−>setRatio() : Modifie le ratio de l'objet....732
6.59.32 SWFDisplayItem−>skewX() : Incline suivant les X relativement....733
6.59.33 SWFDisplayItem−>skewXTo() : Incline suivant les X....734
6.59.34 SWFDisplayItem−>skewY() : Incline suivant les Y relativement....734
6.59.35 SWFDisplayItem−>skewYTo() : Incline suivant les Y....734
6.59.36 SWFDisplayItem() : Crée un nouvel objet d'affichage displayitem....734
6.59.37 SWFFill−>moveTo() : Déplace l'origine de l'objet SWFFill....735
6.59.38 SWFFill−>rotateTo() : Tourne la forme....735
6.59.39 SWFFill−>scaleTo() : Modifie l'échelle de la forme....735
6.59.40 SWFFill−>skewXTo() : Incline (abscisses)....735
6.59.41 SWFFill−>skewYTo() : Incline (ordonnées)....736
6.59.42 SWFFill() : Crée un objet de remplissage....736
6.59.43 swffont−>getwidth() : Retourne la taille de la chaîne....736
6.59.44 SWFFont() : Charge une police....736
6.59.45 SWFGradient−>addEntry() : Ajoute une couleur à la liste du gradient....737
6.59.46 SWFGradient() : Crée un objet gradient....737
6.59.47 SWFMorph−>getshape1() : Sélectionne la forme de départ....738
6.59.48 SWFMorph−>getshape2() : Sélectionne la forme de fin....738
6.59.49 SWFMorph() : Crée un morphing....738
6.59.50 SWFMovie−>add() : Ajoute un objet dans une animation....739
6.59.51 SWFMovie−>nextframe() : Passe à l'image suivante....740
6.59.52 SWFMovie−>output() : Envoie votre animation au navigateur....740
6.59.53 SWFMovie−>remove() : Supprime un objet d'une animation....740
6.59.54 SWFMovie−>save() : Sauve dans un fichier....740
6.59.55 SWFMovie−>setbackground() : Modifie la couleur de fond....741
6.59.56 SWFMovie−>setdimension() : Modifie les dimensions de l'animation....741
6.59.57 SWFMovie−>setframes() : Modifie le nombre total d'images dans l'animation....741
6.59.58 SWFMovie−>setrate() : Modifie la vitesse de l'animation....741
6.59.59 SWFMovie−>streammp3() : Envoie un fichier MP3 en streaming....741
6.59.60 SWFMovie() : Crée un objet 'animation'....742
6.59.61 SWFShape−>addFill() : Ajoute un remplissage plein à la forme....742
6.59.62 SWFShape−>drawCurve() : Dessine une courbe relativement....743
6.59.63 SWFShape−>drawCurveTo() : Dessine une courbe....744
6.59.64 SWFShape−>drawLine() : Dessine une ligne relativement....744
6.59.65 SWFShape−>drawLineTo() : Dessine une ligne....744
6.59.66 SWFShape−>movePen() : Déplace le stylo relativement....745
6.59.67 SWFShape−>movePenTo() : Déplace le stylo....745
6.59.68 SWFShape−>setLeftFill() : Modifie la couleur de rastérisation de gauche....745
6.59.69 SWFShape−>setLine() : Modifie le style de ligne de la forme....745
6.59.70 SWFShape−>setRightFill() : Modifie la couleur de rastériation de droite....747
6.59.71 SWFShape() : Crée une nouvelle forme....747
6.59.72 SWFSprite−>add() : Ajoute un objet à un sprite....747
6.59.73 SWFSprite−>nextframe() : Va à la prochaine image du sprite....748

Sommaire

6 Index des fonctions

6.59.74 SWFSprite−>remove() : Supprime un objet dans un sprite....748
6.59.75 SWFSprite−>setframes() : Fixe le nombre maximum d'image dans le sprite....748
6.59.76 SWFSprite() : Crée un sprite....748
6.59.77 SWFText−>addString() : Ajoute du texte....749
6.59.78 SWFText−>getWidth() : Calcule la longueur d'une chaîne....749
6.59.79 SWFText−>moveTo() : Déplace le stylo de texte....749
6.59.80 SWFText−>setColor() : Modifie la couleur de la police....749
6.59.81 SWFText−>setFont() : Sélectionne la police courante....750
6.59.82 SWFText−>setHeight() : Modifie la hauteur de la police courante....750
6.59.83 SWFText−>setSpacing() : Modifie l'espacement de police....750
6.59.84 SWFText() : Crée un nouvel objet texte....750
6.59.85 SWFTextField−>addstring() : Ajoute au texte....751
6.59.86 SWFTextField−>align() : Modifie l'alignement du texte....751
6.59.87 SWFTextField−>setbounds() : Sélectionne la largeur et hauteur du champs....751
6.59.88 SWFTextField−>setcolor() : Modifie la couleur du champs texte....751
6.59.89 SWFTextField−>setFont() : Modifie la police du champs....751
6.59.90 SWFTextField−>setHeight() : Modifie la hauteur de la police du champs texte....752
6.59.91 SWFTextField−>setindentation() : Modifie l'indentation de la première ligne....752
6.59.92 SWFTextField−>setLeftMargin() : Modifie la marge de gauche....752
6.59.93 SWFTextField−>setLineSpacing() : Modifie l'espacement de lignes....752
6.59.94 SWFTextField−>setMargins() : Modifie les marges du champs texte....752
6.59.95 SWFTextField−>setname() : Nomme le champs texte....753
6.59.96 SWFTextField−>setrightMargin() : Modifie la marge de droite....753
6.59.97 SWFTextField() : Crée un nouveau champs texte....753
6.60 Fonctions diverses....754
6.60.1 Introduction....754
6.60.2 Pré−requis....754
6.60.3 Installation....754
6.60.4 Configuration à l'exécution....754
6.60.5 Types de ressources....755
6.60.6 Constantes Prédefinies....755
6.60.8 connection_status() : Retourne les bits de status de la connexion....755
6.60.9 connection_timeout() : Indique si le script a expiré....755
6.60.10 constant() : Retourne la valeur d'une constante....756
6.60.11 define() : Définit une constante....756
6.60.12 defined() : Vérifie qu'une constante existe....757
6.60.13 die() : Alias de la fonction exit....757
6.60.14 eval() : Evalue une chaîne comme un script PHP....757
6.60.15 exit() : Termine le script courant....758
6.60.16 get_browser() : Indique de quoi est capable le navigateur client....758
6.60.17 highlight_file() : Colorisation de la syntaxe d'un fichier....760
6.60.18 highlight_string() : Applique la syntaxe colorisée à du code PHP....760
6.60.19 ignore_user_abort() : Active l'option décidant si, lors de la déconnexion du client, le script doit poursuivre son exécution ou non....760
6.60.20 leak() : Fuite de mémoire....760
6.60.21 pack() : Compacte des données dans une chaîne binaire....761
6.60.22 show_source() : Colorisation de la syntaxe d'un fichier....762

Sommaire

6 Index des fonctions

6.60.23 sleep() : Retarde l'exécution....762
6.60.24 uniqid() : Génère un identifiant unique....762
6.60.25 unpack() : Déconditionne des données depuis une chaîne binaire....763
6.60.26 usleep() : Retarde l'exécution en micro−secondes....763
6.61 mnoGoSearch....764
6.61.1 Introduction....764
6.61.2 Pré−requis....764
6.61.3 Installation....764
6.61.4 Configuration à l'exécution....764
6.61.5 Types de ressources....764
6.61.6 Constantes Prédefinies....765
6.61.8 Udm_Alloc_Agent() : Alloue une session mnoGoSearch....767
6.61.9 udm_api_version() : Lit la version des API mnoGoSearch....768
6.61.10 udm_cat_list() : Liste toutes les catégories soeurs d'une catégorie....768
6.61.11 udm_cat_path() : Lit le chemin de la catégorie courante....769
6.61.12 udm_clear_search_limits() : Annule toutes les limitations de recherche....769
6.61.13 Udm_Errno() : Numéro d'erreur mnoGoSearch....770
6.61.14 Udm_Error() : Message d'erreur mnoGoSearch....770
6.61.15 Udm_Find() : Effectue une recherche....770
6.61.16 Udm_Free_Agent() : Détruit une session mnoGoSearch....771
6.61.17 udm_free_ispell_data() : Libère la mémoire allouée pour ispell....771
6.61.18 Udm_Free_Res() : Libère un résultat mnoGoSearch....771
6.61.19 udm_get_doc_count() : Lit le nombre total de documents dans les bases....771
6.61.20 Udm_Get_Res_Field() : Lit un champs de résultat mnoGoSearch....772
6.61.21 Udm_Get_Res_Param() : Lit les paramètres de résultats mnoGoSearch....772
6.61.22 udm_load_ispell_data() : Charge les données ispell....773
6.61.23 udm_set_agent_param() : Modifie les paramètres de l'agent mnoGoSearch....775
6.62 mSQL....778
6.62.1 Introduction....778
6.62.2 Pré−requis....778
6.62.3 Installation....778
6.62.4 Configuration à l'exécution....778
6.62.5 Types de ressources....778
6.62.6 Constantes Prédefinies....778
6.62.8 msql_close() : Ferme une connexion mSQL....779
6.62.9 msql_connect() : Ouvre une connexion mSQL....780
6.62.10 msql_create_db() : Crée une base de données mSQL....780
6.62.11 msql_createdb() : Crée une base de données mSQL....780
6.62.12 msql_data_seek() : Déplace le pointeur interne....781
6.62.13 msql_dbname() : Lit le nom de la base de données courante....781
6.62.14 msql_drop_db() : Efface une base de données mSQL....781
6.62.15 msql_dropdb() : Efface une base de données mSQL....781
6.62.16 msql_error() : Retourne le message d'erreur....781
6.62.17 msql_fetch_array() : Lit une ligne sous la forme d'un tableau....782
6.62.18 msql_fetch_field() : Lit la valeur d'un champs....782
6.62.19 msql_fetch_object() : Lit une ligne sous la forme d'un objet....782
6.62.20 msql_fetch_row() : Retourne une ligne sous la forme d'un objet....783
6.62.21 msql_field_seek() : Fixe d'offset d'un champs....783
6.62.22 msql_fieldflags() : Retourne le flag d'un champs....783
6.62.23 msql_fieldlen() : Retourne la longueur d'un champs....784

Sommaire

6 Index des fonctions

6.62.24 msql_fieldname() : Lit le nom d'un champs....784
6.62.25 msql_fieldtable() : Retourne le nom d'une table à partir d'un nom de champs....784
6.62.26 msql_fieldtype() : Retourne le type de champs....784
6.62.27 msql_free_result() : Libère le résultat de la mémoire....784
6.62.28 msql_freeresult() : Libère le résultat de la mémoire....784
6.62.29 msql_list_dbs() : Liste les bases de données mSQL sur un serveur....785
6.62.30 msql_list_fields() : Liste les champs d'une table....785
6.62.31 msql_list_tables() : Liste les tables mSQL sur une base de données....785
6.62.32 msql_listdbs() : Liste les bases de données mSQL sur un serveur....785
6.62.33 msql_listfields() : Liste les champs d'une table....785
6.62.34 msql_listtables() : Liste les tables mSQL sur une base de données....786
6.62.35 msql_num_fields() : Retourne le nombre de champs dans un résultat....786
6.62.36 msql_num_rows() : Retourne le nombre de lignes dans un résultat....786
6.62.37 msql_numfields() : Retourne le nombre de champs dans un résultat....786
6.62.38 msql_numrows() : Retourne le nombre de lignes dans un résultat....786
6.62.39 msql_pconnect() : Ouvre une connexion persistante à un serveur mSQL....787
6.62.40 msql_query() : Envoie une requête mSQL....787
6.62.41 msql_regcase() : Prépare une chaîne pour une recherche par expression régulière insensible à la casse....787
6.62.42 msql_result() : Retourne les données de résultat....787
6.62.43 msql_select_db() : Sélectionne une base de données mSQL....788
6.62.44 msql_selectdb() : Sélectionne une base de données mSQL....788
6.62.45 msql_tablename() : Retourne le nom d'une table à partir d'un nom de champs....788
6.62.46 msql() : Exécute une requête mSQL....789
6.63 MySQL....790
6.63.1 Introduction....790
6.63.2 Pré−requis....790
6.63.3 Installation....790
6.63.4 Configuration à l'exécution....790
6.63.5 Types de ressources....791
6.63.6 Constantes Prédefinies....791
6.63.7 Exemples....792
6.63.9 mysql_change_user() : Change le nom de session de l'utilisateur actif....793
6.63.10 mysql_client_encoding() : Retourne le nom du jeu de caractères utilisé par le client mysql....794
6.63.11 mysql_close() : Ferme la connexion MySQL....794
6.63.12 mysql_connect() : Ouvre une connexion à un serveur MySQL....795
6.63.13 mysql_create_db() : Crée une base de données MySQL....796
6.63.14 mysql_data_seek() : Déplace le pointeur interne de résultat....796
6.63.15 mysql_db_name() : Lit les noms des bases de données....797
6.63.16 mysql_db_query() : Envoie une requête MySQL à un serveur MySQL....797
6.63.17 mysql_drop_db() : Efface une base de données MySQL....798
6.63.18 mysql_errno() : Retourne le numéro de message d'erreur de la dernière opération MySQL....798
6.63.19 mysql_error() : Retourne le texte associé avec l'erreur générée lors de la dernière requête....798
6.63.20 mysql_escape_string() : Protège une chaîne pour la passer à mysql_query....799
6.63.21 mysql_fetch_array() : Retourne une ligne de résultat sous la forme d'un

Sommaire

6 Index des fonctions

tableau associatif....799
6.63.22 mysql_fetch_assoc() : Lit une ligne de résultats dans un tableau associatif....800
6.63.23 mysql_fetch_field() : Retourne les données enregistrées dans une colonne sous forme d'objet....800
6.63.24 mysql_fetch_lengths() : Retourne la taille de chaque colonne d'une ligne de résultat....801
6.63.25 mysql_fetch_object() : Retourne les lignes résultats sous la forme d'un objet....801
6.63.26 mysql_fetch_row() : Retourne une ligne de résultat sous la forme d'un tableau....802
6.63.27 mysql_field_flags() : Retourne le sémaphore associé à la colonne spécifiée dans le résultat courant....802
6.63.28 mysql_field_len() : Retourne la longueur du champs spécifié....803
6.63.29 mysql_field_name() : Retourne le nom d'une colonne....803
6.63.30 mysql_field_seek() : Déplace le pointeur de résultat....803
6.63.31 mysql_field_table() : Retourne le nom de la table où se trouve une colonne....803
6.63.32 mysql_field_type() : Retourne le type de la colonne spécifiée dans le résultat courant....803
6.63.33 mysql_free_result() : Efface le résultat de la mémoire....804
6.63.34 mysql_get_client_info() : Lit les informations sur le client MySQL....804
6.63.35 mysql_get_host_info() : Lit les informations sur l'hôte MySQL....804
6.63.36 mysql_get_proto_info() : Lit les informations sur le protocole MySQL....805
6.63.37 mysql_get_server_info() : Lit les informations sur le serveur MySQL....805
6.63.38 mysql_info() : Lit des informations a propos de la dernière requête....805
6.63.39 mysql_insert_id() : Retourne l'identifiant généré par la dernière requête INSERT....806
6.63.40 mysql_list_dbs() : Liste les bases de données disponibles sur le serveur MySQL....806
6.63.41 mysql_list_fields() : Liste les champs du résultat MySQL....806
6.63.42 mysql_list_processes() : Liste les processus MySQL....807
6.63.43 mysql_list_tables() : Liste les tables d'une base de données....807
6.63.44 mysql_num_fields() : Retourne le nombre de champs d'un résultat....807
6.63.45 mysql_num_rows() : Retourne le nombre de lignes d'un résultat....808
6.63.46 mysql_pconnect() : Ouvre une connexion persistante à un serveur MySQL....808
6.63.47 mysql_ping() : Pingue le serveur, et se reconnecte au besoin....809
6.63.48 mysql_query() : Envoie une requête SQL à un serveur MySQL....809
6.63.49 mysql_real_escape_string() : Protège les caractères spéciaux d'une commande SQL....810
6.63.50 mysql_result() : Retourne un champs d'un résultat....810
6.63.51 mysql_select_db() : Sélectionne une base de données MySQL....811
6.63.52 mysql_stat() : Retourne le statut courant du serveur....811
6.63.53 mysql_tablename() : Lit le nom de la table qui contient le champs spécifié....812
6.63.54 mysql_thread_id() : Retourne l'identifiant du thread courant....812
6.63.55 mysql_unbuffered_query() : Exécute une requête SQL sans mobiliser les résultats....812
6.64 Sessions Mohawk....814
6.64.1 Introduction....814
6.64.2 Pré−requis....814

Sommaire

6 Index des fonctions

6.64.3 Installation....814
6.64.4 Configuration à l'exécution....814
6.64.5 Types de ressources....814
6.64.6 Constantes Prédefinies....814
6.64.8 msession_count() : Compte le nombre de sessions....815
6.64.9 msession_create() : Crée une session....815
6.64.10 msession_destroy() : Détruit la session....815
6.64.11 msession_disconnect() : Ferme la connexion au serveur msession....815
6.64.12 msession_find() : Trouve une valeur....815
6.64.13 msession_get_array() : Lit un tablaeu de ... (??)....815
6.64.14 msession_get() : Lit une valeur dans une session....816
6.64.15 msession_getdata() : Lit les données ... (??)....816
6.64.16 msession_inc() : Incrémente une valeur dans une session....816
6.64.17 msession_list() : Liste ... (??)....816
6.64.18 msession_listvar() : Liste les sessions contenant une variable....816
6.64.19 msession_lock() : Verrouille une session....817
6.64.20 msession_plugin() : Appele une fonction d'échappement dans les plugins de personnalité msession....817
6.64.21 msession_randstr() : Lit une chaîne aléatoire....817
6.64.22 msession_set_array() : Modifie un tableau de ... (??)....817
6.64.23 msession_set() : Modifie une valeur dans la session....817
6.64.24 msession_setdata() : Modifie les données (??)....817
6.64.25 msession_timeout() : Lit et/ou modifie la durée de vie de la session....818
6.64.26 msession_uniq() : Lit un identifiant unique....818
6.64.27 msession_unlock() : Déverrouille une session....818
6.65 Fonctions muscat....819
6.65.2 muscat_get() : Lit la réponse de l'API muscat....819
6.65.3 muscat_give() : Envoie une chaîne à l'API de base muscat....819
6.65.4 muscat_setup_net() : Crée une nouvelle session muscat....819
6.65.5 muscat_setup() : Crée une nouvelle session....820
6.66 Réseau....821
6.66.1 Introduction....821
6.66.2 Pré−requis....821
6.66.3 Installation....821
6.66.4 Configuration à l'exécution....821
6.66.5 Types de ressources....821
6.66.6 Constantes Prédefinies....821
6.66.8 closelog() : Ferme la connexion à l'historique système....822
6.66.9 debugger_off() : Inactive le debugger interne de PHP....823
6.66.10 debugger_on() : Active le debugger interne de PHP....823
6.66.11 define_syslog_variables() : Initialise toutes les constantes liées au syslog....823
6.66.12 fsockopen() : Ouvre une socket de connexion Internet ou Unix....823
6.66.13 gethostbyaddr() : Retourne le nom d'hôte correspondant à une IP....824
6.66.14 gethostbyname() : Retourne l'adresse IP correspondant à un hôte....824
6.66.15 gethostbynamel() : Retourne la liste d'IP correspondante à un hôte....825
6.66.16 getmxrr() : Retourne les enregistrements MX d'un hôte....825
6.66.17 getprotobyname() : Retourne le numéro de protocole associé à un nom de protocole....825
6.66.18 getprotobynumber() : Retourne le nom de protocole associé à un numéro de protocole....825
6.66.19 getservbyname() : Retourne le numéro de port associé à un service

Sommaire

6 Index des fonctions

Internet et un protocole....826
6.66.20 getservbyport() : Retourne le service Internet qui correspond au port et protocole....826
6.66.21 ip2long() : Convertit une chaîne contenant une adresse (IPv4) IP numérique en adresse littérale....826
6.66.22 long2ip() : Convertit une adresse IP (IPv4) en adresse IP numérique....827
6.66.23 openlog() : Ouvre la connexion à l'historique système....827
6.66.24 pfsockopen() : Ouvre une socket de connexion Internet ou Unix persistante....828
6.66.25 socket_get_status() : Retourne les informations sur une socket....828
6.66.26 socket_set_blocking() : Active/désactive le mode bloquant d'une socket....828
6.66.27 socket_set_timeout() : Fixe la durée de vie de la socket....829
6.66.28 syslog() : Génère un message dans l'historique système....829
6.67 Ncurses terminal screen control functions....831
6.67.1 Introduction....831
6.67.2 Pré−requis....831
6.67.3 Installation....831
6.67.4 Configuration à l'exécution....831
6.67.5 Types de ressources....832
6.67.6 Constantes Prédefinies....832
6.67.8 ncurses_addchnstr() : Add attributed string with specified length at current position....837
6.67.9 ncurses_addchstr() : Add attributed string at current position....837
6.67.10 ncurses_addnstr() : Add string with specified length at current position....837
6.67.11 ncurses_addstr() : Output text at current position....838
6.67.12 ncurses_assume_default_colors() : Define default colors for color 0....838
6.67.13 ncurses_attroff() : Turn off the given attributes....838
6.67.14 ncurses_attron() : Turn on the given attributes....838
6.67.15 ncurses_attrset() : Set given attributes....839
6.67.16 ncurses_baudrate() : Returns baudrate of terminal....839
6.67.17 ncurses_beep() : Let the terminal beep....839
6.67.18 ncurses_bkgd() : Set background property for terminal screen....840
6.67.19 ncurses_bkgdset() : Control screen background....840
6.67.20 ncurses_border() : Draw a border around the screen using attributed characters....840
6.67.21 ncurses_can_change_color() : Check if we can change terminals colors....840
6.67.22 ncurses_cbreak() : Switch of input buffering....841
6.67.23 ncurses_clear() : Clear screen....841
6.67.24 ncurses_clrtobot() : Clear screen from current position to bottom....841
6.67.25 ncurses_clrtoeol() : Clear screen from current position to end of line....842
6.67.26 ncurses_color_set() : Set fore− and background color....842
6.67.27 ncurses_curs_set() : Set cursor state....842
6.67.28 ncurses_def_prog_mode() : Saves terminals (program) mode....843
6.67.29 ncurses_def_shell_mode() : Saves terminals (shell) mode....843
6.67.30 ncurses_define_key() : Define a keycode....843
6.67.31 ncurses_delay_output() : Delay output on terminal using padding characters....843
6.67.32 ncurses_delch() : Delete character at current position, move rest of line left....844
6.67.33 ncurses_deleteln() : Delete line at current position, move rest of screen up....844
6.67.34 ncurses_delwin() : Delete a ncurses window....844

Sommaire

6 Index des fonctions

6.67.35 ncurses_doupdate() : Write all prepared refreshes to terminal....845
6.67.36 ncurses_echo() : Activate keyboard input echo....845
6.67.37 ncurses_echochar() : Single character output including refresh....845
6.67.38 ncurses_end() : Stop using ncurses, clean up the screen....845
6.67.39 ncurses_erase() : Erase terminal screen....846
6.67.40 ncurses_erasechar() : Returns current erase character....846
6.67.41 ncurses_filter() : 846
6.67.42 ncurses_flash() : Flash terminal screen (visual bell)....847
6.67.43 ncurses_flushinp() : Flush keyboard input buffer....847
6.67.44 ncurses_getch() : Read a character from keyboard....847
6.67.45 ncurses_getmouse() : Reads mouse event....847
6.67.46 ncurses_halfdelay() : Put terminal into halfdelay mode....848
6.67.47 ncurses_has_colors() : Check if terminal has colors....848
6.67.48 ncurses_has_ic() : Check for insert− and delete−capabilities....849
6.67.49 ncurses_has_il() : Check for line insert− and delete−capabilities....849
6.67.50 ncurses_has_key() : Check for presence of a function key on terminal keyboard....849
6.67.51 ncurses_hline() : Draw a horizontal line at current position using an attributed character and max. n characters long....850
6.67.52 ncurses_inch() : Get character and attribute at current position....850
6.67.53 ncurses_init_color() : Set new RGB value for color....850
6.67.54 ncurses_init_pair() : Allocate a color pair....850
6.67.55 ncurses_init() : Initialize ncurses....851
6.67.56 ncurses_insch() : Insert character moving rest of line including character at current position....851
6.67.57 ncurses_insdelln() : Insert lines before current line scrolling down (negative numbers delete and scroll up)....851
6.67.58 ncurses_insertln() : Insert a line, move rest of screen down....851
6.67.59 ncurses_insstr() : Insert string at current position, moving rest of line right....852
6.67.60 ncurses_instr() : Reads string from terminal screen....852
6.67.61 ncurses_isendwin() : Ncurses is in endwin mode, normal screen output may be performed....852
6.67.62 ncurses_keyok() : Enable or disable a keycode....852
6.67.63 ncurses_killchar() : Returns current line kill character....853
6.67.64 ncurses_longname() : Returns terminals description....853
6.67.65 ncurses_mouseinterval() : Set timeout for mouse button clicks....853
6.67.66 ncurses_mousemask() : Sets mouse options....854
6.67.67 ncurses_move() : Move output position....855
6.67.68 ncurses_mvaddch() : Move current position and add character....855
6.67.69 ncurses_mvaddchnstr() : Move position and add attrributed string with specified length....855
6.67.70 ncurses_mvaddchstr() : Move position and add attributed string....856
6.67.71 ncurses_mvaddnstr() : Move position and add string with specified length....856
6.67.72 ncurses_mvaddstr() : Move position and add string....856
6.67.73 ncurses_mvcur() : Move cursor immediately....856
6.67.74 ncurses_mvdelch() : Move position and delete character, shift rest of line left....857
6.67.75 ncurses_mvgetch() : Move position and get character at new position....857
6.67.76 ncurses_mvhline() : Set new position and draw a horizontal line using an attributed character and max. n characters long....857
6.67.77 ncurses_mvinch() : Move position and get attributed character at new

Sommaire

6 Index des fonctions

position....857
6.67.78 ncurses_mvvline() : Set new position and draw a vertical line using an attributed character and max. n characters long....858
6.67.79 ncurses_mvwaddstr() : Add string at new position in window....858
6.67.80 ncurses_napms() : Sleep....858
6.67.81 ncurses_newwin() : Create a new window....858
6.67.82 ncurses_nl() : Translate newline and carriage return / line feed....859
6.67.83 ncurses_nocbreak() : Switch terminal to cooked mode....859
6.67.84 ncurses_noecho() : Switch off keyboard input echo....859
6.67.85 ncurses_nonl() : Do not translate newline and carriage return / line feed....860
6.67.86 ncurses_noqiflush() : Do not flush on signal characters....860
6.67.87 ncurses_noraw() : Switch terminal out of raw mode....860
6.67.88 ncurses_putp() :....860
6.67.89 ncurses_qiflush() : Flush on signal characters....861
6.67.90 ncurses_raw() : Switch terminal into raw mode....861
6.67.91 ncurses_refresh() : Refresh screen....861
6.67.92 ncurses_resetty() : Restores saved terminal state....862
6.67.93 ncurses_savetty() : Saves terminal state....862
6.67.94 ncurses_scr_dump() : Dump screen content to file....862
6.67.95 ncurses_scr_init() : Initialize screen from file dump....862
6.67.96 ncurses_scr_restore() : Restore screen from file dump....863
6.67.97 ncurses_scr_set() : Inherit screen from file dump....863
6.67.98 ncurses_scrl() : Scroll window content up or down without changing current position....863
6.67.99 ncurses_slk_attr() : Returns current soft label key attribute....863
6.67.100 ncurses_slk_attroff() :....864
6.67.101 ncurses_slk_attron() :....864
6.67.102 ncurses_slk_attrset() :....864
6.67.103 ncurses_slk_clear() : Clears soft labels from screen....864
6.67.104 ncurses_slk_color() : Sets color for soft label keys....865
6.67.105 ncurses_slk_init() : Initializes soft label key functions....865
6.67.106 ncurses_slk_noutrefresh() : Copies soft label keys to virtual screen....865
6.67.107 ncurses_slk_refresh() : Copies soft label keys to screen....866
6.67.108 ncurses_slk_restore() : Restores soft label keys....866
6.67.109 ncurses_slk_touch() : Fources output when ncurses_slk_noutrefresh is performed....866
6.67.110 ncurses_standend() : Stop using 'standout' attribute....866
6.67.111 ncurses_standout() : Start using 'standout' attribute....867
6.67.112 ncurses_start_color() : Start using colors....867
6.67.113 ncurses_termattrs() : Returns a logical OR of all attribute flags supported by terminal....867
6.67.114 ncurses_termname() : Returns terminals (short)−name....867
6.67.115 ncurses_timeout() : Set timeout for special key sequences....868
6.67.116 ncurses_typeahead() : Specify different filedescriptor for typeahead checking....868
6.67.117 ncurses_ungetch() : Put a character back into the input stream....868
6.67.118 ncurses_ungetmouse() : Pushes mouse event to queue....869
6.67.119 ncurses_use_default_colors() : Assign terminal default colors to color id −1....869
6.67.120 ncurses_use_env() : Control use of environment information about terminal size....869

Sommaire

6 Index des fonctions

6.67.121 ncurses_use_extended_names() : Control use of extended names in terminfo descriptions....870
6.67.122 ncurses_vidattr() : 870
6.67.123 ncurses_vline() : Draw a vertical line at current position using an attributed character and max. n characters long....870
6.67.124 ncurses_wrefresh() : Refresh window on terminal screen....870
6.68 Lotus Notes....872
6.68.1 Introduction....872
6.68.3 notes_copy_db() : Crée une note à partir du formulaire form form_name....872
6.68.4 notes_create_db() : Crée une base Lotus Notes....872
6.68.5 notes_create_note() : Crée une note à partir du formulaire form_name....873
6.68.6 notes_drop_db() : Supprime une base Lotus Notes....873
6.68.7 notes_find_note() : Retourne un identifiant de note, dans le serveur database_name. Spécifie le nom de la note....873
6.68.8 notes_header_info() : Ouvre le message msg_number dans la boîte aux lettres et le serveur spécifié....874
6.68.9 notes_list_msgs() : Retourne les notes rangées dans database_name....874
6.68.10 notes_mark_read() : Marque la note note_id comme lu par l'utilisateur user_name....874
6.68.11 notes_mark_unread() : Marque la note note_id comme non plus pour l'utilisateur user_name....874
6.68.12 notes_nav_create() : Crée un nom de navigateur dans database_name....875
6.68.13 notes_search() : Cherche les notes par mots clés dans database_name....875
6.68.14 notes_unread() : Retourne l'identifiant de la note non lue pour l'utilisateur courant user_name....875
6.68.15 notes_version() : Lit la version de Lotus Notes....876
6.69 ODBC unifié....877
6.69.1 Introduction....877
6.69.2 Pré−requis....877
6.69.3 Installation....877
6.69.4 Configuration à l'exécution....877
6.69.5 Types de ressources....878
6.69.6 Constantes Prédefinies....878
6.69.8 odbc_binmode() : Modifie la gestion des colonnes de données binaires....880
6.69.9 odbc_close_all() : Ferme toutes les connexions ODBC....881
6.69.10 odbc_close() : Ferme une connexion ODBC....881
6.69.11 odbc_columnprivileges() : Liste les colonnes et leurs droits associés....881
6.69.12 odbc_columns() : Liste les colonnes d'une table....882
6.69.13 odbc_commit() : Valide une transaction ODBC....883
6.69.14 odbc_connect() : Connexion à une source....883
6.69.15 odbc_cursor() : Lecture du pointeur de fiche courante (cursorname)....883
6.69.16 odbc_do() : Synonyme de odbc_exec....883
6.69.17 odbc_error() : Lit le dernier code d'erreur....884
6.69.18 odbc_errormsg() : Lit le dernier message d'erreur....884
6.69.19 odbc_exec() : Prépare et exécute une requête SQL....884
6.69.20 odbc_execute() : Exécute une requête SQL préparée....884
6.69.21 odbc_fetch_into() : Lit une ligne de résultat, et la place dans un tableau....885
6.69.22 odbc_fetch_row() : Lit une ligne de résultat....885
6.69.23 odbc_field_len() : Lit la longueur d'un champs....886
6.69.24 odbc_field_name() : Lit le nom de la colonne....886
6.69.25 odbc_field_num() : Numéro de colonne....886

Sommaire

6 Index des fonctions

6.69.26 odbc_field_precision() : Alias de odbc_field_len....886
6.69.27 odbc_field_scale() : Lit l'échelle d'un champs....887
6.69.28 odbc_field_type() : Type de données d'un champs....887
6.69.29 odbc_foreignkeys() : Liste les clés étrangères....887
6.69.30 odbc_free_result() : Libère les ressources associées à un résultat....888
6.69.31 odbc_gettypeinfo() : Liste les types de données supportés par une source....888
6.69.32 odbc_longreadlen() : Gestion des colonnes de type LONG....889
6.69.33 odbc_num_fields() : Nombre de colonnes dans un résultat....889
6.69.34 odbc_num_rows() : Nombre de ligne dans un résultat....889
6.69.35 odbc_pconnect() : Ouvre une connexion persistante à une source de données....889
6.69.36 odbc_prepare() : Prépare une commande pour l'exécution....890
6.69.37 odbc_primarykeys() : Liste les colonnes utilisées dans une clé primaire....890
6.69.38 odbc_procedurecolumns() : Liste les paramètres des procédures....890
6.69.39 odbc_procedures() : Liste les procédure stockées....891
6.69.40 odbc_result_all() : Affiche le résultat sous la forme d'une table HTML....891
6.69.41 odbc_result() : Lit les données de résultat....892
6.69.42 odbc_rollback() : Annule une transaction....892
6.69.43 odbc_setoption() : Modifie les paramètres ODBC....892
6.69.44 odbc_specialcolumns() : Retourne l'ensemble optimal de colonnes, qui permettent de définir uniquement une ligne dans une table....893
6.69.45 odbc_statistics() : Calcule des statistiques sur une table....894
6.69.46 odbc_tableprivileges() : Liste les tables et leurs privilèges....894
6.69.47 odbc_tables() : Liste les tables d'une source....895
6.70.3 aggregate_methods_by_regexp() : Aggrège sélectivement les méthodes d'une classe grâce à une expression régulière....896
6.70.4 aggregate_methods() : Aggrège dynamiquement les méthodes d'une classe à un objet....902
6.70.5 aggregate_properties_by_list() : Aggrège sélectivement les propriétés d'une classe grâce à une liste....902
6.70.6 aggregate_properties_by_regexp() : Aggrège sélectivement les propriétés d'une classe grâce à une expression régulière....902
6.70.7 aggregate_properties() : Aggrège dynamiquement les propriétés d'une classe à un objet....903
6.70.8 aggregate() : Aggrège dynamiquemetn des classes et objets....903
6.70.9 aggregation_info() : Retourne un tableau associatif décrivant les classes aggrégées....903
6.70.10 deaggregate() : Desaggrège un objet....904
6.70 Aggregation d'objets/Composition de fonctions....905
6.71 Oracle 8....906
6.71.1 Introduction....906
6.71.2 Pré−requis....906
6.71.3 Installation....907
6.71.4 Configuration à l'exécution....907
6.71.5 Types de ressources....907
6.71.6 Constantes Prédefinies....907
6.71.7 Exemples....908
6.71.9 OCICancel() : Libère les ressouces....910
6.71.10 OCICollAppend() : Non documenté....910
6.71.11 OCICollAssign() : Bientôt documenté....910
6.71.12 OCICollAssignElem() : Bientôt documenté....910

Sommaire

6 Index des fonctions

6.71.13 OCICollGetElem() : Bientôt documenté....910
6.71.14 OCICollMax() : Bientôt documenté....910
6.71.15 OCICollSize() : Bientôt documenté....911
6.71.16 OCICollTrim() : Bientôt documenté....911
6.71.17 ociColumnIsNULL() : Teste si la valeur d'une colonne est NULL....911
6.71.18 ociColumnName() : Retourne le nom d'une colonne....911
6.71.19 OCIColumnPrecision() : Bientôt documenté....912
6.71.20 OCIColumnScale() : Bientôt documenté....912
6.71.21 ociColumnSize() : Retourne la taille de la colonne....912
6.71.22 ociColumnType() : Retourne le type de données d'une colonne....913
6.71.23 OCIColumnTypeRaw() : Bientôt documenté....913
6.71.24 ociCommit() : Valide les transactions en cours....914
6.71.25 ociDefineByName() : Utilise une variable PHP pour la phase de définition, dans une commande SELECT....914
6.71.26 ociError() : Retourne la dernière erreur de stmt|conn|global....914
6.71.27 ociexecute() : Exécute une commande....915
6.71.28 ociFetch() : Modifie la prochaine ligne dans le résultat....915
6.71.29 ociFetchInto() : Retourne la ligne suivante dans un tableau....915
6.71.30 ociFetchStatement() : Retourne toutes les lignes d'un résultat....915
6.71.31 OCIFreeCollection() : Bientôt documenté....916
6.71.32 ociFreeCursor() : Libère toutes les ressources occupées par un pointeur....916
6.71.33 ociFreeDesc() : Supprime un descripteur de LOB....916
6.71.34 ociFreeStatement() : Libère toutes les ressources occupées par une commande....916
6.71.35 ociinternaldebug() : Active ou désactive l'affichage des données de debuggage....917
6.71.36 OCILoadLob() : Bientôt documenté....917
6.71.37 ociLogOff() : Déconnexion d'un serveur Oracle....917
6.71.38 ociLogon() : Etablit une connexion à un serveur Oracle....917
6.71.39 OCINewCollection() : Bientôt documenté....918
6.71.40 ociNewCursor() : Retourne un nouveau pointeur à utiliser pour lier les pointeurs de références....919
6.71.41 ociNewDescriptor() : Initialise un nouveau pointeur vide de LOB/FILE....920
6.71.42 ociNLogon() : Se connecte à un serveur Oracle avec une nouvelle connexion....921
6.71.43 ociNumCols() : Retourne le nombre de colonnes dans un résultat....922
6.71.44 ociparse() : Analyse une requête....923
6.71.45 ociPLogon() : Connection persistante à un serveur Oracle....923
6.71.46 ociResult() : Retourne la valeur d'une colonne....923
6.71.47 ociRollback() : Annule les transactions en cours....923
6.71.48 ociRowCount() : Retourne le nombre de lignes affectées....923
6.71.49 OCISaveLob() : Bientôt documenté....924
6.71.50 OCISaveLobFile() : Bientôt documenté....924
6.71.51 ociServerVersion() : Retourne une chaîne contenant les informations de version du serveur....924
6.71.52 ocisetprefetch() : Indique le nombre de lignes qui doivent être pré−lues....925
6.71.53 ociStatementType() : Retourne le type de commande OCI....925
6.71.54 OCIWriteLobToFile() : Bientôt documenté....925
6.72 OpenSSL....926
6.72.1 Introduction....926
6.72.2 Pré−requis....926

Sommaire

6 Index des fonctions

6.72.3 Installation....926
6.72.4 Configuration à l'exécution....926
6.72.5 Types de ressources....926
6.72.6 Paramètres clés/certificats....926
6.72.7 Vérification de certificats....927
6.72.9 openssl_free_key() : Libère les ressources....927
6.72.10 openssl_get_privatekey() : Prépare une clé privée au format PEM....927
6.72.11 openssl_get_publickey() : Extrait une clé publique d'un certificat....928
6.72.12 openssl_open() : Ouvre des données scellées....928
6.72.13 openssl_pkcs7_decrypt() : Déchiffre un message S/MIME....929
6.72.14 openssl_pkcs7_encrypt() : Chiffre un message S/MIME....929
6.72.15 openssl_pkcs7_sign() : Signe un message S/MIME....930
6.72.16 openssl_pkcs7_verify() : Vérifie la signature d'un message S/MIME....931
6.72.17 openssl_seal() : Scelle des données....932
6.72.18 openssl_sign() : Signe les données....932
6.72.19 openssl_verify() : Vérifie une signature....933
6.72.20 openssl_x509_checkpurpose() : Vérifie l'usage d'un certificat....933
6.72.21 openssl_x509_free() : Libère les ressources prises par un certificat....934
6.72.22 openssl_x509_parse() : Analyse un certificat X509....935
6.72.23 openssl_x509_read() : Analyse un certificat X.509 et retourne une ressource....935
6.73 Oracle....936
6.73.1 Introduction....936
6.73.2 Constantes Prédefinies....936
6.73.4 Ora_Close() : Ferme un pointeur Oracle....936
6.73.5 Ora_ColumnName() : Retourne le nom de la colonne de résultat....937
6.73.6 Ora_ColumnSize() : Lit la taille d'une colonne....937
6.73.7 Ora_ColumnType() : Retourne le type de la colonne de résultat....937
6.73.8 Ora_Commit() : Valide une transaction Oracle....937
6.73.9 Ora_CommitOff() : Inactive la validation automatique....938
6.73.10 Ora_CommitOn() : Active la validation automatique....938
6.73.11 Ora_Do() : Analyse, exécute et lit une requête....938
6.73.12 Ora_Error() : Retourne le message d'erreur Oracle....938
6.73.13 Ora_ErrorCode() : Retourne le code d'erreur Oracle....939
6.73.14 Ora_Exec() : Exécute une commande analysée sur un pointeur Oracle....939
6.73.15 Ora_Fetch_Into() : Lit une ligne dans un tableau....939
6.73.16 Ora_Fetch() : Retourne une ligne de résultat....940
6.73.17 Ora_GetColumn() : Retourne une donnée d'une ligne lue....940
6.73.18 Ora_Logoff() : Ferme une connexion Oracle....940
6.73.19 Ora_Logon() : Ouvre une connexion Oracle....940
6.73.20 Ora_Numcols() : Retourne le nombre de colonnes....941
6.73.21 Ora_Numrows() : Retourne le nombre de colonnes....941
6.73.22 Ora_Open() : Ouvre un pointeur Oracle....941
6.73.23 Ora_Parse() : Analyse une requête SQL....941
6.73.24 Ora_pLogon() : Ouvre une connexion persistante à Oracle....942
6.73.25 Ora_Rollback() : Annule une transaction....942
6.74 Ovrimos SQL....943
6.74.2 ovrimos_commit() : Valide une transaction....943
6.74.3 ovrimos_connect() : Connexion à un serveur....944
6.74.4 ovrimos_cursor() : Retourne le nom du curseur....944
6.74.5 ovrimos_exec() : Exécute une requête SQL....944

Sommaire

6 Index des fonctions

6.74.6 ovrimos_execute() : Exécute une requête préparée....945
6.74.7 ovrimos_fetch_into() : Lit une ligne dans un résultat....945
6.74.8 ovrimos_fetch_row() : Lit une ligne dans un résultat....946
6.74.9 ovrimos_field_len() : Retourne la taille d'une colonne....946
6.74.10 ovrimos_field_name() : Retourne le nom d'une colonne....946
6.74.11 ovrimos_field_num() : Retourne le numéro de colonne....947
6.74.12 ovrimos_field_type() : Retourne le type numérique d'une colonne....947
6.74.13 ovrimos_free_result() : Libère les ressources utilisées par un résultat....947
6.74.14 ovrimos_longreadlen() : Indique la taille des données à lire dans une colonne de grande taille....947
6.74.15 ovrimos_num_fields() : Retourne le nombre de colonnes....948
6.74.16 ovrimos_num_rows() : Retourne le nombre de lignes affectées par une modification....948
6.74.17 ovrimos_prepare() : Prépare une requête SQL....948
6.74.18 ovrimos_result_all() : Affiche un résultat sous forme de table HTML....949
6.74.19 ovrimos_result() : Lit le contenu d'une colonne....950
6.74.20 ovrimos_rollback() : Annule une transaction....950
6.75 Entrées/sorties....951
6.75.1 Introduction....951
6.75.2 Pré−requis....951
6.75.3 Installation....951
6.75.4 Configuration à l'exécution....951
6.75.5 Types de ressources....952
6.75.6 Constantes Prédefinies....952
6.75.7 Exemples....952
6.75.8 Voir aussi....952
6.75.10 ob_clean() : Efface le buffer de sortie....953
6.75.11 ob_end_clean() : Détruit les données du buffer de sortie, et éteint la bufferisation de sortie....953
6.75.12 ob_end_flush() : Envoie les données du buffer de sortie, et éteint la bufferisation de sortie....953
6.75.13 ob_flush() : Envoie le buffer de sortie....954
6.75.14 ob_get_contents() : Retourne le contenu du buffer de sortie....954
6.75.15 ob_get_length() : Retourne la longueur du contenu du buffer de sortie....954
6.75.16 ob_get_level() : Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie....954
6.75.17 ob_get_status() : Lit le statut du buffer de sortie....954
6.75.18 ob_gzhandler() : Fonction de callback pour la compression automatique des buffers....955
6.75.19 ob_implicit_flush() : Active/désactive l'envoi implicite....955
6.75.20 ob_start() : Enclenche la bufferisation de sortie....956
6.76 Overload....958
6.76.1 Introduction....958
6.77 PDF....960
6.77.1 Confusion entre les vieilles versions de PDFLib....960
6.77.2 Conseils pour installer PDFLib 3.x....961
6.77.3 Choix de la version de PDFlib....961
6.77.4 Installation des anciennes versions de PDFLib....961
6.77.5 Exemples....962
6.77.7 pdf_add_bookmark() : Ajoute un signet à la page courante....965
6.77.8 pdf_add_launchlink() : Ajoute une annotation éxécutable dans la page

Sommaire

6 Index des fonctions

courante....966
6.77.9 pdf_add_locallink() : Ajoute un lien sur une annotation dans la page courante....966
6.77.10 pdf_add_note() : Ajoute une note d'annotation dans la page courante....966
6.77.11 pdf_add_outline() : Obsolète: Ajoute un signet dans la page courante....966
6.77.12 pdf_add_pdflink() : Ajoute un lien sur un fichier dans la page courante....966
6.77.13 pdf_add_weblink() : Ajoute un lien hypertexte dans la page courante....967
6.77.14 pdf_arc() : Dessine un arc....967
6.77.15 pdf_attach_file() : Attache un fichier à la page courante....967
6.77.16 pdf_begin_page() : Commence une nouvelle page....967
6.77.17 pdf_circle() : Dessine un cercle....967
6.77.18 pdf_clip() : Aligne sur le chemin courant....968
6.77.19 pdf_close_image() : Ferme une image....968
6.77.20 pdf_close() : Ferme un document PDF....968
6.77.21 pdf_closepath_fill_stroke() : Remplis, dessine et ferme le chemin courant....968
6.77.22 pdf_closepath_stroke() : Ferme le chemin et dessine le long du chemin....968
6.77.23 pdf_closepath() : Ferme et clos le chemin....969
6.77.24 pdf_concat() : Concatène une matrice avec CTM....969
6.77.25 pdf_continue_text() : Affiche un texte sur une nouvelle ligne....969
6.77.26 pdf_curveto() : Dessine une courbe....969
6.77.27 pdf_delete() : Efface un objet PDF....969
6.77.28 pdf_end_page() : Termine une page....970
6.77.29 pdf_endpath() : Ferme le chemin courant....970
6.77.30 pdf_fill_stroke() : Remplis et dessine le chemin courant....970
6.77.31 pdf_fill() : Remplis le chemin courant....970
6.77.32 pdf_findfont() : Prépare une police pour utilisation ultérieure....970
6.77.33 pdf_get_buffer() : Lit un buffer contenant des données PDF....971
6.77.34 pdf_get_font() : Obsolète : gestion de police....971
6.77.35 pdf_get_fontname() : Obsolète : gestion de nom de police....971
6.77.36 pdf_get_fontsize() : Obsolète : gestion de taille de police....971
6.77.37 pdf_get_image_height() : Retourne la hauteur d'une image....971
6.77.38 pdf_get_image_width() : Retourne la largeur d'une image....972
6.77.39 pdf_get_parameter() : Lit la valeur d'un paramètre PDFLib chaîne....972
6.77.40 pdf_get_value() : Lit la valeur d'un paramètre PDFLib numérique....972
6.77.41 pdf_lineto() : Dessine une ligne....972
6.77.42 pdf_moveto() : Déplace le point courant....972
6.77.43 pdf_new() : Crée un nouvel objet PDF....972
6.77.44 pdf_open_CCITT() : Ouvre une nouvelle image à partir de données CCITT....973
6.77.45 pdf_open_file() : Ouvre un nouvel objet PDF....973
6.77.46 pdf_open_gif() : Obsolète: Ouvre une image GIF....973
6.77.47 pdf_open_image_file() : Lit une image depuis un fichier....973
6.77.48 pdf_open_image() : Fonction générique pour les images....974
6.77.49 pdf_open_jpeg() : Obsolète: Ouvre une image JPEG....974
6.77.50 pdf_open_memory_image() : Ouvre une image créée par les fonctions images PHP....974
6.77.51 pdf_open_png() : Obsolète: Ouvre une image PNG....975
6.77.52 pdf_open_tiff() : Obsolète: Ouvre une image TIFF....975
6.77.53 pdf_open() : Obsolète: Ouvre un nouvel objet PDF....975
6.77.54 pdf_place_image() : Place une image dans la page....975
6.77.55 pdf_rect() : Dessine un rectangle....975
6.77.56 pdf_restore() : Restaure un environnement sauvé....976

Sommaire

6 Index des fonctions

6.77.57 pdf_rotate() : Choisi la rotation....976
6.77.58 pdf_save() : Enregistre l'environnement courant....976
6.77.59 pdf_scale() : Modifie l'échelle....976
6.77.60 pdf_set_border_color() : Modifie la couleur des liens et annotations....977
6.77.61 pdf_set_border_dash() : Modifie les pointillés des liens et annotations....977
6.77.62 pdf_set_border_style() : Modifie le bord des liens et annotations....977
6.77.63 pdf_set_char_spacing() : Fixe l'espacement des caractères....977
6.77.64 pdf_set_duration() : Choisi la durée de transition entre deux pages....978
6.77.65 pdf_set_font() : Sélectionne la police et sa taille....978
6.77.66 pdf_set_horiz_scaling() : Fixe l'echelle horizontale du texte....978
6.77.67 pdf_set_info() : Remplis les entêtes du document....978
6.77.68 pdf_set_leading() : Obsolète : Modifie la distance entre les lignes du texte....979
6.77.69 pdf_set_parameter() : Modifie certains paramètres....979
6.77.70 pdf_set_text_matrix() : Obsolète: Modifie la transition des pages....979
6.77.71 pdf_set_text_pos() : Fixe la position du texte....980
6.77.72 pdf_set_text_rendering() : Détermine le rendu du texte....980
6.77.73 pdf_set_value() : Modifie certains paramètre numériques....980
6.77.74 pdf_set_word_spacing() : Fixe l'espacement des mots....980
6.77.75 pdf_setdash() : Modifie les caractères de remplissage....980
6.77.76 pdf_setflat() : Modifie la platitude (flatness)....981
6.77.77 pdf_setfont() : Modifie la police courante....981
6.77.78 pdf_setgray_fill() : Modifie la couleur grise comme couleur de remplissage....981
6.77.79 pdf_setgray_stroke() : Modifie la couleur de dessin à un niveau de gris....981
6.77.80 pdf_setgray() : Modifie la couleur grise comme couleur de remplissage et de dessin....981
6.77.81 pdf_setlinecap() : Modifie le paramètre linecap....982
6.77.82 pdf_setlinejoin() : Modifie le paramètre linejoin....982
6.77.83 pdf_setlinewidth() : Modifie la largeur de ligne....982
6.77.84 pdf_setmiterlimit() : Modifie la "miter limit"....982
6.77.85 pdf_setpolydash() : Modifie les pointillés compliqués....982
6.77.86 pdf_setrgbcolor_fill() : Modifie la couleur rgb comme couleur de remplissage....982
6.77.87 pdf_setrgbcolor_stroke() : Modifie la couleur rgb comme couleur de dessin....983
6.77.88 pdf_setrgbcolor() : Modifie la couleur rgb comme couleur de dessin et de remplissage....983
6.77.89 pdf_show_boxed() : Affiche un texte dans un rectangle....983
6.77.90 pdf_show_xy() : Affiche un texte à une position donnée....983
6.77.91 pdf_show() : Affiche un texte à la position courante....984
6.77.92 pdf_skew() : Modifie le système de coordonnées....984
6.77.93 pdf_stringwidth() : Retourne la largeur du texte avec la police courante....984
6.77.94 pdf_stroke() : Dessine le long du chemin....984
6.77.95 pdf_translate() : Modifie l'origine du système de coordonnées....984
6.78 Paiement par Verisign....986
6.78.1 Introduction....986
6.78.2 Pré−requis....986
6.78.3 Installation....986
6.78.4 Configuration à l'exécution....986
6.78.5 Types de ressources....987
6.78.6 Constantes Prédefinies....987
6.78.8 pfpro_init() : Initialise la librairie Payflow Pro....987

Sommaire

6 Index des fonctions

6.78.9 pfpro_process_raw() : Envoie une transaction brute à Payflow Pro....987
6.78.10 pfpro_process() : Effectue une transaction avec Payflow Pro....988
6.78.11 pfpro_version() : Lit le numéro de version de Payflow Pro....989
6.79 Options PHP et informations....990
6.79.1 Introduction....990
6.79.2 Pré−requis....990
6.79.3 Installation....990
6.79.4 Configuration à l'exécution....990
6.79.5 Types de ressources....991
6.79.6 Constantes Prédefinies....991
6.79.8 assert() : Vérifie si une assertion est fausse....993
6.79.9 dl() : Charge une extension PHP à la volée....993
6.79.10 extension_loaded() : Détermine si une extension est chargée ou non....994
6.79.11 get_cfg_var() : Retourne la valeur d'une option de PHP....994
6.79.12 get_current_user() : Retourne le nom du possesseur du script courant....994
6.79.13 get_defined_constants() : Retourne la liste des constantes et leur valeur....994
6.79.14 get_extension_funcs() : Liste les fonctions d'une extension....995
6.79.15 get_included_files() : Retourne un tableau avec les noms des fichiers qui sont inclus dans un script....995
6.79.16 get_loaded_extensions() : Retourne la liste de tous les modules compilés et chargés....996
6.79.17 get_magic_quotes_gpc() : Retourne la configuration actuelle de l'option magic_quotes_gpc....997
6.79.18 get_magic_quotes_runtime() : Retourne la configuration actuelle de l'option magic_quotes_runtime....997
6.79.19 get_required_files() : Retourne un tableau avec les noms des fichiers qui sont requis et inclus dans un script....998
6.79.20 getenv() : Retourne la valeur de la variable d'environnement....998
6.79.21 getlastmod() : Retourne la date de dernière modification de la page....998
6.79.22 getmygid() : Retourne le GID du propriétaire du script....999
6.79.23 getmyinode() : Retourne l'inode du script....999
6.79.24 getmypid() : Retourne le numéro de processus courant....999
6.79.25 getmyuid() : Retourne l'UID du propriétaire du script actuel....999
6.79.26 getopt() : Lit des options passés dans la ligne de commande....1000
6.79.27 getrusage() : Retourne le niveau d'utilisation des ressources....1000
6.79.28 ini_alter() : Change la valeur d'une option de configuration....1001
6.79.29 ini_get_all() : Lit toutes les valeurs de configuration....1001
6.79.30 ini_get() : Lit la valeur d'une option de configuration....1001
6.79.31 ini_restore() : Restaure la valeur de l'option de configuration....1001
6.79.32 ini_set() : Sets the value of a configuration option....1001
6.79.33 php_ini_scanned_files() : Retourne la liste des fichiers .ini analysédans les dossiers de configuration supplémentaires....1007
6.79.34 php_logo_guid() : Retourne le logo....1008
6.79.35 php_sapi_name() : Retourne le type d'interface utilisé entre le serveur web et PHP....1008
6.79.36 php_uname() : Retourne les informations sur le système d'exploitation....1009
6.79.37 phpcredits() : Imprime les crédits de PHP....1009
6.79.38 phpinfo() : Affiche de nombreuses informations sur le PHP....1010
6.79.39 phpversion() : Retourne le numéro de la version courante de PHP....1010
6.79.40 putenv() : Fixe la valeur d'une variable d'environnement....1011
6.79.41 set_magic_quotes_runtime() : Active/désactive l'option

Sommaire

6 Index des fonctions

magic_quotes_runtime....1011
6.79.42 set_time_limit() : Fixe le temps maximum d'exécution d'un script....1012
6.79.43 version_compare() : Compare deux versions de PHP....1012
6.79.44 zend_logo_guid() : Retourne le logo de Zend....1012
6.79.45 zend_version() : Lit la version courante du moteur Zend....1013
6.80 POSIX....1014
6.80.2 posix_getcwd() : Chemin du dossier courant....1014
6.80.3 posix_getegid() : Retourne l'ID effectif du groupe du processus courant....1014
6.80.4 posix_geteuid() : Retourne l'UID effectif de l'utilisateur du processus courant....1015
6.80.5 posix_getgid() : Retourne l'UID du groupe du processus courant....1015
6.80.6 posix_getgrgid() : Retourne des informations sur un groupe....1015
6.80.7 posix_getgrnam() : Retourne des informations sur un groupe....1015
6.80.8 posix_getgroups() : Retourne les identifiants du groupe du processus courant....1015
6.80.9 posix_getlogin() : Retourne le nom de login....1015
6.80.10 posix_getpgid() : Retourne l'id du groupe de processus....1016
6.80.11 posix_getpgrp() : Retourne l'identifiant du groupe de processus....1016
6.80.12 posix_getpid() : Retourne l'identifiant du processus courant....1016
6.80.13 posix_getppid() : Retourne l'identifiant du processus parent....1016
6.80.14 posix_getpwnam() : Retourne des informations sur un utilisateur....1016
6.80.15 posix_getpwuid() : Retourne des informations sur un utilisateur....1017
6.80.16 posix_getrlimit() : Retourne les limites système....1018
6.80.17 posix_getsid() : Retourne le sid du processus....1018
6.80.18 posix_getuid() : Retourne l'ID de l'utilisateur du processus courant....1018
6.80.19 posix_isatty() : Détermine si un pointeur de fichier est un terminal interactif....1018
6.80.20 posix_kill() : Envoie un signal à un processus....1018
6.80.21 posix_mkfifo() : Crée un fichier fifo (first in, first out) (un pipe nommé)....1019
6.80.22 posix_setegid() : Modifie le GID réel du processus courant....1019
6.80.23 posix_seteuid() : Set the effective UID of the current process....1019
6.80.24 posix_setgid() : Fixe le GID effective du processus courant....1019
6.80.25 posix_setpgid() : Fixe l'identifiant de group de processus....1019
6.80.26 posix_setsid() : Fait du processus courant un chef de session....1020
6.80.27 posix_setuid() : Fixe l'UID effective du processus courant....1020
6.80.28 posix_times() : Utilisation des ressources....1020
6.80.29 posix_ttyname() : Retourne le nom de device du terminal....1020
6.80.30 posix_uname() : Retourne le nom du système....1020
6.81 PostgreSQL....1022
6.81.1 Introduction....1022
6.81.2 Pré−requis....1022
6.81.3 Installation....1022
6.81.4 Configuration à l'exécution....1022
6.81.5 Trucs et astuces d'utilisation....1023
6.81.6 Constantes Prédefinies....1024
6.81.7 Exemples....1025
6.81.9 pg_client_encoding() : Lit l'encodage du client....1026
6.81.10 pg_Close() : Termine une connexion PostgreSQL....1027
6.81.11 pg_connect() : Ouvre une connexion....1027
6.81.12 pg_DBname() : Nom de la base de données....1028
6.81.13 pg_end_copy() : Synchronise avec le serveur PostgreSQL....1028

Sommaire

6 Index des fonctions

6.81.14 pg_Fetch_Array() : Lit une ligne dans un tableau....1028
6.81.15 pg_Fetch_Object() : Lit une ligne dans un objet....1029
6.81.16 pg_Fetch_Row() : Lit une ligne dans un tableau....1030
6.81.17 pg_field_is_null() : Teste si un champs est à NULL....1031
6.81.18 pg_field_Name() : Retourne le nom d'un champs....1031
6.81.19 pg_field_Num() : Retourne le numéro d'une colonne....1031
6.81.20 pg_field_PrtLen() : Retourne la taille imprimée....1031
6.81.21 pg_field_Size() : Retourne la taille interne de stockage d'un champs donné....1032
6.81.22 pg_field_Type() : Retourne le type d'un champs donné par index....1032
6.81.23 pg_Free_Result() : Libère la mémoire....1032
6.81.24 pg_Host() : Retourne le nom d'hôte....1032
6.81.25 pg_lo_close() : Ferme un objet de grande taille....1032
6.81.26 pg_lo_create() : Crée un objet de grande taille....1033
6.81.27 pg_lo_export() : Exporte un objet de grande taille vers un fichier....1033
6.81.28 pg_lo_import() : Importe un objet de grande taille depuis un fichier....1033
6.81.29 pg_lo_open() : Ouvre un objet de grande taille....1034
6.81.30 pg_lo_read_all() : Lit un objet de grande taille en totalité....1034
6.81.31 pg_lo_read() : Lit un objet de grande taille....1034
6.81.32 pg_lo_unlink() : Efface un objet de grande taille....1034
6.81.33 pg_lo_write() : Ecrit un objet de grande taille....1035
6.81.34 pg_Num_Fields() : Retourne le nombre de champs....1035
6.81.35 pg_num_rows() : Retourne le nombre de lignes....1035
6.81.36 pg_Options() : Retourne les options....1035
6.81.37 pg_pConnect() : Etablit une connexion persistante....1036
6.81.38 pg_Port() : Retourne le numéro de port....1036
6.81.39 pg_put_line() : Envoie une chaîne au serveur PostgreSQL....1036
6.81.40 pg_set_client_encoding() : Choisit l'encodage du client....1037
6.81.41 pg_trace() : Active le suivi d'une connexion PostgreSQL....1037
6.81.42 pg_tty() : Retourne le nom de tty....1037
6.81.43 pg_untrace() : Termine le suivi d'une connexion PostgreSQL....1037
6.82 Contrôle des processus....1039
6.82.1 Introduction....1039
6.82.2 Pré−requis....1039
6.82.3 Installation....1039
6.82.4 Configuration à l'exécution....1039
6.82.5 Types de ressources....1039
6.82.6 Constantes Prédefinies....1039
6.82.7 Exemples....1040
6.82.8 Voir aussi....1041
6.82.10 pcntl_fork() : Forks the currently running process....1042
6.82.11 pcntl_signal() : Installe un gestionnaire de signaux....1042
6.82.12 pcntl_waitpid() : Attend la fin de l'exécution d'un processus fils....1042
6.82.13 pcntl_wexitstatus() : Retourne le code d'un processus fils terminé....1043
6.82.14 pcntl_wifexited() : Retourne TRUE si le code de retour représente une fin normale....1043
6.82.15 pcntl_wifsignaled() : Returns TRUE if status code represents a termination due to a signal....1043
6.82.16 pcntl_wifstopped() : Returns TRUE if child process is currently stopped....1044
6.82.17 pcntl_wstopsig() : Returns the signal which caused the child to stop....1044
6.82.18 pcntl_wtermsig() : Returns the signal which caused the child to terminate....1044

Sommaire

6 Index des fonctions

6.83 Exécution de programmes externes....1045
6.83.2 escapeshellcmd() : Echappe les méta−caractères Shell....1045
6.83.3 exec() : Exécute un programme externe....1045
6.83.4 passthru() : Exécute un programme externe et affiche le résultat brut....1046
6.83.5 proc_close() : Ferme un processus ouvert par proc_open() et retourne le code de sortie du processus....1046
6.83.6 proc_open() : Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties....1046
6.83.7 shell_exec() : Exécute une commande via le shell et retourne le résultat sous forme de chaîne....1047
6.83.8 system() : Exécute un programme externe et affiche le résultat....1048
6.84 Printer functions....1049
6.84.1 Introduction....1049
6.84.2 Installation....1049
6.84.3 Configuration à l'exécution....1049
6.84.5 printer_close() : Close an open printer connection....1050
6.84.6 printer_create_brush() : Create a new brush....1050
6.84.7 printer_create_dc() : Create a new device context....1050
6.84.8 printer_create_font() : Create a new font....1051
6.84.9 printer_create_pen() : Create a new pen....1052
6.84.10 printer_delete_brush() : Delete a brush....1052
6.84.11 printer_delete_dc() : Delete a device context....1052
6.84.12 printer_delete_font() : Delete a font....1052
6.84.13 printer_delete_pen() : Delete a pen....1052
6.84.14 printer_draw_bmp() : Draw a bmp....1053
6.84.15 printer_draw_chord() : Draw a chord....1053
6.84.16 printer_draw_elipse() : Draw an ellipse....1054
6.84.17 printer_draw_line() : Draw a line....1055
6.84.18 printer_draw_pie() : Draw a pie....1055
6.84.19 printer_draw_rectangle() : Draw a rectangle....1056
6.84.20 printer_draw_roundrect() : Draw a rectangle with rounded corners....1057
6.84.21 printer_draw_text() : Draw text....1057
6.84.22 printer_end_doc() : Close document....1058
6.84.23 printer_end_page() : Close active page....1058
6.84.24 printer_get_option() : Retrieve printer configuration data....1058
6.84.25 printer_list() : Return an array of printers attached to the server....1059
6.84.26 printer_logical_fontheight() : Get logical font height....1059
6.84.27 printer_open() : Open connection to a printer....1059
6.84.28 printer_select_brush() : Select a brush....1060
6.84.29 printer_select_font() : Select a font....1061
6.84.30 printer_select_pen() : Select a pen....1061
6.84.31 printer_set_option() : Configure the printer connection....1062
6.84.32 printer_start_doc() : Start a new document....1063
6.84.33 printer_start_page() : Start a new page....1063
6.84.34 printer_write() : Write data to the printer....1063
6.85 Pspell....1064
6.85.1 Introduction....1064
6.85.2 Pré−requis....1064
6.85.3 Installation....1064
6.85.4 Configuration à l'exécution....1064
6.85.5 Types de ressources....1064

Sommaire

6 Index des fonctions

6.85.6 Constantes Prédefinies....1064
6.85.8 pspell_add_to_session() : Ajoute le mot au dictionnaire personnel de la session courante....1065
6.85.9 pspell_check() : Vérifie un mot....1065
6.85.10 pspell_clear_session() : Remet à zéro la session courante....1065
6.85.11 pspell_config_create() : Crée une configuration utilisée pour ouvrir un dictionnaire....1066
6.85.12 pspell_config_ignore() : Ignore les mots des moins de N caractères....1067
6.85.13 pspell_config_mode() : Change le mode de suggestion....1067
6.85.14 pspell_config_personal() : Choisit le fichier qui contient le dictionnaire personnel....1067
6.85.15 pspell_config_repl() : Choisit le fichier qui contient les paires de remplacement....1068
6.85.16 pspell_config_runtogether() : Considère deux mots accolés comme un composé....1068
6.85.17 pspell_config_save_repl() : Active la sauvegarde des paires de remplacement....1069
6.85.18 pspell_new_config() : Charge un nouveau dictionnaire....1069
6.85.19 pspell_new_personal() : Charge un nouveau dictionnaire avec un dictionnaire personnel....1069
6.85.20 pspell_new() : Charge un nouveau dictionnaire....1070
6.85.21 pspell_save_wordlist() : Sauve le dictionnaire personnel dans un fichier....1071
6.85.22 pspell_store_replacement() : Enregistre une paire de remplacement pour un mot....1072
6.85.23 pspell_suggest() : Suggèegrave;re une orthographe....1072
6.86 Readline (GNU)....1073
6.86.2 readline_clear_history() : Efface l'historique....1073
6.86.3 readline_completion_function() : Enregistre une fonction de complétion....1073
6.86.4 readline_info() : Lit/modifie diverses variables internes....1073
6.86.5 readline_list_history() : Liste l'historique....1074
6.86.6 readline_read_history() : Lit l'historique....1074
6.86.7 readline_write_history() : Ecrit dans l'historique....1074
6.86.8 readline() : Lit une ligne....1074
6.87 Recode (GNU)....1075
6.87.2 recode_string() : Recode une chaîne en fonction de la requête....1075
6.87.3 recode() : Recode une fonction grâce à une requête....1075
6.88 Expressions régulières compatibles Perl....1076
6.88.1 Introduction....1076
6.88.2 Pré−requis....1076
6.88.3 Installation....1076
6.88.4 Configuration à l'exécution....1076
6.88.5 Types de ressources....1076
6.88.6 Constantes Prédefinies....1076
6.88.7 Exemples....1077
6.88.9 syntaxe des masques() : Fonctionnement des expressions régulières....1078
6.88.10 preg_grep() : Retourne un tableau avec les résultat de la recherche....1078
6.88.11 preg_match_all() : Expression régulière globale....1078
6.88.12 preg_match() : Expression régulière standard....1080
6.88.13 preg_quote() : Echappement des caractères spéciaux des expressions régulières....1081
6.88.14 preg_replace_callback() : Rechercher/remplacer avec fonction de

Sommaire

6 Index des fonctions

callback....1082
6.88.15 preg_replace() : Rechercher et remplacer par expression régulière standard....1082
6.88.16 preg_split() : Eclatement d'une chaîne par expression régulière....1084
6.89 qtdom....1086
6.89.2 qdom_tree() : Crée une arbre à partir d'un code XML....1086
6.90 Expressions régulières....1087
6.90.2 ereg() : Expression régulière standard....1088
6.90.3 eregi_replace() : Remplacement par expression régulière insensible à la casse....1088
6.90.4 eregi() : Recherche par expression régulière insensible à la casse....1089
6.90.5 split() : Scinde une chaîne en un tableau, grâce à une expression régulière....1089
6.90.6 spliti() : Scinde une chaîne en un tableau, grâce à une expression régulière....1090
6.90.7 sql_regcase() : Prépare une expression régulière pour effectuer une recherche insensible à la casse....1090
6.91 Sémaphores et gestion de la mémoire partagée....1091
6.91.1 Introduction....1091
6.91.2 Pré−requis....1091
6.91.3 Installation....1091
6.91.4 Configuration à l'exécution....1091
6.91.5 Types de ressources....1092
6.91.6 Constantes Prédefinies....1092
6.91.8 sem_get() : Retourne un identifiant de sémaphore....1092
6.91.9 sem_release() : Libère un sémaphore....1092
6.91.10 shm_attach() : Crée ou ouvre un segment de mémoire partagée....1093
6.91.11 shm_detach() : Libère un segment de mémoire partagée....1093
6.91.12 shm_get_var() : Lit une variable dans la mémoire partagée....1093
6.91.13 shm_put_var() : Insère ou modifie une variable de la mémoire partagée....1094
6.91.14 shm_remove_var() : Efface une variable de la mémoire partagée....1094
6.91.15 shm_remove() : Supprime un segment de mémoire partagée sous Unix....1094
6.92 SESAM....1095
6.92.1 Introduction....1095
6.92.2 Configuration à l'exécution....1095
6.92.3 Notes sur la configuration....1096
6.92.4 Considérations à l'exécution....1097
6.92.5 Types de curseurs....1097
6.92.6 Portage....1098
6.92.7 Sécurité....1098
6.92.8 Migration d'une autre base SQL....1098
6.92.9 Notes sur l'utilisation de types SQL divers....1099
6.92.10 Support des "champs multiples" de SESAM....1100
6.92.11 Voir aussi....1100
6.92.13 sesam_commit() : Valide la transaction SESAM en cours....1101
6.92.14 sesam_connect() : Ouvre une connexion SESAM....1101
6.92.15 sesam_diagnostic() : Retourne l'état de la dernière requête SESAM....1102
6.92.16 sesam_disconnect() : Déconnexion d'une base SESAM....1103
6.92.17 sesam_errormsg() : retourne le message d'erreur....1104
6.92.18 sesam_execimm() : Exécute immédiatement une requête SQL....1104
6.92.19 sesam_fetch_array() : Lit une ligne dans un tableau associatif....1105

Sommaire

6 Index des fonctions

6.92.20 sesam_fetch_result() : Retourne tout ou partie d'un résultat SESAM....1106
6.92.21 sesam_fetch_row() : Lit une ligne dans un tableau....1107
6.92.22 sesam_field_array() : Retourne des informations sur une colonne....1108
6.92.23 sesam_field_name() : Retourne le nom d'une colonne....1110
6.92.24 sesam_free_result() : Libère les ressources....1110
6.92.25 sesam_num_fields() : Retourne le nombre de colonne dans un résultat....1110
6.92.26 sesam_query() : Exécute une requête SESAM....1110
6.92.27 sesam_rollback() : Annule une transaction SESAM....1112
6.92.28 sesam_seek_row() : Déplace un curseur à défilement....1112
6.92.29 sesam_settransaction() : Modifie les paramètres de transaction SESAM....1113
6.93 Sessions....1114
6.93.1 Introduction....1114
6.93.2 Sessions and security....1114
6.93.3 Pré−requis....1115
6.93.4 Installation....1115
6.93.5 Configuration à l'exécution....1115
6.93.6 Types de ressources....1117
6.93.7 Constantes Prédefinies....1117
6.93.8 Exemples....1117
6.93.9 Passer l'identifiant de session (session ID)....1119
6.93.10 Gestion personnalisée des sessions....1120
6.93.12 session_cache_limiter() : Lit et/ou modifie le limiteur de cache....1121
6.93.13 session_decode() : Décode les données de session....1122
6.93.14 session_destroy() : Détruit une session....1122
6.93.15 session_encode() : Encode les données de session....1123
6.93.16 session_get_cookie_params() : Lit la configuration du cookie de session....1123
6.93.17 session_id() : Let et/ou modifie l'identifiant courant de session....1123
6.93.18 session_is_registered() : Vérifie si une variable est enregistrée dans la session....1123
6.93.19 session_module_name() : Lit et/ou modifie le module de session courant....1124
6.93.20 session_name() : Lit et/ou modifie le nom de la session....1124
6.93.21 session_readonly() : Initialise une session en mode lecture....1124
6.93.22 session_register() : Enregistre une variable dans une session....1125
6.93.23 session_save_path() : Lit et/ou modifie le chemin de sauvegarde des sessions....1126
6.93.24 session_set_cookie_params() : Modifie les paramètres du cookie de session....1126
6.93.25 session_set_save_handler() : Configure les fonctions de stockage de sessions....1126
6.93.26 session_start() : Initialise une session....1128
6.93.27 session_unregister() : Supprime une variable de la session....1128
6.93.28 session_unset() : Détruit toutes les variables de session....1129
6.93.29 session_write_close() : Ecrit les données de session et ferme la session....1129
6.94 Mémoire partagée....1130
6.94.2 shmop_delete() : Détruit un bloc de mémoire partagée....1130
6.94.3 shmop_open() : Crée ou ouvre un bloc de mémoire partagée....1131
6.94.4 shmop_read() : Lit un bloc....1131
6.94.5 shmop_size() : Lire la taille du bloc de mémoire partagée....1132
6.94.6 shmop_write() : Ecrire dans un bloc de mémoire partagée....1132
6.95 Shockwave Flash....1133
6.95.1 Introduction....1133

Sommaire

6 Index des fonctions

6.95.2 Pré−requis....1133
6.95.3 Installation....1133
6.95.4 Configuration à l'exécution....1133
6.95.5 Types de ressources....1133
6.95.6 Constantes Prédefinies....1133
6.95.7 Exemples....1134
6.95.9 swf_actiongotoframe() : Joue un frame puis stoppe....1136
6.95.10 swf_actiongotolabel() : Affiche le frame nommé....1136
6.95.11 swf_actionnextframe() : Avance d'un frame....1136
6.95.12 swf_actionplay() : Joue l'animation flash à partir du frame courant....1137
6.95.13 swf_actionprevframe() : Recule d'un frame....1137
6.95.14 swf_actionsettarget() : Fixe le contexte des actions....1137
6.95.15 swf_actionstop() : Arrête l'animation flash....1137
6.95.16 swf_actiontogglequality() : Choisit le niveau de qualité haut ou bas....1137
6.95.17 swf_actionwaitforframe() : Ignore les actions si le frame n'est pas chargé....1137
6.95.18 swf_addbuttonrecord() : Contrôle la situation, l'aparance et la zone active du bouton courant....1138
6.95.19 swf_addcolor() : Fixe la couleur globale d'addition (? : the global add color)....1138
6.95.20 swf_closefile() : Ferme le fichier courant Shockwave Flash....1138
6.95.21 swf_definebitmap() : Définit une image bitmap....1139
6.95.22 swf_definefont() : Définit une police....1140
6.95.23 swf_defineline() : Définit une ligne....1140
6.95.24 swf_definepoly() : Définit un polygone....1140
6.95.25 swf_definerect() : Définit un rectangle....1140
6.95.26 swf_definetext() : Définit une chaîne de texte....1140
6.95.27 swf_endbutton() : Termine la définition du bouton courant....1140
6.95.28 swf_enddoaction() : Termine l'action courante....1141
6.95.29 swf_endshape() : Complète la définition de la forme courante....1141
6.95.30 swf_endsymbol() : Termine la définition de symbole....1141
6.95.31 swf_fontsize() : Change la taille de la police....1141
6.95.32 swf_fontslant() : Change l'inclinaison de la police courante....1141
6.95.33 swf_fonttracking() : Change l'espacement des caractères....1141
6.95.34 swf_getbitmapinfo() : Lit les informations sur une image....1142
6.95.35 swf_getfontinfo() : Retourne la hauteur du A majuscule, et du x minuscule....1142
6.95.36 swf_getframe() : Retourne le numéro de frame courant....1142
6.95.37 swf_labelframe() : Nomme le frame courant....1142
6.95.38 swf_lookat() : Définit une transformation de vue....1142
6.95.39 swf_modifyobject() : Modifie un objet....1143
6.95.40 swf_mulcolor() : Fixe la couleur globale de multiplication (? : the global multiply color)....1143
6.95.41 swf_nextid() : Retourne le prochain identifiant d'objet libre....1143
6.95.42 swf_oncondition() : Décrit une transition utilisée pour déclencher une liste d'actions....1143
6.95.43 swf_openfile() : Ouvre un nouveau fichier Shockwave Flash....1144
6.95.44 swf_ortho() : Définit une projection orthogonale entre les coordonnées utilisateur et le port courant....1144
6.95.45 swf_ortho2() : Définit une projection orthogonale à 2 dimensions entre les coordonnées utilisateur et le port courant....1144

Sommaire

6 Index des fonctions

6.95.46 swf_perspective() : Définit une projection orthogonale à 3 dimensions entre les coordonnées utilisateur et le port courant....1145
6.95.47 swf_placeobject() : Place un objet sur la scène....1145
6.95.48 swf_polarview() : Défini le point de vue de l'utilisateur en coordonnées polaire....1145
6.95.49 swf_popmatrix() : Dépile la matrice de transformation....1145
6.95.50 swf_posround() : Active l'approximation des translation d'objets....1146
6.95.51 swf_pushmatrix() : Empile la matrice de transformation courante dans la pile....1146
6.95.52 swf_removeobject() : Enlève un objet....1146
6.95.53 swf_rotate() : Rotation de la transformation courante....1146
6.95.54 swf_scale() : Homothétie....1146
6.95.55 swf_setfont() : Change la police courante....1146
6.95.56 swf_setframe() : Fixe le frame courant....1147
6.95.57 swf_shapearc() : Dessine une arc de cercle....1147
6.95.58 swf_shapecurveto() : Dessine une courbe de Bézier quadratique entre deux points....1147
6.95.59 swf_shapecurveto3() : Dessine une courbe Bézier cubique....1147
6.95.60 swf_shapefillbitmapclip() : Choisit le mode de remplissage par texture....1147
6.95.61 swf_shapefillbitmaptile() : Choisit le mode de remplissage par texture repétée....1147
6.95.62 swf_shapefilloff() : Inactive le remplissage....1148
6.95.63 swf_shapefillsolid() : Fixe la couleur pour le style courant de remplissage....1148
6.95.64 swf_shapelinesolid() : Fixe le style courant de ligne....1148
6.95.65 swf_shapelineto() : Dessine une ligne....1148
6.95.66 swf_shapemoveto() : Change la position courante....1148
6.95.67 swf_showframe() : Affiche le frame courant....1149
6.95.68 swf_startbutton() : Commence la définition d'un bouton....1149
6.95.69 swf_startdoaction() : Commence la description d'une liste d'action pour la frame courante....1149
6.95.70 swf_startshape() : Commence une forme complexe....1149
6.95.71 swf_startsymbol() : Définit un symbole....1149
6.95.72 swf_textwidth() : Retourne la longueur d'une chaîne....1149
6.95.73 swf_translate() : Translate la transformation courante....1150
6.95.74 swf_viewport() : Sélectionne une nouvelle zone pour un dessin ultérieur....1150
6.96 SNMP....1151
6.96.2 snmp_set_quick_print() : Ecrit la valeur courante de l'option quick_print de la librairie UCD....1151
6.96.3 snmpget() : Reçoit un objet SNMP....1152
6.96.4 snmprealwalk() : Retourne tous les objets, y compris leur ID d'objet....1152
6.96.5 snmpset() : Envoie un objet SNMP....1152
6.96.6 snmpwalk() : Reçoit tous les objets SNMP d'un agent....1153
6.96.7 snmpwalkoid() : Demande d'informations d'arbre sur une entité du réseau....1153
6.97 Socket....1155
6.97.1 Introduction....1155
6.97.2 Pré−requis....1155
6.97.3 Installation....1155
6.97.4 Configuration à l'exécution....1155
6.97.5 Types de ressources....1155

Sommaire

6 Index des fonctions

6.97.6 Constantes Prédefinies....1155
6.97.7 Socket Errors....1156
6.97.8 Exemples....1157
6.97.10 socket_bind() : Lie un nom à une socket....1158
6.97.11 socket_clear_error() : Efface toutes les erreurs précédemment générées par une socket....1158
6.97.12 socket_close() : Ferme une socket....1158
6.97.13 socket_connect() : Crée une connexion sur une socket....1159
6.97.14 socket_create_listen() : Ouvre une socket sur un port pour accepter les connexions....1159
6.97.15 socket_create_pair() : Crée une paire de socket identiques et les stocke dans fds....1160
6.97.16 socket_create() : Crée une socket (point de communication)....1160
6.97.17 socket_get_option() : Lit les options de socket....1161
6.97.18 socket_getpeername() : Interroge l'autre extrémité de la communication....1161
6.97.19 socket_getsockname() : Interroge la socket locale....1162
6.97.20 socket_iovec_add() : Ajoute un nouveau vecteur dans le tableau....1162
6.97.21 socket_iovec_alloc() : Construit un 'struct iovec'....1163
6.97.22 socket_iovec_delete() : Efface un vecteur....1163
6.97.23 socket_iovec_fetch() : Lit les données du vecteur specifié par iovec_id[iovec_position]....1163
6.97.24 socket_iovec_free() : Détruit le vecteur iovec specifié par iovec_id....1164
6.97.25 socket_iovec_set() : Modifie les données du vecteur iovec_id[iovec_position]....1164
6.97.26 socket_last_error() : Lit la dernière erreur générée par une socket....1164
6.97.27 socket_listen() : Attend une connexion (listen) sur une socket....1165
6.97.28 socket_read() : Lit des données d'une socket....1165
6.97.29 socket_readv() : Lit des données dans un fd, en utilisant le tableau scatter−gather définit par iovec_id....1166
6.97.30 socket_recv() : Reçoit des données d'une socket connectée....1166
6.97.31 socket_recvfrom() : Reçoit des données d'une socket, connectée ou pas....1166
6.97.32 socket_recvmsg() : Reçoit des messages sur uen socket, qu'elle soit orientée connexion ou pas....1167
6.97.33 socket_select() : Exécute l'appel système select() un tableau de socket avec une durée d'expiration....1167
6.97.34 socket_send() : Envoie des données à une socket connectée....1169
6.97.35 socket_sendmsg() : Envoie un message à une socket, qu'elle soit orientée connexion ou pas....1169
6.97.36 socket_sendto() : Envoie une message à une socket, qu'elle soit connectée ou pas....1169
6.97.37 socket_set_nonblock() : Sélectionne le mode non bloquant d'un descripteur de fichier....1170
6.97.38 socket_set_option() : Modifie les options de socket....1170
6.97.39 socket_shutdown() : Eteind une socket en lecture et/ou écriture....1170
6.97.40 socket_strerror() : Retourne une chaîne décrivant un message d'erreur....1171
6.97.41 socket_write() : Ecrit dans une socket....1171
6.97.42 socket_writev() : Ecrit des données dans un descripteur de fichier, fd, en utilisant le tableau scatter−gather définie par iovec_id....1172
6.98 Chaînes de caractères (Strings)....1173
6.98.2 addslashes() : Ajoute des anti−slashs dans une chaîne....1174
6.98.3 bin2hex() : Convertit des données binaires en représentation

Sommaire

6 Index des fonctions

hexadécimale....1175
6.98.4 chop() : Alias de rtrim....1175
6.98.5 chr() : Retourne un caractère spécifique....1175
6.98.6 chunk_split() : Scinde une chaîne....1175
6.98.7 convert_cyr_string() : Convertit une chaîne d'un jeu de caractères cyrillique à l'autre....1176
6.98.8 count_chars() : Retourne des statistiques sur les caractères utilisés dans une chaîne....1176
6.98.9 crc32() : Calculate la somme de vérification CRC32....1177
6.98.10 crypt() : Chiffrage indéchiffrable (hashing)....1177
6.98.11 echo() : Affiche une chaîne de caractères....1178
6.98.12 explode() : Coupe une chaîne en segments....1179
6.98.13 get_html_translation_table() : Returns the translation table used by htmlspecialchars and htmlentities....1180
6.98.14 get_meta_tags() : Extrait tous les méta tags d'un fichier....1181
6.98.15 hebrev() : Convertit un texte logique hébreux en texte visuel....1181
6.98.16 hebrevc() : Convertit un texte logique hébreux en texte visuel, avec retours à la ligne....1181
6.98.17 htmlentities() : Convertit tous les caractères éligibles en entités HTML....1182
6.98.18 htmlspecialchars() : Convertit les caractères spéciaux en entités HTML....1182
6.98.19 implode() : Rassemble les éléments d'un tableau en une chaîne....1183
6.98.20 join() : Rassemble les éléments d'un tableau en une chaîne....1184
6.98.21 levenshtein() : Calcule la distance Levenshtein entre deux chaînes....1184
6.98.22 localeconv() : Lit la configuration locale....1185
6.98.23 ltrim() : Supprime les caractères invisibles de début de chaîne....1187
6.98.24 md5_file() : Calcule le md5 d'un fichier....1187
6.98.25 md5() : Calcule le md5 d'une chaîne....1188
6.98.26 metaphone() : Calcule la clé métaphone....1188
6.98.27 nl_langinfo() : Rassemble des informations sur la langue et la configuration locale....1188
6.98.28 nl2br() : Insère des retours à la ligne HTML à chaque nouvelle ligne....1188
6.98.29 number_format() : Formate un nombre pour l'affichage....1189
6.98.30 ord() : Retourne le code ASCII d'un caractère....1189
6.98.31 parse_str() : Analyse une requête HTTP....1190
6.98.32 print() : Affiche une chaîne de caractères....1190
6.98.33 printf() : Affiche une chaîne de caractères formattée....1191
6.98.34 quoted_printable_decode() : Convertit une chaîne quoted−printable en chaîne 8 bits....1191
6.98.35 quotemeta() : Echappe les méta−caractères....1192
6.98.36 rtrim() : Supprime les espaces de fin de chaîne....1192
6.98.37 setlocale() : Modifie les informations de localisation....1193
6.98.38 similar_text() : Calcule la similarité de deux chaînes....1193
6.98.39 soundex() : Calcule la clé soundex....1194
6.98.40 sprintf() : Retourne une chaîne formattée....1194
6.98.41 sscanf() : Analyse une chaîne à l'aide d'un format....1196
6.98.42 str_pad() : Complète une chaîne jusqu'à une taille donnée....1197
6.98.43 str_repeat() : Répéte une chaîne....1197
6.98.44 str_replace() : Remplace toutes les occurrences dans une chaînes....1198
6.98.45 str_rot13() : Effectue une transformation rot13....1198
6.98.46 str_word_count() : Compte le nombre de mots utilisés dans une chaîne....1199
6.98.47 strcasecmp() : Comparaison de chaînes binaires....1199

Sommaire

6 Index des fonctions

6.98.48 strchr() : Trouve la première occurence d'un caractère dans une chaîne....1199
6.98.49 strcmp() : Comparaison binaire de chaînes....1200
6.98.50 strcoll() : Comparaison de chaînes localisée....1200
6.98.51 strcspn() : Trouve un segment de chaîne ne contenant pas certains caractères....1200
6.98.52 strip_tags() : Supprime les balises HTML et PHP d'une chaîne....1200
6.98.53 stripcslashes() : Supprime les anti−slash d'une chaîne C....1201
6.98.54 stripslashes() : Supprimer les anti−slash d'une chaîne....1201
6.98.55 stristr() : Trouve la première occurence dans une chaîne (insensible à la casse)....1201
6.98.56 strlen() : Calcule la taille d'une chaîne....1202
6.98.57 strnatcasecmp() : Comparaison de chaînes avec l'algorithme d'"ordre naturel" (insensible à la casse)....1202
6.98.58 strnatcmp() : Comparaison de chaînes avec l'algorithme d'"ordre naturel"....1202
6.98.59 strncasecmp() : Compare en binaire des chaînes de caractères....1203
6.98.60 strncmp() : Comparaison binaire des n premiers caractères....1204
6.98.61 strpos() : Trouve la position d'un caractère dans une chaîne....1204
6.98.62 strrchr() : Retourne la fin de la chaîne....1205
6.98.63 strrev() : Inverse une chaîne....1205
6.98.64 strrpos() : Trouve la position de la dernière occurence d'un caractère dans une chaîne....1206
6.98.65 strspn() : Trouve le premier segment de chaîne....1206
6.98.66 strstr() : Trouve la première occurence dans une chaîne....1207
6.98.67 strtok() : Coupe une chaîne en segments....1207
6.98.68 strtolower() : Make a string lowercase....1209
6.98.69 strtoupper() : Make a string uppercase....1209
6.98.70 strtr() : Remplace des caractères....1210
6.98.71 substr_count() : Compte de le nombre d'occurences de segments dans une chaîne....1210
6.98.72 substr_replace() : Remplace un segment dans une chaîne....1211
6.98.73 substr() : Retourne un segment de chaîne....1211
6.98.74 trim() : Supprime les espaces en début et fin de chaîne....1213
6.98.75 ucfirst() : Met le premier caractère en majuscule....1213
6.98.76 ucwords() : Met en majuscule la première lettre de tous les mots....1214
6.98.77 vprintf() : Affiche une chaîne formatée....1214
6.98.78 vsprintf() : Retourne une chaîne formatée....1215
6.98.79 wordwrap() : Effectue la césure d'une chaîne....1215
6.99 Sybase....1217
6.99.1 Introduction....1217
6.99.2 Pré−requis....1217
6.99.3 Installation....1217
6.99.4 Configuration à l'exécution....1217
6.99.5 Types de ressources....1218
6.99.6 Constantes Prédefinies....1218
6.99.8 sybase_close() : Ferme une connexion Sybase....1219
6.99.9 sybase_connect() : Ouvre une connexion à un serveur Sybase....1219
6.99.10 sybase_data_seek() : Déplace le pointeur interne de lignes....1220
6.99.11 sybase_fetch_array() : Retourne une ligne sous la forme d'un tableau....1220
6.99.12 sybase_fetch_field() : Lit les informations d'un champs....1220
6.99.13 sybase_fetch_object() : Retourne une ligne sous la forme d'un objet....1221

Sommaire

6 Index des fonctions

6.99.14 sybase_fetch_row() : Retourne une ligne sous la forme d'un tableau énuméré 1221
6.99.15 sybase_field_seek() : Modifie l'index d'un champs 1221
6.99.16 sybase_free_result() : Libère un résultat de la mémoire 1222
6.99.17 sybase_get_last_message() : Retourne le dernier message du serveur 1222
6.99.18 sybase_min_client_severity() : Fixe la sévérité minimale du client 1222
6.99.19 sybase_min_error_severity() : Fixe la sévérité minimale du client pour les erreurs 1222
6.99.20 sybase_min_message_severity() : Fixe la sévérité minimale du client pour les messages 1223
6.99.21 sybase_min_server_severity() : Fixe la sévérité minimale du client pour le serveur 1223
6.99.22 sybase_num_fields() : Retourne le nombre de champs dans un résultat 1223
6.99.23 sybase_num_rows() : Retourne le nombre de lignes dans un résultat 1223
6.99.24 sybase_pconnect() : Ouvre une connexion persistante à un serveur Sybase 1223
6.99.25 sybase_query() : Envoie une requête à une base Sybase 1224
6.99.26 sybase_result() : Lit une valeur dans un résultat 1224
6.99.27 sybase_select_db() : Sélectionne une base de données Sybase 1225
6.100 Analyseur de code PHP 1226
6.100.1 Introduction 1226
6.100.2 Configuration à l'exécution 1226
6.100.3 Constantes Prédefinies 1226
6.100.5 token_name() : Lit le nom d'un élément de code source 1228
6.101 URL 1229
6.101.1 base64_decode() : Décode une chaîne en MIME base64 1229
6.101.3 parse_url() : Analyse une URL et retourne ses composants 1229
6.101.4 rawurldecode() : Décode une chaîne URL 1229
6.101.5 rawurlencode() : Encode une chaîne en URL, selon la RFC1738 1229
6.101.6 urldecode() : Décode une chaîne encodée URL 1230
6.101.7 urlencode() : Encode une chaîne en URL 1230
6.102 Variables 1232
6.102.1 Introduction 1232
6.102.2 Pré−requis 1232
6.102.3 Installation 1232
6.102.4 Configuration à l'exécution 1232
6.102.5 Types de ressources 1232
6.102.6 Constantes Prédefinies 1232
6.102.8 empty() : Détermine si une variable est affectée 1233
6.102.9 floatval() : Convertit une chaîne en nombre à virgule flottante 1234
6.102.10 get_defined_vars() : Liste toutes les variables définies 1234
6.102.11 get_resource_type() : Retourne le type de ressource 1235
6.102.12 gettype() : Retourne le type de la variable 1235
6.102.13 import_request_variables() : Importe les variables de GET/POST/Cookie dans l'environnement global 1236
6.102.14 intval() : Retourne la valeur numérique (integer) de la variable 1236
6.102.15 is_array() : Détermine si une variable est un tableau 1237
6.102.16 is_bool() : Détermine si une variable est un tableau booléen 1237
6.102.17 is_callable() : Détermine si l'argument peut être appelé comme fonction 1237
6.102.18 is_double() : Détermine si une variable est de type double 1237
6.102.19 is_float() : Détermine si une variable est de type float 1237

Sommaire

6 Index des fonctions

6.102.20 is_int() : Détermine si une variable est de type integer....1238
6.102.21 is_integer() : Détermine si une variable est de type int....1238
6.102.22 is_long() : Détermine si une variable est de type integer....1238
6.102.23 is_null() : Indique si une variable est NULL....1238
6.102.24 is_numeric() : Détermine si une variable est un type numérique....1238
6.102.25 is_object() : Détermine si une variable est de type object....1239
6.102.26 is_real() : Détermine si une variable est de type real....1239
6.102.27 is_resource() : Détermine si une variable est une ressource....1239
6.102.28 is_scalar() : Indique si une variable est un scalaire....1239
6.102.29 is_string() : Détermine si une variable est de type string....1240
6.102.30 isset() : Détermine si une variable est affectée....1240
6.102.31 print_r() : Affiche des informations lisibles pour une variable....1241
6.102.32 serialize() : Linéarise une variable....1241
6.102.33 settype() : Affecte un type à une variable....1242
6.102.34 strval() : Retourne la valeur de la variable, au format chaîne....1242
6.102.35 unserialize() : Crée une variable PHP à partir d'une valeur linéarisée....1242
6.102.36 unset() : Détruit une variable....1243
6.102.37 var_dump() : Dumpe les informations d'une variable....1245
6.102.38 var_export() : Retourne le code PHP utilisé pour générer une variable....1245
6.103 vpopmail....1247
6.103.2 vpopmail_add_alias_domain() : Ajout un alias pour un domaine virtuel....1247
6.103.3 vpopmail_add_domain_ex() : Ajoute un nouvaeu domaine virtuel....1247
6.103.4 vpopmail_add_domain() : Ajoute un nouveau domaine virtuel....1248
6.103.5 vpopmail_add_user() : Ajoute un nouvel utilisateur à un domaine virtuel....1248
6.103.6 vpopmail_alias_add() : Ajoute un alias virtuel....1248
6.103.7 vpopmail_alias_del_domain() : Efface tous les alias virtuels d'un domaine....1248
6.103.8 vpopmail_alias_del() : Efface tous les alias d'un utilisateur....1249
6.103.9 vpopmail_alias_get_all() : Lit toutes les lignes d'alias d'un domaine....1249
6.103.10 vpopmail_alias_get() : Lit toutes les lignes d'un alias de domaine....1249
6.103.11 vpopmail_auth_user() : Valide le mot de passe d'un utilisateur pour un domaine. Retourne TRUE / FALSE....1250
6.103.12 vpopmail_del_domain_ex() : Efface un domaine virtuel....1250
6.103.13 vpopmail_del_domain() : Efface un domaine virtuel....1250
6.103.14 vpopmail_del_user() : Efface un utilisateur d'un domaine virtuel....1250
6.103.15 vpopmail_error() : Lit le dernier message d'erreur vpopmail....1251
6.103.16 vpopmail_passwd() : Remplace le mot de passe d'un utilisateur virtuel....1251
6.103.17 vpopmail_set_user_quota() : Modifie le quota d'un utilisateur virtuel....1251
6.104 API windows....1253
6.104.1 Introduction....1253
6.104.2 Pré−requis....1253
6.104.3 Installation....1253
6.104.4 Configuration à l'exécution....1253
6.104.5 Types de ressources....1253
6.104.6 Constantes Prédefinies....1253
6.104.7 Exemples....1254
6.104.9 w32api_init_dtype() : Crée une instance de type de données et la remplit...1255
6.104.10 w32api_invoke_function() : Appelle une fonction windows 32....1255
6.104.11 w32api_register_function() : Enregistre une fonction win32 dans PHP....1255
6.104.12 w32api_set_call_method() : Modifie le nom de la méthode appelée....1256
6.105 WDDX....1257

Sommaire

6 Index des fonctions

6.105.2 wddx_deserialize() : Lire un paquet WDDX....1258
6.105.3 wddx_packet_end() : Clos un paquet WDDX....1258
6.105.4 wddx_packet_start() : Commencer un nouveau paquet WDDX avec une structure....1258
6.105.5 wddx_serialize_value() : Enregistrer une valeur dans un paquet WDDX....1258
6.105.6 wddx_serialize_vars() : Enregistrer plusieurs valeurs dans un paquet WDDX....1258
6.106 Analyseur syntaxique XML....1260
6.106.1 Introduction....1260
6.106.2 Pré−requis....1260
6.106.3 Installation....1260
6.106.4 Configuration à l'exécution....1260
6.106.5 Types de ressources....1261
6.106.6 Constantes Prédefinies....1261
6.106.7 gestionnaires d'évènements....1261
6.106.8 Problèmes de casse....1262
6.106.9 Codes d'erreurs....1262
6.106.10 Codage des caractères....1263
6.106.11 Quelques exemples....1263
6.106.13 utf8_encode() : Convertit une chaîne ISO−8859−1 en UTF−8....1268
6.106.14 xml_error_string() : Lit le message d'erreur de l'analyseur XML....1269
6.106.15 xml_get_current_byte_index() : Retourne l'index de l'octet courant d'un analyseur XML....1269
6.106.16 xml_get_current_column_number() : Retourne le nombre courant de colonne d'un analyseur XML....1269
6.106.17 xml_get_current_line_number() : Retourne le numéro de ligne courant d'un analyseur XML....1269
6.106.18 xml_get_error_code() : Retourne le nombre courant de colonne d'un analyseur XML....1270
6.106.19 xml_parse_into_struct() : Analyse une structure XML....1270
6.106.20 xml_parse() : Commence l'analyse d'un fichier XML....1273
6.106.21 xml_parser_create() : Création d'un analyseur XML....1274
6.106.22 xml_parser_free() : Détruit un analyseur XML....1274
6.106.23 xml_parser_get_option() : Lit les options d'un analyseur XML....1274
6.106.24 xml_parser_set_option() : Affecte les options d'un analyseur XML....1275
6.106.25 xml_set_character_data_handler() : Affecte les gestionnaires de caractère bruts....1275
6.106.26 xml_set_default_handler() : Affecte le gestionnaire par défaut....1276
6.106.27 xml_set_element_handler() : Affecte les gestionnaires de début et de fin....1276
6.106.28 xml_set_external_entity_ref_handler() : Modifie le gestionnaire de référence externes....1277
6.106.29 xml_set_notation_decl_handler() : Affecte les gestionnaires de notation....1278
6.106.30 xml_set_object() : Utilise un analyseur XML à l'intérieur d'un objet....1279
6.106.31 xml_set_processing_instruction_handler() : Affecte les gestionnaires d'instructions exécutables....1279
6.106.32 xml_set_unparsed_entity_decl_handler() : Affecte les gestionnaires d'entité non déclaré....1280
6.107 XML−RPC....1282
6.107.1 Introduction....1282
6.107.2 Pré−requis....1282

Sommaire

6 Index des fonctions

6.107.3 Installation....1282
6.107.4 Configuration à l'exécution....1282
6.107.5 Types de ressources....1282
6.107.6 Constantes Prédefinies....1282
6.107.8 xmlrpc_decode() : Décode le code XML en types PHP natifs....1283
6.107.9 xmlrpc_encode_request() : Génère le code XML d'une requête de méthode....1283
6.107.10 xmlrpc_encode() : Génère le code XML pour une valeur PHP....1283
6.107.11 xmlrpc_get_type() : Retourne le type XMLRPC d'une valeur PHP. Particulièrement pratique pour les types chaînes de type base64 et datetime....1284
6.107.12 xmlrpc_parse_method_descriptions() : Décode le code XML en une liste de descriptions de méthodes....1284
6.107.13 xmlrpc_server_add_introspection_data() : Ajoute des données d'introspection....1284
6.107.14 xmlrpc_server_call_method() : Analyse une requête XML est appelle les méthodes associées....1285
6.107.15 xmlrpc_server_create() : Crée un serveur XMLRPC....1285
6.107.16 xmlrpc_server_destroy() : Détruit un serveur XMLRPC....1285
6.107.17 xmlrpc_server_register_introspection_callback() : Enregsitre une fonction PHP pour générer la documentation....1286
6.107.18 xmlrpc_server_register_method() : Enregistre une fonction PHP avec une méthode....1286
6.107.19 xmlrpc_set_type() : Modifie le type XMLRPC, base64 ou datetime, pour une chaînes de caractères PHP....1286
6.108 XSLT....1287
6.108.1 Introduction....1287
6.108.2 Pré−requis....1287
6.108.3 Installation....1287
6.108.4 Configuration à l'exécution....1287
6.108.5 Types de ressources....1287
6.108.6 Constantes Prédefinies....1288
6.108.8 xslt_errno() : Retourne le numéro d'erreur courant....1288
6.108.9 xslt_error() : Retourne le message d'erreur courant....1288
6.108.10 xslt_free() : Détruit un analyseur XSLT....1289
6.108.11 xslt_process() : Transforme des données XML....1289
6.108.12 xslt_set_sax_handler() : Modifie les gestionnaires SAX de l'analyseur XSLT....1290
6.109 YAZ....1291
6.109.1 Introduction....1291
6.109.2 Installation....1291
6.109.3 Configuration à l'exécution....1291
6.109.4 Types de ressources....1291
6.109.5 Constantes Prédefinies....1292
6.109.6 Exemples....1292
6.109.8 yaz_ccl_conf() : Configure l'analyseur CCL....1293
6.109.9 yaz_ccl_parse() : Appelle l'analyseur CCL....1293
6.109.10 yaz_close() : Ferme une connexion YAZ....1294
6.109.11 yaz_connect() : Prépare une connexion à un hôte YAZ....1294
6.109.12 yaz_database() : Spécifie la base d'une session....1294
6.109.13 yaz_element() : Spécifie le type d'éléments à lire....1294
6.109.14 yaz_errno() : Retourne le numéro d'erreur....1295

Sommaire

6 Index des fonctions

6.109.15 yaz_error() : Retourne une description de l'erreur....1295
6.109.16 yaz_hits() : Retourne le nombre de résultat de la dernière recherche....1295
6.109.17 yaz_itemorder() : Prépare une requête Z39.50 Item Order avec le package ILL−Request....1295
6.109.18 yaz_present() : Prépare à la lecture (Z39.50 present)....1297
6.109.19 yaz_range() : Spécifie le nombre maximal de résultat à lire....1297
6.109.20 yaz_record() : Retourne un résultat....1297
6.109.21 yaz_scan_result() : Retourne le résultat d'un scan....1298
6.109.22 yaz_scan() : Prépare un scan....1298
6.109.23 yaz_search() : Prépares une recherche....1299
6.109.24 yaz_sort() : Configure les critères de tri....1299
6.109.25 yaz_syntax() : Spécifie la syntaxe de lecture des lignes....1299
6.109.26 yaz_wait() : Attend l'exécution d'une requête....1300
6.110 NIS....1301
6.110.1 Introduction....1301
6.110.2 Installation....1301
6.110.3 Constantes Prédefinies....1301
6.110.5 yp_cat() : Retourne un tableau contenant toute la carte....1302
6.110.6 yp_err_string() : Retourne la chaîne d'erreur associée à la dernière opération....1302
6.110.7 yp_errno() : Retourne le code d'erreur de la dernière opération....1302
6.110.8 yp_first() : Retourne le premier couple (clé ; valeur) d'une carte donnée....1303
6.110.9 yp_get_default_domain() : Retourne le domaine NIS par défaut....1303
6.110.10 yp_master() : Retourne le nom de la machine maître pour une carte....1303
6.110.11 yp_match() : Retourne la ligne associée....1304
6.110.12 yp_next() : Retourne le couple (clé ; valeur) suivant d'une carte donnée....1304
6.110.13 yp_order() : Retourne le numéro d'ordre d'une carte....1305
6.111 Accès aux fichiers Zip (Lecture seule)....1306
6.111.1 Introduction....1306
6.111.2 Pré−requis....1306
6.111.3 Installation....1306
6.111.4 Configuration à l'exécution....1306
6.111.5 Types de ressources....1306
6.111.6 Constantes Prédefinies....1306
6.111.7 Exemples....1306
6.111.9 zip_entry_close() : Ferme un dossier d'archive....1307
6.111.10 zip_entry_compressedsize() : Lit la taille compressée d'un dossier d'archive....1308
6.111.11 zip_entry_compressionmethod() : Lit la méthode de compression utilisée sur un dossier d'archive....1308
6.111.12 zip_entry_filesize() : Lit la taille décompressée d'un dossier d'archive....1308
6.111.13 zip_entry_name() : Lit le nom du dossier d'archive....1308
6.111.14 zip_entry_open() : Ouvre un fichier d'archive en lecture....1309
6.111.15 zip_entry_read() : Lit le contenu d'un fichier dans un dossier....1309
6.111.16 zip_open() : Ouvre une archive ZIP....1309
6.111.17 zip_read() : Lit la prochaine entrée dans une archive ZIP....1310
6.112 Compression Zlib....1311
6.112.1 Introduction....1311
6.112.2 Pré−requis....1311
6.112.3 Configuration à l'exécution....1311

Sommaire

6 Index des fonctions
6.112.4 Types de ressources....1312
6.112.5 Constantes Prédefinies....1312
6.112.6 Exemples....1312
6.112.8 gzcompress() : Compresse une chaîne....1313
6.112.9 gzdeflate() : Compresse une chaîne avec la méthode DEFLATE....1314
6.112.10 gzencode() : Crée une chaîne compressée....1314
6.112.11 gzeof() : Indique si la fin d'un fichier compressé est atteinte....1315
6.112.12 gzfile() : Lit la totalité d'un fichier compressé....1315
6.112.13 gzgetc() : Lit un caractère dans un fichier compressé....1315
6.112.14 gzgets() : Lit une ligne dans un fichier compressé....1316
6.112.15 gzgetss() : Lit une ligne dans un ficheir compressé, et supprime les balises....1316
6.112.16 gzinflate() : Décompresse une chaîne....1316
6.112.17 gzopen() : Ouvre un fichier compressé avec gzip....1316
6.112.18 gzpassthru() : Affiche le contenu d'un fichier en une seule fois....1317
6.112.19 gzputs() : Ecrit une chaîne dans un fichier compressé....1317
6.112.20 gzread() : Lecture de fichier compressé binaire....1318
6.112.21 gzrewind() : Recommence la lecture du fichier....1318
6.112.22 gzseek() : Déplace le pointeur de lecture....1318
6.112.23 gztell() : Lit la position courante du pointeur de lecture....1319
6.112.24 gzuncompress() : Décompresse une chaîne compressée....1319
6.112.25 gzwrite() : Ecrit dans un fichier compressé, en gzip....1319
6.112.26 readgzfile() : Lit tout le fichier compressé....1319

7 Streams API for PHP Extension Authors....1323
7.1 Overview....1323
7.2 Streams Basics....1323

8 FAQ : Foire Aux Questions....1325
8.1 General Information....1325
8.1.1 faq.general.what....1325
8.2.1 faq.mailinglist.isthere....1325
8.3.1 faq.obtaining.where....1325
8.4.1 faq.databases.mssql....1325
8.5.1 faq.installation.phpinior' −−!>....1325
8.5.10 faq.install.phpandiis....1326
8.6.1 faq.build.configurescript Red Hat ships is broken. Look for this line:....1326
8.6.13 faq.build.microtime....1326
8.7.1 faq.using.anyform....1326
8.8.1 faq.html.encoding 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:....1326
8.9 PHP and COM....1326
8.9.1 faq.com.q1....1327
8.10.1 faq.languages.asp....1327
8.11.1 faq.migrating.phpfi....1327
8.12.1 faq.migration4.php34....1327
8.13.1 faq.misc.popup....1327

Sommaire

9 Appendices....1328
9.1 Histoire de PHP....1328
9.1.1 Histoire de PHP....1328
9.1.2 Quelques projets liés à PHP....1329
9.1.3 Livres traitant de PHP....1331
9.1.4 Publications about PHP....1331
9.2 Migration de PHP 3.0 à PHP 4.0....1331
9.2.1 Ce qui a changé en PHP 4.0....1331
9.2.2 Utiliser PHP 3 et PHP 4 simultanément....1331
9.2.3 Migration des fichiers de configuration....1332
9.2.4 Comportement de l'analyseur....1333
9.2.5 Rapport d'erreur....1333
9.2.6 Initialiseur....1335
9.2.7 empty("0")....1335
9.2.8 Fonctions manquantes....1335
9.2.9 Extensions PHP 3.0....1336
9.2.10 Substitution de variables dans les chaînes....1336
9.2.11 Cookies....1336
9.3 Migration de PHP/FI 2.0 à PHP 3.0....1336
9.3.1 A propos des incompatibilités en 3.0....1336
9.3.2 Balises PHP....1337
9.3.3 Syntaxe if..endif....1338
9.3.4 Syntaxe while....1338
9.3.5 Types d'expression....1339
9.3.6 Les messages d'erreur ont changé....1339
9.3.7 Evaluation rapide des booléens....1339
9.3.8 L TRUE / FALSE....1340
9.3.9 Diverses incompatibilités....1340
9.4 Débuggeur PHP....1341
9.4.1 A propos du débuggeur....1341
9.4.2 Utiliser le débuggeur PHP....1341
9.4.3 Protocole du débuggeur....1341
9.5 Développement PHP....1342
9.5.1 Créer une fonction PHP 3....1343
9.5.2 Appeler des fonctions utilisateurs....1351
9.5.3 Rapport d'erreurs....1352
9.6 Liste d'alias....1353
9.7 Mot réservés en PHP....1360
9.7.1 Liste de mots−clé....1360
9.7.2 Variables prédéfinies....1362
9.7.3 Classes prédéfinies....1367
9.8 Types des ressources PHP....1368
9.9 List of Supported Protocols/Wrappers....1380
9.9.1 HTTP and HTTPS....1380
9.9.2 FTP and FTPS....1381
9.9.3 PHP input/output streams....1381
9.9.4 Compression Streams....1382
9.10 Liste des token de l'analyseur....1382
9.11 A propos du manuel....1385
9.11.1 Formats....1385
9.11.2 A propos des notes utilisateurs....1386
9.11.3 Où trouver plus d'information sur PHP?....1386

Sommaire

9 Appendices

9.11.4 Comment aider à l'amélioration de la documentation....1386
9.11.5 Comment sont générées les documentations....1387
9.11.6 Traductions....1387

10 Index....1388

10.1 Index des fonctions....1388
10.2 Index des concepts....1436
10.3 Index des exemples....1484

Informations sur l'ouvrage

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 : 14/11/2002 06:11:29

1.2 L'éditeur

- Damien Seguy

© **1997**, **1998**, **1999**, **2000**, **2001**, **2002**, *PHP Documentation Group* ,

1.3 Note légale

Copyright

Ce manuel est © Copyright 1997, 1998, 1999, 2000, 2001, 2002 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.

2 Préface

Introduction

2.1 A propos de ce manuel

Ce manuel est écrit en XML avec DocBook XML DTD , en utilisant DSSSL (Document Style and Semantics Specification Language) pour l'affichage. Les outils utilisés pour les formats HTML et TEX sont Jade , écrit par James Clark et The Modular DocBook Stylesheets écrit par Norman Walsh . Nous utilisons aussi Microsoft HTML Help Workshop pour générer le format HTML.

Vous pouvez télécharger le manuel courant dans divers langages et formats, y compris en texte seul, HTML , PDF , PalmPilot DOC, PalmPilot iSilo et WinHelp, depuis http://www.php.net/docs.php . Les manuels sont mis à jour quotidiennement.

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, grâce à une version adaptée de texi .

Vous pouvez avoir d'autres informations sur le téléchargement des sources XML de cette documentation à http://cvs.php.net/ . La documentation est stockée dans le module phpdoc .

Si vous rencontrez des erreurs dans ce manuel, rapportez les dans le système de rapport de bug : http://bugs.php.net/ , et classez les comme des "Problème de documentation" (en anglais, "Documentation Problem"). De cette façon, nous pourrons être prévenu de chaque bug, et nous pourrons les corriger.

Pour les problèmes liés à la traduction, vous pouvez aussi contacter directement l'éditeur de ce manuel damien.seguy@nexen.net .

3 Comment commencer

3.1 Introduction

3.1.1 Qu'est ce que PHP?

PHP (officiellement "PHP: Hypertext Preprocessor") est un langage de script HTML, qui fonctionne côté serveur.

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

Exemple d'introduction

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

Il est à noter la différence avec les autres scripts CGI écrit 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 navigateur 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.

3.1.2 Que peut faire PHP?

Le langage PHP possède les même fonctionnalités que les autres langages permettant d'écrire des scripts CGI, comme collecter des données, générer dynamiquement des pages web ou bien envoyer et recevoir des cookies.

La plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données. Réaliser une page web dynamique interfacant une base de donnés est extrêmement simple. Les bases de données suivantes sont supportées par PHP:

- Adabas D
- dBase
- Empress

- FilePro (lecutre seule)
- Hyperwave
- IBM DB2
- Informix
- Ingres
- InterBase
- FrontBase
- mSQL
- Direct MS−SQL
- MySQL
- ODBC
- Oracle (OCI7 et OCI8)
- Ovrimos
- PostgreSQL
- Sesam
- Solid
- Sybase
- Velocis
- Unix dbm

Le langage PHP inclus le support des services utilisant les protocoles tels que `IMAP` , `SNMP` , `NNTP` , `POP3` ou encore `HTTP` . Vous pouvez également ouvrir des connexions et interagir en utilisant d'autres protocoles.

3.1.3 La genèse du PHP

Le langage PHP a été conçu durant l'automne 1994 par Rasmus Lerdorf. Les premières versions (qui restèrent privées) étaient utilisées afin de savoir qui venait consulter son CV en ligne. La première version publique fut disponible au début de l'année 1995. Elle fut connue sous le nom de "Personal Sommaire Page Tools". Elle était composée d'un analyseur extrêmement simple qui ne reconnaissait que quelques macros spéciales et d'un petit nombre d'utilitaires couramment utilisés dans les pages web. Un livre d'or, un compteur, etc... L'analyseur fut réécrit durant l'été 1995 et fut appelé PHP/FI Version 2. FI etaient les initiales d'un autre package que Rasmus avait écrit qui interprétait les formulaires HTML. C'est alors qu'il combina le "Personnal Sommaire Page tools" avec le "Form Interpreter" et il y ajouta le support de mSQL: c'est comme cela que naquît PHP/FI. PHP/FI grandit de manière spectaculaire et de nombreuses personnes commencèrent à contribuer à son amélioration.

Il est relativement peu aisé de donner des statistiques, mais on estime que PHP/FI est utilisé sur 15 000 sites web dans le monde entier, fin 1996. Ce chiffre atteint 50 000 durant l'été 1997. L'été 1997 voit aussi un profond changement dans le développement du PHP: d'un projet personnel (celui de Ramsus), on passe alors à une projet d'équipe. L'analyseur fut de nouveau réécrit par Zeev Suraski et Andi Gutmans et ce nouvel analyseur forma la base de la version 3 du PHP. Une grande partie du code de PHP/FI fut complètement réécrit alors que l'autre partie fut portée pour donner le PHP Version 3. La dernière version de PHP (PHP 4) utilise le moteur d'analyse Zend pour atteindre de nouveaux niveaux de performance, et supporter un nombre encore plus grand de librairies et extensions. Il tourne de manière native sur tous les serveurs web les plus répandus.

Aujourd'hui (Janvier 2001) PHP 3 ou PHP 4 sont distribués avec de nombreux produits commerciaux comme "C2's StrongHold web server" et "RedHat Linux" et il est admis (d'après les chiffres de NetCraft , et leurs statistiques Netcraft Web Server Survey) que le PHP est utilisé sur 5 100 000 sites web dans le monde entier. Pour comparaison, ce chiffre est légèrement supérieur au nombre de serveurs tournant sous Microsoft Information server (IIS) : 5.03 millions.

Enfin, à l'heure où ce document est rédigé, la nouvelle génération du PHP est en cours de création. Elle utilisera les qualités de Zend pour améliorer les performances et améliorera le support des serveurs web autres que Apache.

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 coté 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 lequels vous allez utiliser des balises magiques, qui feront beaucoup de choses pour vous.

3.2.2 Votre première page PHP

Créez un fichier appelé `bonjour.php` dans votre dossier webm 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>
```

Notez que cela n'est pas un script CGI. Ce fichier 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 surgit 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.

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 commadnes 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.

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 interesse ici est is `$_SERVER["HTTP_USER_AGENT"]` .

Note
Attention pour les auto−globales de PHP $_SERVER est une variable spéciale de PHP, qui contient toutes les informations relatives au serveur web. C'est une variable réservée de PHP, et une auto−globale. Reportez−vous aux pages du manuel traitant des Auto−globales (aussi connues sous le noms de super−globales). Ces variables spéciales ont été introduites en PHP 4.1.0. Auparavant, il fallait utiliser les variables `$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"]; ?>
```

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 qui 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 afficahge. 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 />";
}
?>
```

Ici, nous introduisont plusieurs nouveaux concepts. Nous avons une structure if . Si vous êtes familier avec les syntaxes de base du langage C, cela ne vous surprendra. 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 trouvez 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
}
?>
```

Au lieu d'utilise une commande `echo` , pour afficher du texte, vous pouvez utiliser du code HTML pur. Le point important a noter ici et 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'examples 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.
```

Le fonctionnement du script devrait être évident. Il n'y a rien de plus compliqué. Les variables `$_POST["nom"]` et `$_POST["age"]` sont automatiquement créés par PHP. Un peut 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 **method** . 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 pluspart, les dévelopeurs 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 introduit en PHP 4.1.0. Ce sont les variables suivantes : $_GET , $_POST , $_COOKIE , $_SERVER , $_ENV , $_REQUEST et $_SESSION . Les vieux tableaux $HTTP_*_VARS , tels que $HTTP_POST_VARS existent toujours depuis PHP 3.
- 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 .

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. Vous pouvez trouver une liste de ces compagnies à http://hosts.php.net/ .

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 (librairie `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 bloqué, 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 base 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.

Instructions d'installation pour HP-UX 10

```
From: paul_mckay@clearwater-it.co.uk
```

04-Jan-2001 09:49

(Ces conseils sont destinés à PHP 4.0.4 et Apache v1.3.9)
Vous voulez installer PHP et Apache sur une HP-UX 10.20?

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.dep
, 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, décompressez leur,
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 instal

Si vous avez des erreurs à cette étape, c'est probablement par ce 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 puis faites "tar -xvf" avec les deux archives.
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 pa
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'éxécuter Apache avec l'utilisateur nobody sous
HP-UX. A partir de ce moment là, PHP et Apache doivent fonctionner.

J'espère que cela aidera quelqu'un.

Paul Mckay.
```

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 .

3.3.6.2 Utiliser port tree

Ceci est la méthode recommandée d'installation de PHP sous OpenBSD, car elle prend en compte les dernières modifications et mises à jour de sécurité. Pour utiliser cette méthode, assurez vous que vous avez bien ports tree récent. Choisissez alors simplement la version que vous souhaitez installer, et utilisez la commande `make install` . Ci−dessous, voici un exemple.

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 le manuel Unix page packages(7) pour plus de détails sur les packages binaires d'OpenBSD.

3.3.6.2 Utiliser port tree

Vous pouvez aussi compiler PHP depuis les sources avec ports tree . Cependant, ce n'est recommandé qu'aux utilisateurs familiers avec OpenBSD. PHP4 est réparti en trois sous dossiers : core, extensions et pear. Les dossiers d'extensions génèrent des sous dossiers pour tous les modules PHP. Si vous ne souhaitez pas créer ces modules, utilisez la commande `no_*` FLAVOR. Par exemple, pour inactiver le module IMAP, ajoutez à FLAVOR : `no_imap` .

3.3.6.3 Versions plus anciennes

Les anciennes versions de OpenBSD utilisait 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 touours 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 Unix/Mac OS X installs

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

3.3.7.1 Using Packages

Il existe quelques versions pré−packagées et pré−compilée de PHP pour MacOSX. Cela être utile pour mettre en place une configuration standard, mais si vous avez besoin d'accéder a des fonctionnalités spécifiques (comme un serveur sécurisé, ou un pilote de base de données exotique), 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
```

D'autres exemples pour le client Mac OS X et le serveur Mac OS X sont disponibles chez Stepwise .

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 diese (#), 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 complète des options de configuration

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
- XML

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, certains 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 Database options

--with-db
: Inclut le support de xDBM (obsolète).

--enable-dba=shared
: Compile DBA comme module partagé.

--with-gdbm[=DIR]
: Inclut le support GDBM.

--with-ndbm[=DIR]
: Inclut le support NDBM.

--with-db2[=DIR]
: Inclut le support Berkeley DB2.

--with-db3[=DIR]
: Inclut le support Berkeley DB3.

--with-dbm[=DIR]
: Inclut le support DBM.

--with-cdb[=DIR]
: Inclut le support CDB.

--enable-dbase
: Active la librairie dbase fournie dans la distribution.

--with-dbplus
: Inclut le support dbplus.

--enable-dbx
: Enable dbx.

--with-fbsql[=DIR]
: Inclut le support FrontBase. DIR est le dossier d'installation de FrontBase.

--enable-filepro
: Active le support de la librairie filePro (lecture seule). Cette librairie fait partie de la distribution.

--with-fribidi[=DIR]
: Inclut le suport de fribidi (requiert FriBidi >=0.1.12). DIR DIR est le dossier d'installation de fribidi. Par défaut, c'est /usr/local/.

--with-informix[=DIR]
: Inclut le support Informix. DIR est le dossier d'installation d'Informix. Par défaut, c'est rien du tout.

--with-ingres[=DIR]
: Inclut le support Ingres II. DIR est le dossier d'installation d'Ingres. Par défaut, c'est /II/ingres.

--with-interbase[=DIR]
: Inclut le support InterBase. DIR est le dossier d'installation d'InterBase. Par défaut, c'est /usr/interbase.

--with-msql[=DIR]
: Inclut le support mSQL. DIR est le dossier d'installation de mSQL. Par défaut, c'est /usr/local/Hughes.

--with-mysql[=DIR]
: Inclut le support MySQL. DIR est le dossier d'installation de MySQL. Si DIR n'est pas spécifié, la librairie MySQL inclue dans la distribution sera utilisée.

--with-oci8[=DIR]
: Inclut le support Oracle-oci8. DIR est le dossier d'installation de Oracle 8. Par défaut, c'est ORACLE_HOME.

--with-adabas[=DIR]
: Inclut le support Adabas D. DIR est le dossier d'installation d'Adabas D. Par défaut, c'est /usr/local.

--with-sapdb[=DIR]

Inclut le support SAP DB. DIR est le dossier d'installation de SAP DB. Par défaut, c'est /usr/local.

--with-solid[=DIR]
Inclut le support Solid. DIR est le dossier d'installation de Solid. Par défaut, c'est /usr/local/solid.

--with-ibm-db2[=DIR]
Inclut le support IBM DB2. DIR est le dossier d'installation de DB2. Par défaut, c'est /home/db2inst1/sqllib.

--with-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=-lunix CUSTOM_ODBC_LIBS="-ldblib -lodbc".

--with-iodbc[=DIR]
Inclut le support iODBC. DIR est le dossier d'installation de iODBCBirdstep. 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).

--with-oracle[=DIR]
Inclut le support Oracle-oci7. DIR est le dossier d'installation Oracle 7. Par défaut, c'est ORACLE_HOME.

--with-ovrimos[=DIR]
Inclut le support Ovrimos SQL Server. DIR est le dossier d'installation Ovrimos. Par défaut, c'est le dossier d'installation de libsqlcli.

--with-pgsql[=DIR]
Inclut le support PostgreSQL. DIR est le dossier d'installation PostgreSQL. Par défaut, c'est /usr/local/pgsql. Utilisez DIR=shared pour compiler PostGreSQL comme librairie partagée, ou bien DIR=shared,chemin pour compiler PostGreSQL comme librairie partagée tout en en précisant le chemin.

--with-sybase[=DIR]

Inclut le support Sybase−DB. DIR est le dossier d'installation Sybase. Par défaut, c'est /home/sybase.

−−with−sybase−ct[=DIR]
Inclut le support Sybase−CT. DIR DIR est le dossier d'installation Sybase−CT. Par défaut, c'est /home/sybase.

−−disable−unified−odbc
Inactive le support de l'ODBC unifié. Uniquement utilisable si iODBC, Adabas, Solid, Velocis ou une interface ODBC spéciale st activée. Valable uniquement pour PHP 3.

3.3.8.1.2 Graphics options

−−with−gd[=DIR]
Inclut le support GD. DIR est le dossier d'installation de la librairie GD. Par défaut, c'est /II/ingres. Utilisez DIR=shared pour compiler GD comme librairie partagée, ou bien DIR=shared,chemin pour compiler PostGreSQL comme librairie partagée tout en en précisant le chemin.

−−enable−gd−native−ttf
GD: Active la fonction de chaîne écrites en TrueType pour GD.

−−with−jpeg−dir=DIR
GD: Configure le chemin jusqu'à la librairie libjpeg.

−−with−png−dir=DIR
GD: Configure le chemin jusqu'à la librairie libpng.

−−with−xpm−dir=DIR
GD: Configure le chemin jusqu'à la librairie libXpm.

−−with−freetype−dir=DIR
GD: Configure le chemin jusqu'à la librairie freetype2.

−−with−ttf[=DIR]
GD: Inclut le support FreeType 1.x.

−−with−t1lib[=DIR]
GD: Inclut le support T1lib.

−−with−cpdflib[=DIR]
Inclut le support de cpdflib (requiert cpdflib >= 2). DIR est le dossier d'installation de la librairie cpdflib. Par défaut, c'est /usr.

−−with−jpeg−dir[=DIR]
cpdflib 2.x: Configure le chemin jusqu'à la librairie freetype2.

−−with−tiff−dir[=DIR]
cpdflib 2.x: Configure le chemin jusqu'à la librairie TIFF.

−−with−pdflib[=DIR]
Inclut le support PDFlib. DIR DIR est le dossier d'installation de la librairie pdflib. Par défaut, c'est /usr/local. Utilisez DIR=shared pour compiler PDFlib comme librairie partagée, ou bien DIR=shared,chemin pour compiler PostGreSQL comme librairie partagée tout en en précisant le chemin.

−−with−jpeg−dir[=DIR]
PDFLIB: Configure le chemin jusqu'à la librairie libjpeg.

−−with−png−dir[=DIR]
PDFLIB: Configure le chemin jusqu'à la librairie libpng.

−−with−tiff−dir[=DIR]
PDFLIB: Configure le chemin jusqu'à la librairie libtiff.

−−with−swf[=DIR]
Inclut le support swf (ShockWave Flash).

−−without−gd
Inactive le support GD. Pour PHP 3 seulement!

−−with−imagick

Inclut le support ImageMagick. DIR est le dossier d'installation de la librairie ImageMagick. Par défaut, PHP essaiera de la trouver par lui−même. Pour PHP 3 seulement!

−−with−ming[=DIR]
Inclut le support ming.

3.3.8.1.3 Misc options

−−enable−force−cgi−redirect
Active la vérification de sécurité des redirections internes du serveur. Utilisez cette configuration si vous utilisez PHP comme CGI avec Apache.

−−enable−discard−path
Si cette option est activée, l'exécutable PHP peut être placés hors de l'arborescence web, afin que le fichier .htaccess ne soit pas contournable.

−−with−fastcgi=SRCDIR
Compile PHP comme application FastCGI.

−−enable−debug
Compile PHP avec les symbole 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.

−−with−openssl[=DIR]
Inclut le support de OpenSSL (requiert OpenSSL >= 0.9.5).

−−enable−sigchild
Active le gestionnaire de SIGCHLD propre à PHP.

−−disable−rpath
Inactive 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 flôts de PHP (expérimental). Ne pas utiliser a moins que vous ne testiez le code.

−−with−zlib−dir=<DIR>
Définit le chemin de the location of zlib install directory.

−−with−zlib[=DIR]
Inclut le support de zlib (requiert zlib >= 1.0.9). DIR est le dossier d'installation de zlib.

−−with−aspell[=DIR]
Inclut le support d'ASPELL.

−−enable−bcmath
Active la librairie bcmath, pour le support des nombres de taille arbitraire.

−−with−bz2[=DIR]
Inclut le support de BZip2.

−−enable−calendar
Active le support des conversions calendaires.

−−with−ccvs[=DIR]
Inclut le support de CCVS.

−−with−crack[=DIR]
Inclut le support de Crack.

--enable-ctype
: Active le support de ctype.

--with-curl[=DIR]
: Inclut le support de CURL.

--with-cybercash[=DIR]
: Inclut le support de CyberCash. DIR est le dossier d'installation de CyberCash MCK.

--with-cybermut[=DIR]
: Inclut le support de CyberMut (télépaiement du Crédit Mutuel).

--with-cyrus
: Inclut le support de cyrus IMAP.

--enable-exif
: Active le support de exif.

--with-fdftk[=DIR]
: Inclut le support de fdftk.

--enable-ftp
: Active le support du FTP.

--with-gettext[=DIR]
: Inclut le support de GNU gettext. DIR est le dossier d'installation de gettext. Par défaut, c'est /usr/local.

--with-gmp
: Inclut le support de GMP.

--with-hyperwave
: Inclut le support de Hyperwave.

--with-icap[=DIR]
: Inclut le support d'ICAP.

--with-iconv[=DIR]
: Inclut le support d'iconv.

--with-imap[=DIR]
: Inclut le support d'IMAP. DIR est le dossier d'installation de c-client. Par défaut, c'est /usr/local.

--with-kerberos[=DIR]
: IMAP: Inclut le support de Kerberos. DIR est le dossier d'installation de Kerberos.

--with-imap-ssl[=DIR]
: IMAP: Inclut le support de SSL. DIR est le dossier d'installation de OpenSSL.

--with-ircg-config
: Chemin jusqu'au script de configuration ircg-config.

--with-ircg
: Inclut le support de ircg.

--with-java[=DIR]
: Inclut le support de Java. DIR est le dossier d'installation du JDK. Cette extension ne peut être compilée que sous forme de librairie partagée.

--with-ldap[=DIR]
: Inclut le support de LDAP. DIR est le dossier d'installation de LDAP.

--enable-mailparse
: Active le support de mailparse.

--enable-mbstring
: Active le support des chaînes de caractères multi-octets.

--enable-mbstr-enc-trans
: Active la traduction en jeu de caractères japonais.

--with-mcal[=DIR]
: Inclut le support de MCAL.

--with-mcrypt[=DIR]
: Inclut le support de mcrypt. DIR est le dossier d'installation de c-client.

--with-mhash[=DIR]

Inclut le support de mhash. DIR est le dossier d'installation de mhash.

--with-mnogosearch[=DIR]
Inclut le support de mnoGoSearch. DIR est le dossier d'installation de mnoGoSearch. Par défaut, c'est /usr/local/mnogosearch.

--with-muscat[=DIR]
Inclut le support de muscat.

--with-ncurses
Inclut le support des ncurses.

--enable-pcntl
Active le support expérimental de pcntl. CGI uniquement!

--without-pcre-regex
Exclut le support des expressions régulières Perl. Utilisez --with-pcre-regex=DIR pour spécifier le dossier où sont le fichier d'entête, et le fichier de librairie, si vous n'utilisez pas la librairie de la distribution PHP.

--with-pfpro[=DIR]
Inclut le support de Verisign Payflow Pro.

--disable-posix
Inactive les fonctions POSIX.

--with-pspell[=DIR]
Inclut le support de PSPELL.

--with-qtdom
Inclut le support de QtDOM (requiert Qt >= 2.2.0).

--with-libedit[=DIR]
Inclut le support de libedit readline.

--with-readline[=DIR]
Inclut le support de readline. DIR est le dossier d'installation de readline.

--with-recode[=DIR]
Inclut le support de recode. DIR est le dossier d'installation de recode.

--with-satellite[=DIR]
Active le support expérimental de CORBA via Satellite. DIR est le dossier d'installation d'ORBit.

--with-mm[=DIR]
Inclut le support de mm pour le stockage de sessions.

--enable-trans-sid
Active la propagation transparente des sessions.

--disable-session
Inactive le support des sessions.

--enable-shmop
Active le support de shmop (mémoire partagée).

--with-snmp[=DIR]
Inclut le support de SNMP. DIR est le dossier d'installation de SNMP. Par défaut, il recherche dans un certains nombre de chemins d'installation classiques. Utilisez DIR=shared pour compiler SNMP comme librairie partagée, ou bien DIR=shared,chemin pour compiler SNMP comme librairie partagée tout en en précisant le chemin.

--enable-ucd-snmp-hack
Active le hack pour UCD SNMP.

--enable-sockets
Active le support des sockets.

--with-regex=TYPE
Type de librairie d'expression régulières. les valeurs possibles sont : system, apache et php.

--with-system-regex
Utilise la librairie d'expression régulières du système (obsolète).

--enable-sysvsem
: Active le support des sémpaphores System V.

--enable-sysvshm
: Active le support de la mémoire partagée System V shared.

--with-vpopmail[=DIR]
: Inclut le support de vpopmail.

--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 paralleles).

--with-pic
: Essaie d'utiliser uniquement des objest PIC/non-PIC. Par défaut, c'est both (les deux).

--with-yaz[=DIR]
: Inclut le support de YAZ (ANSI/NISO Z39.50). DIR est le dossier d'installation de YAZ.

--enable-memory-limit
: Compile PHP avec le support de la limitation de mémoire.

--disable-url-fopen-wrapper
: Inactive 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.

--disable-bcmath
: Compile PHP sans la librairie bcmaths. Valable uniquement pour PHP 3!

--with-imsp[=DIR]
: Inclut le support de IMSp. DIR est le dossier d'IMSP's qui contient les fichiers d'entêtes et le fichier libimsp.a. Valable uniquement pour PHP 3!

--with-ftp
: Inclut le support de FTP. 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!

--disable-overload
: Inactive le support de surchargement d'objet par les utilisateurs. Valable uniquement pour PHP 3!

--enable-yp
: Inclut le support d'YP. Valable uniquement pour PHP 3!

--with-zip
: Inclut le support de ZIP (requiert zziplib >= 0.10.6). 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!

3.3.8.1.4 PHP options

--enable-maintainer-mode
Active des règles de compilation (make) et des dépendances par 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 activ. Par défaut, c'est DIR vaut /usr/local/php/bin.

--enable-magic-quotes
Active les guillemets magiques.

--disable-short-tags
Inactive les balises courtes `<?` .

3.3.8.1.5 Server options

--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 transferts pour mod_charset (Rus 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-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-phttpd=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).

3.3.8.1.6 XML options

--with-dom[=DIR]
Inclut le support DOM (requiert libxml >= 2.4.2). DIR est le dossier d'installation de libxml. Par défaut, c'est /usr.

--disable-xml
Inactive le support XML basé sur la librairie expat, fournie dans la distribution de PHP.

--with-expat-dir=DIR
XML: dossier d'installation de la librairie externe expat.

--with-xmlrpc[=DIR]
Inclut le support XMLRPC-EPI.

--enable-wddx
Inclut le support WDDX.

3.3.9 Installation sous Windows

Cette section s'applique aux systèmes Windows 95/98/Me et Windows NT/2000/XP. (La configuration n'est actuellement pas confirmée sous XP). Ne vous attendez pas à ce que PHP fonctionne sur les plateformes 16 bits, comme Windows 3.1. Parfois, on décrira les plateformes supportées sous le nom de Win32.

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'installeur Windows de PHP disponible depuis les pages de téléchargement , 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.

Installez votre serveur HTTP favori sur votre système et assurez-vous qu'il fonctionne.

Exécutez l'installeur et suivez les instructions fournies par le wizard. 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.

Le wizard 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'arboresence web est affichée, et vous pouvez sélectionner les dossiers qui utiliseront PHP.

Une fois l'installation terminée, l'installeur 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 installeur automatique vous permet de réaliser l'installation en un tour de main, mais n'est pas destinée à l'utilisation sur des serveurs de production.

3.3.9.2 Manual Installation Steps

Instructions Générales d'installation

Ce guide vous permet d'installer et de configurer manuellement PHP sur vos stations Windows 9x/Me/NT/2000. La première version de ce guide a été compilée par Bob Silva . La version originale est disponible (en anglais) à http://www.umesd.k12.or.us/php/win32install.html .

Ce guide fournit une aide d'installation pour :

- Personal Web Server (Version la plus récente recommandée)
- Internet Information Server 3 ou 4
- Apache 1.3.x
- Omni HTTPd 2.0b1 et plus récent
- Oreilly Website Pro
- Xitami
- Netscape Enterprise Server, iPlanet

PHP 4 pour Windows est décliné en deux versions : un exécutable CGI (`php.exe`), et plusieurs modules SAPI (par exemple php4isapi.dll). Cette dernière forme est nouvelle pour PHP 4 et founit des performances améliorées ainsi que des fonctionnalités supplémentaires.

Attention
Les modules SAPI ont été significativement améliorée en version 4.1. Toutes fois, vous pouvez rencontrer des erreurs de serveur ou des erreurs de modules tels qu'ASP, dans les anciens systèmes.

Si vous choisissez l'un des modules SAPI et utilisez Windows 95, pensez à télécharger la mise à jour DCOM à Microsoft DCOM pages . Pour le module ISAPI, comme un serveur web compatible est nécessaire (testé avec IIS 4.0, PWS 4.0 et IIS 5.0). IIS 3.0 **n'est pas** supporté; vous devez télécharger et installer le Windows NT 4.0 Option Pack avec IIS 4.0 si vous voulez le support natif de PHP.

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

- Extrayez la distribution dans le dossier de votre choix. `"C:\php\"` est une bonne idée.

- Vous devez vous assurer que les DLL que PHP utilise sont disponibles. La liste précise des DLL dépend du serveur web que vous utiliser, et de la méthode que vous utiliser pour faire exécuter vos script (CGI ou module). `php4ts.dll` est toujours nécessaire. Si vous utiliser PHP en module, (e.g. ISAPI or Apache), vous aurez besoin de la DLL ad hoc, dans le dossier `sapi` . Si vous utilisez des extensions PHP, vous aurez besoin des DLL associées. Pour être certains que les DLL sont accessibles, vous pouvez les copier dans le dossier système (i.e. `winnt/system32` ou `windows/system`) ou vous pouvez vous assurer qu'elles sont dans le même dossier que l'exécutable PHP ou les autres DLL de votre serveur web (i.e. php.exe, ou php4apache.dll).

  L'exécutable binaire PHP, les modules SAPI, et certaines extensions utilisent des DLL externes. Assurez vous que ces DLL sont dans votre distribution, et dans un dossier qui est cité dans le PATH Windows. Le mieux à faire est de copier les fichiers ci−dessous dans votre dossier système, qui est généralement :

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

  Les fichiers à copier sont :

  - php4ts.dll, s'il existe, écrasez le
  - Les fichiers dlls de votre distribution. Si vous les avez déjà installé, ne les remplacez pas, sauf si quelquechose ne fonctionne pas correctement (avant de les écraser, il est recommandé de les sauver de toutes manières, au cas où quelque chose irait de travers).

  Téléchargez la dernière version de Microsoft Data Access Components (MDAC) pour votre plateforme, spécialement si vous utilisez Microsoft Windows 9x/NT4. MDAC est disponible à http://www.microsoft.com/data/ .
- Copiez le fichier `php.ini−dist` dans votre dossier %WINDOWS% sous Windows 95/98, ou vers votre dossier %SYSTEMROOT% sous Windows NT ou Windows 2000 et renommez le en `php.ini` . Votre dossier %WINDOWS% 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 pourez le faire, même si PHP focntionne parfaitement avec les configurations par défaut.
- Editez votre fichier `php.ini` :
  - Vous devez changer votre option 'extension_dir' pour qu'il pointe sur votre dossier d'installation PHP, ou vers l'endroit où vous avez installé vos 'php_*.dll'. ex: c:\php\extensions
  - Si vous utilisez Omni Httpd, sautez l'étape suivante. Modifiez 'doc_root' pour qu'il pointe sur votre racine de serveur web. ex: `c:\apache\htdocs` ou `c:\webroot` .
  - Choisissez les modules que vous voulez charger lorsque PHP démarre. Vous pouvez décommenter les lignes 'extension=php_*.dll' pour charger ces modules. Certains modules requièrent que des librairies supplémentaires soient installées sur votre système. La FAQ PHP a plus d'informations sur ces librairies. Vous pouvez aussi charger dynamiquement ces librairies avec `dl("php_*.dll");` . Voyez la section sur les extensions Windows .
  - Sous PWS et IIS, vous pouvez configurer le fichier `browscap.ini` pour qu'il pointe sur `c:\windows\system\inetsrv\browscap.ini` sous Windows 9x/Me, `c:\winnt\system32\inetsrv\browscap.ini` sous NT/2000 et `c:\windows\system32\inetsrv\browscap.ini` sous XP.

- Notez que le dossier `mibs` fourni avec la distribution Windows, contient des fichiers de support du protocole SNMP. Ce dpssier doit être déplacé dans le dossier `DRIVE:\usr\mibs` (`DRIVE` est le périphérique où PHP est installé.)
- Si vous n'utilisez pas NTFS sous Windows NT, 2000 et XP, assurez vous que l'utilisateur qui fait tourner le serveur web a bien les permission pour accéder à votre `php.ini` .

- Avec PWS, donnez les droits d'exécution au webmestre :
  - Démarrez le PWS Web Manager
  - Editez les Properties du dossier "Home"
  - Sélectionnez la coche "execute"

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 Préparation

Avant de commencer, il faut télécharger un maximum de fichiers!

- Pour commencer, téléchargez le Cygwin depuis le miroir cygwin le plus proche. Cela vous donnera les utilitaires GNU les plus populaires, utilisés durant le processus de compilation.
- Téléchargez le reste des utilitaires de compilation dont vous aurez besoin depuis le site PHP à http://www.php.net/extra/win32build.zip .
- Téléchargez le code source du DNS utilisé par PHP à http://www.php.net/extra/bindlib_w32.zip . Il remplacera le fichier `resolv.lib` inclut dans `win32build.zip` .
- Si vous n'avez pas d'utilitaire de dézippage, vous devez en télécharger un. Une version libre est disponible à InfoZip .
- Si vous prévoyez de compiler PHP comme module statique pour Apache, vous aurez aussi besoin des sources Apache .

Finalement, vous aurez besoin des sources PHP 4 elles−mêmes!! Les dernières versions sont accessibles sur le serveur CVS anonyme . Si vous téléchargez une version intermédiaire ou la source , vous devez non seulement extraire les fichiers, mais aussi convertir les nouvelles lignes en leur équivalent windows (crlf) dans les fichiers `*.dsp` et `*.dsw` avant que Microsoft Visual C++ ne soit capable de les comprendre.

Note
Placez les dossiers `Zend` et `TSRM` dans le dossier `php4` pour que les projets puissent les trouver durant la compilation.

3.3.9.3.2 Compiler tout ensemble

- Suivez les instructions pour installer l'utilitaire d'unzip de votre choix.
- Exécutez `setup.exe` et suivez les instructions d'installation. Si vous décidez d'installer dans un autre dossier que `c:\cygnus` , indiquez le au processus de compilation en modifiant la variable d'environnement Cygwin. Sous Windows 9x/Me, modifier une variable d'environnement se fait en ajoutant une ligne dans le fichier `autoexec.bat` . Sous Windows NT, allez dans le menu "Démarrer => Paramètres => Panneau de contrôle => Système " ("My Computer => Control Panel => System ") et sélectionnez l'onglet "environnement" ("environment").

Attention
Créez un dossier temporaire pour Cygwin, sinon de nombreuses commandes (comme bison) échoueront. Sous Windows 9x/Me, `mkdir C:\TMP` . Sous Windows NT, `mkdir %SystemDrive%\tmp` .

- Créez un dossier et dézippez `win32build.zip` dedans.
- Lancez Microsoft Visual C++, et allez dans le menu "select Tools => Options". Dans le dialogue, sélectionnez l'onglet "directories". Assurez−vous que `cygwin\bin` , `win32build\include` , et `win32build\lib` sont bien dans les menus déroulants "Executables", "Include", et "Library". (Pour ajouter une entrée, sélectionnez une ligne blanche, et tapez). Une entrée typique ressemble à ceci :
    - `c:\cygnus\bin`
    - `c:\php−win32build\include`
    - `c:\php−win32build\lib`

  Pressez "OK", et sortez de Visual C++.
- Créez un autre dossier et dézippez `bindlib_w32.zip` dedans. Décidez si vous avez besoin des symboles de débuggage (bindlib − Win32 Debug) ou non (bindlib − Win32 Release). Compilez la configuration adéquate :
    - Pour les utilisateurs de GUI, 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.

3.3.9.3.3 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 une librairie `php.exe` utilisable, soit dans votre dossier

Debug_TS soit sans le dossier Release_TS .

Répétez les instructions ci−dessus avec php4isapi.dsp (qui est dans sapi\isapi) pour compiler le code nécessaire pour intégrer PHP avec Microsoft IIS.

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

3.3.9.4 Installation of Windows extensions

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 4.0.6 le support des extensions BCMath, Calendar, COM, FTP, MySQL, ODBC, PCRE, Session, WDDX et XML est activé **par défaut** . Vous n'avez pas à charger d'extensions particulière pour utiliser ces fonctions. Lisez le fichier README.txt inclus dans la distribution, ou bien install.txt pour connaitre la liste des modules automatiquement activés.

Note
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
php_bz2.dll	Fonctions de compression bzip2	Aucune
php_calendar.dll	Fonctions de conversions calendaires Calendar	Disponible depuis PHP 4.0.3
php_cpdf.dll	Fonctions ClibPDF	Aucune
php_crack.dll	Fonctions de CrackLib	Aucune
php3_crypt.dll	Fonctions de Crypt	Inconnu
php_ctype.dll	Fonctions ctype	Aucune

php_curl.dll	CURL, Client URL library functions	Requiert : libeay32.dll, ssleay32.dll (fournies)
php_cybercash.dll	Fonctions de paiement Cybercash	Aucune
php_db.dll	Fonctions DBM	Obsolètes. Utilisez DBA à la place (php_dba.dll)
php_dba.dll	DBA: DataBase (dbm−style) Abstraction layer functions	Aucune
php_dbase.dll	Fonctions dBase	Aucune
php3_dbm.dll	Fonctions Berkeley DB2	Inconnu
php_domxml.dll	Fonctions de DOM XML	Requiert : libxml2.dll (fournie)
php_dotnet.dll	Fonctions .NET	Aucune
php_exif.dll	Fonctions de lecture des entêtes JPEG Read EXIF	Aucune
php_fbsql.dll	Fonctions de FrontBase	Aucune
php_fdf.dll	Fonctions FDF: Forms Data Format.	Requiert : fdftk.dll (fournies)
php_filepro.dll	Fonctions filePro	Accès en lecture seule
php_ftp.dll	Fonctions FTP	Disponible depuis PHP 4.0.3
php_gd.dll	Fonctions de traitement des images GD	Aucune
php_gettext.dll	Fonctions Gettext	Requiert : gnu_gettext.dll (fournies)
php_hyperwave.dll	Fonctions HyperWave	Aucune
php_iconv.dll	Fonctions de conversions de jeux de caractères ICONV	Requiert : iconv−1.3.dll (fournies)
php_ifx.dll	Fonctions Informix	Requiert : Informix libraries
php_iisfunc.dll	Fonctions de gestion de IIS	Aucune
php_imap.dll	Fonctions IMAP, POP3 et NNTP	PHP 3: php3_imap4r1.dll
php_ingres.dll	Fonctions Ingres II	Requiert : Ingres II libraries
php_interbase.dll	Fonctions InterBase	Requiert : gds32.dll (fournies)
php_java.dll	Extension Java	Requiert : jvm.dll (fournies)
php_ldap.dll	Fonctions LDAP	Requiert : libsasl.dll (fournies)
php_mhash.dll	Fonctions Mhash	Aucune
php_ming.dll	Fonctions de générations de fichiers Flash avec Ming	Aucune
php_msql.dll	Fonctions de mSQL	Requiert : msql.dll (fournies)
php3_msql1.dll	mSQL 1 client	Inconnu
php3_msql2.dll	mSQL 2 client	Inconnu
php_mssql.dll	Fonctions MSSQL	Requiert : ntwdblib.dll (fournies)
php3_mysql.dll	Fonctions MySQL	Built−in in PHP 4
php3_nsmail.dll	Fonctions de email Netscape	Inconnu
php3_oci73.dll	Fonctions Oracle	Inconnu
php_oci8.dll	Fonctions Oracle 8	Requiert : librairies clientes Oracle 8
php_openssl.dll	Fonctions OpenSSL	Requiert : libeay32.dll (fournies)
php_oracle.dll	Fonctions de Oracle 7	Requiert : librairies clientes Oracle 7
php_pdf.dll	Fonctions de PDF	Aucune
php_pgsql.dll	Fonctions de PostgreSQL	Aucune
php_printer.dll	Fonctions d'impressions	Aucune
php_xslt.dll	Fonctions XSLT	Requiert : sablot.dll (fournies)

php_snmp.dll	Fonctions SNMP	NT only!
php_sybase_ct.dll	Fonctions Sybase	Requiert : librairies clientes Sybase
php_yaz.dll	Fonctions YAZ	Aucune
php_zlib.dll	Fonctions de compression ZLib	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 d'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)

```
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 étape, vous pouvez aussi copier le binaire
  httpd et remplacer votre exécutable actuel. Assurez-vous tout
  de même que le serveur est bien éteint.
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 éditer votre fichier php.ini pour modifier
  certaines options PHP. Si vous préférez placer ce fichier ailleurs,
  utilisez --with-config-file-path=/path lors de l'étape 8.
17. Editez votre fichier httpd.conf ou srm.conf file et ajoutez :
     Pour PHP 3:   AddType application/x-httpd-php3 .php3
     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'éteindre 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 à apachectl:
/path/to/httpsdctl stop
/path/to/httpsdctl start
4. En utilisant mod_ssl, ou un autre seveur 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 a `locate` ou `whereis` ou `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` de 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 lien statiquement avec `httpd` . Le support PostgreSQL est alors inclut 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'éxécuter plus rapidement PHP. Pour plus d'informations, voyez la page web sur le support des DSO .

Note

Le fichier de configuation d'Apache actuel est fourni avec une section qui ressemble à ceci :

```
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 --with-apxs=/path/to/apxs. Vous ne devez PAS utiliser la version d'apxs qui 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 Instructions Générales d'installation , 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 .phtml`

- `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 .phtml`

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
You may find after using the windows installer for Apache that you need to define the `AddModule` directive for `mod_php4.c` in the configuration file (`httpd.conf`). This is done by adding `AddModule mod_php4.c` to the `AddModule` list, near the beginning of the configuration file. This is especially important if the `ClearModuleList` directive is defined. Failure to do this may mean PHP will not be registered as an Apache module.

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"); ?>` . Substitutez `original_php_script.php` par le nom du fichier que vous souhaitez afficher.

Note
Sous Win-Apache tous les antislash 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 prodoction, 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

Tant que PHP 4.3.0 n'est pas publié, nous recommandons l'utilisation les versions suivantes , avec la dernière version d'Apache 2.0 :

- Les dernières sources stables en développement, téléchargeable sur http://snaps.php.net/
- une pré version de PHP 4.3.0 téléchargaeble sur http://qa.php.net/
- la version CVS de PHP, disponible sur le serveur CVS , en connexion anonyme.

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

Note
Le supoprt des SAPI d'Apache 2.0 a commencé avec PHP 4.2.0. PHP 4.2.3 est connu pour fonctionner avec 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 mentionnées de PHP fonctionnent avec Apache 1.3.x.

3.3.11.2 Installer PHP et Apache 2 sur Linux

Téléchargez la plus récente version de Apache 2.0 et une version appropriée de PHP , tel que décrit ci−dessus. Ce guide rapide couvre les manipulations de base, nécessaire à l'installation de Apache 2.0 et PHP . Pour plus d'informations, lisez la Documentation Apache . Les numéros de version sont omis ici, poru 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, configuré avec le support des modules dynamiques, et le préfork 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, utiliser /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 placez ce fichier à un autre endroit, utilisez l'option
--with-config-file-path=/path en étape 10.

14. Edites 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 sur la driot 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 sur, il y a bien d'autres options de configuration disponibles pour les deux logiciels, Apache et PHP . Pour plus de détails, utilisez la commande `./configure --help` dans le dossier de souces approprié. Si vous souhaitez compiler uen version multi-threadé de Apache 2.0 vous devrez remplacer le module standard MPM `prefork` avec `worker` ou `perchild` . Pour ce faire, ajoutez à la ligne de configuration de l'étape 6, l'option `--with-mpm=worker` ou `--with-mpm=perchild` . Soyez conscient des conséquences, et comprenez bien ce que vous faites. Pour plus de détails sur ce sujet, lisez la documentation Apache sur le module MPM .

Note
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 a ce moment.

Téléchargez la plus récente version de Apache 2.0 et une version de PHP appropriée, tel que mentionné au début de ce chapiter. Suivez les instructions d'installation manuelle puis revenez ici pour réaliser l'intégration de PHP et Apache.

Il ya 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.

Configurer PHP 4.3.0 avec Apache 2.0 en CGI

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

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 avez à 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 comme une CGI. 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 de lancer des scripts PHP sous leur UID respectives. Lisez attentivement le chapitre consacré à la sécurité si vous souhaitez utiliser cette solution.

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, bien plus facilement que si vous attendez.

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 ne risquez pas 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'éxécution de vos scripts. `make bench` ignore le fichier de configuration file .

Note
`make bench` n'est disponible qu'en PHP 3.

3.3.12.3 Utiliser les variables prédéfinies

Certaines variables d'environnements fournies par les serveurs web ne sont pas disponibles dans les spécifications actuelles CGI/1.1. 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 supposerons
    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 extenstions 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 registry 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 registry.

- 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'icone. Vous pouvez ignorer ces étapes si vous préférez que vos scripts PHP s'ouvrent dans 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 clée dans le panneau de droite, et double cliquez dans "default value", puis entrez `phpfile` .
- Répétez ces instructions pour toutes les extensions que vous avez associées 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 doite, 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 la nouvelle registry.

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 SAPI, 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 serveyr 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 apparaitra 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'assirurera que le script existe, et résoudra les problèmes d'autentification 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 retournés.

  Répétez les étapes 3 et 4 pour chaque extension que vous souhaitez associer aux scripts PHP. (`.php` et `.phtml` sont les plus répandues.)
- Configurer la sécurité nécessaire (dans "Internet Service Manager"), et si votre serveur NT utilise NTFS, ajoutez les droits adéquats pour I_USR_, 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 spécifique à l'installation Netscape et iPlanet, aussi bien pour Sun Solaris que Windows .

Vous pouvez trouvez 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` .

Exemple d'installation pour Netscape Enterprise sous Solaris

```
Instructions pour Sun Solaris 2.6 avec Netscape Enterprise Server 3.6
From: bhager@invacare.com
1. Installez les packages suivants depuis le serveur www.sunfreeware.com
ou un miroir ad hoc :
flex−2_5_4a−sol26−sparc−local
gcc−2_95_2−sol26−sparc−local
gzip−1.2.4−sol26−sparc−local
perl−5_005_03−sol26−sparc−local
bison−1_25−sol26−sparc−local
make−3_76_1−sol26−sparc−local
m4−1_4−sol26−sparc−local
autoconf−2.13
automake−1.4
mysql−3.23.24−beta (if you want mysql support)
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
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. cd ../php−x.x.x
6. Pour les étapes suivantes, assurez−vous que /opt/netscape/suitespot/
correspond
bien à votre installation du serveur netscape. Sinon, indiquez le chemin
correct :
/configure
−−with−mysql=/usr/local/mysql
−−with−nsapi=/opt/netscape/suitespot/
−−enable−track−vars −−enable−libgcc
7. make
8. 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/` .

Exemple de configuration pour Netscape Enterprise

```
Instructions de configuration for Netscape Enterprise Server
From: bhager@invacare.com
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 dépendre de
votre OS, pour Unix c'est quelque chose de proche de
/opt/netscape/suitespot/bin/libphp4.so.
Il est conseillé de placer les lignes suivantes après les
lignes de types mime :
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!"
<object name="default">
.
.
.
.#NOTE La ligne suivante doit être placée après toutes
.#les lignes 'ObjectType'
.# et avant les lignes 'AddLog'
Service fn="php4_execute" type="magnus-internal/x-httpd-php"
.
.
</Object>
<Object name="x-httpd-php">
ObjectType fn="force-type" type="magnus-internal/x-httpd-php"
Service fn=php4_execute
</Object>
Configuration d'authentification
L'authentification PHP ne peut pas être utilisée avec d'autre
authentification.
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>
Pour configurer l'authentification PHP pour un dossier, ajoutez
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 :

Exemple de configuration pour Netscape Enterprise 4.x

```
    Placez ces lignes après les types MIME, et tout le reste ressemble
    à l'exemple ci-dessus :
From: Graeme Hoose (GraemeHoose@BrightStation.com)

Init fn="load-modules" shlib="/path/to/server4/bin/libphp4.so" funcs="php4_init,php4_close,php4_exec
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 créee 5 lignes importantes dans le fichier obj.conf et permet au serveur de gérer les scripts CGI).
- Dans le serveur Netscape Enterprise Administration Server, créeez 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, créeez 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
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 avamt 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 ded détails sur la configiration de PHP en filtre NSAPI sont disponibles :
http://benoit.noss.free.fr/php/install-php4.html

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 on prouvé que ce n'est pas une solution stable.

- Step 1: Installez le serveur OmniHTTPd.
- Step 2: Faites un clic droit sur l'icone 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`

Répetez 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 béta 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 Installation Xitami

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

3.3.19.1 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 champs "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.

3.3.20 Autres serveurs web

PHP peut être compilé pour fonctionner avec de nombreux autres serveurs web. Reportez−vous à Options particulières aux 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.

3.3.21 Des problèmes?

Lisez la FAQ

Certains problèmes sont récurrents : les plus communs sont listés dans la FAQ PHP, disponible à http://www.php.net/FAQ.php .

3.3.21.1 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 mailing liste : `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 ...), et n'hésitez pas à envoyer suffisamment de code pour que nous puissions reproduire l'erreur.

3.3.21.2 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 le 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.

Extrait du php.ini

```
; tout texte sur une ligne, situé après un point−virgule  ";" est ignoré
[php] ; les marqueur de section (texte entre crochets) sont aussi ignorés
; Les valeurs 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"
```

Lorsque vous utilisez le module Apache, vous pouvez aussi changer les paramètres de configurations 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 n'y a que quelques directives dans le fichier de configuration d'Apache qui vous permettent de modifier la configuration de PHP.

php_value name value
: Cette directive affecte une valeur à la variable spécifiée.

php_flag name on|off
: Cette directive est utilisée pour activer ou désactiver une option.

php_admin_value name value
: Cette directive affecte une valeur à la variable spécifiée. La directive "Admin" ne peut être utilisée que dans le fichier de configuration d'Apache, et non dans un fichier `.htaccess` .

php_admin_flag name on|off
: Cette directive est utilisée pour activer ou désactiver l'option précédente.

Exemple de configuration Apache

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

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 la fonction `get_cfg_var` .

3.4.1.1 Directives de configuration générale

allow_url_fopen boolean
: Cette option autorise les accès au réseau des fonctions `fopen` . Par défaut, l'accès est autorisé aux procédures d' accès distants , avec les protocoles `FTP` , `NNTP` , et certaines extensions telles ques zlib.

  Note

  Cette option a été introduite immédiatement après la version 4.0.3. Pour les versions jusqu'à 4.0.3 incluse, vous pouvez désactiver cette fonctionnalité au moment de la compilation en utilisant la configuration `--disable-url-fopen-wrapper` .

asp_tags booléen
: Active l'utilisation des balises de type ASP <% %>, en plus des traditionnelles balises <?php ?> . Cela inclut l'utilisation du raccourcid <%= $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.

auto_append_file chaîne de caractères
: 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-appending.

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

`auto_prepend_file` *`chaîne de caractères`*
: 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.

`cgi_ext` *`chaîne de caractères`*
: (NDT : aucune documentation n'est fournie).

`display_errors` *`booléen`*
: Cette directive détermine si les erreurs doivent être affichées à l'écran au format HTML ou non.

`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.

`engine` *`boolean`*
: Cette directive ne sert vraiment que si PHP est un module Apache. Elle sert aux sites qui veulent activer ou désactiver l'analyse des fichiers par PHP, dossier par dossier. En mettant `php3_engine off` au bon endroit, dans le fichier `httpd.conf` , PHP peut être activé ou désactivé.

`error_log` *`string`*
: Nom du fichier où les erreurs seront enregistrées. Si la valeur spéciale `syslog` est utilisée, les erreurs sont envoyées au système standard d'historique. Sous UNIX, c'est `syslog(3)` et sous Windows NT, c'est l'historique d'événements. L'historique système n'est pas supporté sous Windows 95.

`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 :

valeur du bit	niveau choisi
1	erreurs normales
2	alertes normales
4	erreurs d'analyseur (parser errors)
8	alertes non critiques

  Par défaut, la valeur est de 7 (erreurs normales, alertes normales et erreurs d'analyseur sont affichées).

`open_basedir` *`string`*
: Limite l'espace où PHP peut ouvrir des fichiers.

  Lorsqu'un script essaie d'ouvrir un fichier avec les fonctions fopen ou gzopen (par exemple), la localisation du fichier est vérifiée. Si ce fichier est hors du dossier cité dans cette directive, PHP refusera de l'ouvrir. Tous les liens symboliques sont résolus, et subissent aussi la restriction.

  La valeur spéciale `.` indique que le dossier courant du script est utilisé comme open_basedir.

  Sous Windows, séparez les noms de dossiers par un point virgule (;). Sur les autres systèmes, séparez les noms de dossiers par des deux points (:). Lorsque PHP est un module Apache, la valeur de la directive `open_basedir` des dossiers parents est automatiquement héritée par les fils.

Note
Le support pour les dossiers multiples a été ajouté dans 3.0.7.

La valeur par défaut est : libre accès à tous les fichiers.

gpc_order chaîne de caractères
: Etablit l'ordre de préséance des méthodes GET/POST/COOKIE. Par défaut, cette directive est établie à "GPC". En affectant "GP" à cette directive, PHP ignorera les cookies, et écrasera toute méthode GET utilisée par une méthode POST avec des variables du même nom.

ignore_user_abort chaîne de caractères
: Désactivée par défaut. Si cette directive est activée, alors tous les scripts lancés iront jusqu'à leur terme, même si le client se déconnecte en plein milieu. Voir aussi la fonction `ignore_user_abort` .

include_path string
: Spécifie une liste de dossiers, où les fonctions `require` , `include` et fopen_with_path() (NDtraducteur : cette fonction semble avoir disparu) 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=.:/home/httpd/php-lib
```

Windows include_path

```
include_path=".;c:\www\phplib"
```

La valeur par défaut pour cette directive est `.` , c'est-à-dire le dossier courant.

isapi_ext string
: (Aucune documentation n'est fournie)

log_errors boolean
: Indique où les messages d'erreur générés doivent être écrits. Cette fonction est spécifique aux serveurs.

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.

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.

magic_quotes_sybase boolean
: Si `magic_quotes_sybase` est activé, les guillemets simples seront échappés avec un autre guillemet simple, plutôt qu'un antislash, si `magic_quotes_gpc` ou `magic_quotes_runtime` est activé.

max_execution_time integer
: Fixe le temps maximal d'éxécution d'un script, en secondes. Cela permet d'éviter que des scripts en boucles infinies saturent le serveur.

memory_limit entier
: Grâce à cette option, vous pouvez donner la quantité maximale de mémoire qu'un

script peut allouer. Ceci permet de se prevenir contre des scripts mal codés pouvant consommer toute la mémoire disponible d'un serveur

nsapi_ext chaîne de caractères
: Aucune documentation n'est fournie.

register_globals boolean
: Cette option active l'enregistrement des variables EGPCS (Environnement, GET, POST, Cookie, Serveur), en tant que variables globales. Vous pouvez désactiver cette fonction si vous ne voulez pas truffer vos scripts avec des valeurs utilisateurs. Cette option est surtout utile lorsqu'elle est utilisée conjointement avec track_vars − dans ce cas, vous pouvez accéder à toutes les variables EGPCS grâce aux tableaux `$HTTP_ENV_VARS` , `$HTTP_GET_VARS` , `$HTTP_POST_VARS` , `$HTTP_COOKIE_VARS` , et `$HTTP_SERVER_VARS` .

short_open_tag booléen
: Active ou désactive l'utilisation des balises courtes, (<? ?>). Si vous voulez utiliser PHP et XML en même temps, vous devez désactiver cette option. Si cette option est désactivée, vous devez utiliser la forme longue des tags, (`<?php ?>`).

sql.safe_mode booléen
: Aucune documentation n'est fournie.

track_errors booléen
: Si cette option est activée, le dernier message d'erreur sera placé dans la variable globale `$php_errormsg` .

track_vars booléen
: Si cette option est activée, lors de l'appel des méthodes GET, POST et l'utilisation des cookies, les variables sont disponibles dans des tableaux associatifs globaux appelés respectivement `$HTTP_GET_VARS` , `$HTTP_POST_VARS` et `$HTTP_COOKIE_VARS` .

upload_tmp_dir chaîne de caractères
: Indique le répertoire utilisé lors du chargement d'un fichier sur un serveur. Ce répertoire doit être accessible en lecture pour l'utilisateur qui lance le script PHP.

upload_max_filesize entier
: La taille maximale d'un fichier téléchargé sur le serveur. La valeur doit être en octets. Par défaut, elle est de 2 méga−octets. in bytes.

user_dir chaîne de caractères
: Répertoire où sont stockés les fichiers PHP dans le répertoire d'un utilisateur. Par exemple, `public_html` .

warn_plus_overloading booléen
: Si cette option est activée, PHP émet une alerte lorsque l'opérateur plus (+) est utilisé sur une chaîne de caractères. Cela permet de réperer plus facilement les scripts qui doivent être réécrits en utilisant l'opérateur de concaténation (`.`) plutôt que l'opérateur plus.

3.4.1.2 Directives de configuration concernant le mail

SMTP chaîne de caractères
: Sous Windows, adresse IP ou nom que PHP doit utiliser pour envoyer du mail avec la fonction `mail` .

sendmail_from chaîne de caractères
: Sous Windows, valeur du champs "From:" qui doit être utilisée lors de l'envoi de mail.

sendmail_path chaîne de caractères
: 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` .

3.4.1.3 Directives de configuration du "Safe Mode"

`safe_mode` *`booléen`*
: Cette directive active ou désactive l'option safe mode , un mode de sécurisation de PHP lors du partage avec plusieurs utilisateurs.

`safe_mode_exec_dir` *`chaîne de caractères`*
: Si l'option "SAFE MODE" est activée, `system` et les autres fonctions exécutant des programmes systèmes refusent de se lancer si ces programmes ne sont pas placés dans ce répertoire.

3.4.1.4 Directives de configuration de débugage.

`debugger.host` *`chaîne de caractères`*
: Adresse IP ou nom de l'hôte utilisé pour le débugage.

`debugger.port` *`chaîne de caractères`*
: Numéro du port utilisé pour le débugage.

`debugger.enabled` *`booléen`*
: Activation ou désactivation du débugueur.

3.4.1.5 Directives de chargement des extensions

`enable_dl` *`booléen`*
: Cette directive n'est réellement utile que dans le cas d'une compilation comme module Apache. Vous pouvez activer le chargement dynamique des extensions avec la fonction `dl` , et cela de manière locale à chaque serveur virtuel ou à chaque répertoire.

  La principale raison qui pousse à désactiver le chargement dynamique est un problème de sécurité. Lorsque le chargement dynamique est activé, il est possible d'ignorer les directives safe mode ou "open_basedir".

  Par défaut, il est possible d'utiliser le chargement dynamique, sauf lorsque la directive safe mode est activée. En effet, il est alors impossible d'utiliser la fonction `dl` .

`extension_dir` *`chaîne de caractères`*
: Définit le répertoire dans lequel le PHP doit chercher les extensions lors du chargement dynamique.

`extension` *`chaîne de caractères`*
: Définit les extensions qui doivent être chargées lors du démarrage du PHP.

3.4.1.6 Directives de configuration MySQL

`mysql.allow_persistent` *`booléen`*
: Active ou désactive les connexions persistantes à la base de données MySQL.

`mysql.default_host` *`chaîne de caractères`*

Adresse par défaut du serveur, à utiliser lors de la connexion à un serveur MySQL, si aucun hôte n'est spécifié.

mysql.default_user *chaîne de caractères*
Utilisateur par défaut, à utiliser lors de la connexion à un serveur MySQL, si aucun utilisateur n'est spécifié.

mysql.default_password *chaîne de caractères*
Mot de passe par défaut, à utiliser lors de la connexion à un serveur MySQL, si aucun mot de passe n'est spécifié.

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 conecte au serveur local, si aucune autre socket n'est spécifiée.

mysql.max_persistent *entier*
Nombre maximum de connexions persistantes à une base de données MySQL, par processus.

mysql.max_links *entier*
Nombre de connexions maximum à une base de données MySQL, par processus, incluant les connexions persistantes

3.4.1.7 Directives de configuration mSQL

msql.allow_persistent *booléen*
Active ou désactive les connexions persistantes à la base de données mSQL.

msql.max_persistent *entier*
Nombre maximum de connexions persistantes à une base de données mSQL, par processus.

msql.max_links *entier*
Nombre maximum de connexions à une base de données mSQL, par processus, incluant les connexions persistantes.

3.4.1.8 Directives de configuration Postgres

pgsql.allow_persistent *booléen*
Active ou désactive les connexions persistantes à la base de données Postgres.

pgsql.max_persistent *entier*
Nombre maximum de connexions persistantes à une base de données Postgres, par processus.

pgsql.max_links *entier*
Nombre maximal de connexions à une base de données Postgres, par processus, incluant les connexions persistantes.

3.4.1.9 Directives de configuration SESAM

sesam_oml *string*
Nom de la librairieBS2000 PLAM contenant les pilotes de connexion SESAM.

Obligatoire pour les fonctions SESAM. La librairie BS2000 PLAM doit être configurée avec ACCESS=READ,SHARE=YES , car elle doit être accessible à l'utilisateur Apache.

sesam_configfile string

Nom du fichier de configuration de l'application SESAM. Obligatoire pour les fonctions SESAM. Le fichier BS2000 doit être accessible être accessible à l'utilisateur Apache.

Le fichier de configuration de l'application va contenir la configuration sur le schéma suivant (voir le manuel SESAM) :

```
CNF=B
NAM=K
NOTYPE
```

sesam_messagecatalog string

Nom du catalogue de messages SESAM. Dans la plupart des cas, cette directive n'est pas nécessaire. Seulement, si le fichier de messages SESAM n'est pas installé dans la table de messages BS2000, il faut indiquer sa localisation avec cette directive.

Le catalogue de messages doit être paramétré avec ACCESS=READ,SHARE=YES , car il doit être accessible à l'utilisateur Apache.

3.4.1.10 Directives de configuration Sybase

sybase.allow_persistent booléen

Active ou désactive les connexions persistantes à la base de données Sybase.

sybase.max_persistent entier

Nombre maximum de connexions persistantes à une base de données Sybase par processus.

sybase.max_links entier

Nombre maximum de connexions à une base de données Sybase, par processus, incluant les connexions persistantes.

3.4.1.11 Directives de configuration Sybase−CT

sybct.allow_persistent booléen

Active ou désactive les connexions persistantes à la base de données Sybase−CT. Par défaut, cette option est activée.

sybct.max_persistent entier

Nombre maximum de connexions persistantes à une base de données Sybase−CT par processus. Par défaut, cette option est à −1, ce qui signifie que le nombre de connexion est illimité.

sybct.max_links entier

Nombre maximum de connexions à une base de données Sybase−CT, par processus, incluant les connexions persistantes. Par défaut, cette option est à −1, ce qui signifie que le nombre de connexions est illimité.

sybct.min_server_severity entier

Les messages en provenance du serveur d'un niveau d'erreur égal à sybct.min_server_severity seront considérés comme des alertes (warnings). Cette valeur peut être modifiée à l'intérieur du script en appelant la fonction sybase_min_server_severity (NDtraducteur : cette fonction semble ne pas exister). Par défaut, cette valeur vaut 10.

sybct.min_client_severity *entier*
: Les messages en provenance de la librairie client avec un niveau d'erreur égal ou supérieur à sybct.min_client_severity seront considérés comme des alertes. Cette valeur peut être modifiée à l'intérieur du script en appelant la fonction sybase_min_client_severity (NDtraducteur : cette fonction semble ne pas exister). Par défaut, cette valeur vaut 10, ce qui annule tout rapport d'erreur.

sybct.login_timeout *entier*
: Délai de validité d'une tentative de connexion. Il est à noter que si max_execution_time est dépassé avant que la connexion n'expire, le script sera terminé avant le message d'erreur. Par défaut, cette valeur vaut 1 minute.

sybct.timeout *entier*
: Temps maximum en secondes avant qu'une tentative de requête "select_db" ou "query" non aboutie renvoie une erreur. Il est à noter que si max_execution_time est dépassé avant que la requête n'expire, votre script sera terminé avant le message d'erreur. Par défaut, il n'y a pas de limite.

sybct.hostname *chaîne de caractères*
: Nom de l'hôte à partir duquel vous vous connectez, afin d'être affiché par la fonction sp_who (NDtraducteur : cette fonction semble ne pas exister). Par défaut, cette valeur égale à 0.

3.4.1.12 Directives de configuration Informix

ifx.allow_persistent *booléen*
: Active les connexions persistantes à une base de données Informix.

ifx.max_persistent *entier*
: Nombre maximum de connexions persistantes à une base de données Informix, par processus.

ifx.max_links *entier*
: Nombre maximum de connexions à une base de données Informix par processus, en incluant les connexions persistantes.

ifx.default_host *chaîne de caractères*
: Hôte par défaut où se connecter si aucun hôte n'est spécifié par les fonctions `ifx_connect` ou `ifx_pconnect` .

ifx.default_user *chaîne de caractères*
: Utilisateur par défaut si aucun utilisateur n'est spécifié par les fonctions `ifx_connect` ou `ifx_pconnect` .

ifx.default_password *chaîne de caractères*
: Mot de passe par défaut si aucun mot de passe n'est spécifié par les fonctions `ifx_connect` ou `ifx_pconnect` .

ifx.blobinfile *booléen*
: 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 *booléen*
: 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 *booléen*
: 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 *booléen*
Lorsque cette option est activée, les espaces en fin de chaîne de caractères seront conservés lors d'une commande FETCH.

ifx.nullformat *booléen*
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 .

3.4.1.13 Directives de configuration pour les calculs mathématiques.

bcmath.scale *entier*
Nombre de chiffres après la virgule pour toutes les fonctions de précision mathématique arbitraire.

3.4.1.14 Directives de configuration du navigateur.

browscap *chaîne de caractères*
Nom du fichier de descriptif des clients HTML. Voir aussi get_browser .

3.4.1.15 Directives de configuration du pilote ODBC unifié

uodbc.default_db *chaîne de caractères*
Source de données ODBC à utiliser par défaut avec les fonctions odbc_connect ou odbc_pconnect .

uodbc.default_user *chaîne de caractères*
Nom d'utilisateur défaut avec les fonctions odbc_connect ou odbc_pconnect .

uodbc.default_pw *chaîne de caractères*
Mot de passe par défaut dans les fonctions odbc_connect ou odbc_pconnect .

uodbc.allow_persistent *booléen*
Cette option active ou désactive les connexions persistantes à la base de données, via le canal ODBC.

uodbc.max_persistent *entier*
Nombre maximum de connexions persistantes autorisées à la base de données.

uodbc.max_links *entier*
Nombre maximum de connexions (persistantes ou non), par processus, à la base de données.

3.4.1.16 Directives de configuration des chaînes mulit−octets

mbstring.internal_encoding *string*
mbstring.internal_encoding définit l'encodage interne par défaut.

mbstring.http_input *string*
mbstring.http_input définit l'encodage de reception HTTP par défaut.

mbstring.http_output *string*
mbstring.http_output définit l'encodage d'affichage HTTP par défaut.

mbstring.detect_order *string*
mbstring.detect_order définit l'ordre de détection des encodages par défaut.

mbstring.substitute_character *string*
mbstring.substitute_character définit l'encodage de substitution par défaut : il est utilisé pour les caractères invalides.

3.5 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 configurations 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.

3.5.1 Considérations générales

Un système complètement sûr est une impossiblité 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 demandaient 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 sysèmes de sécurité trop complexes, qui s'érodent au cour du temps.

Un principe qu'il est bon de retenir : Un système est aussi sur que son maillon le plus faible. Si toutes les transactions sont bien notées dans une base, avec confirmation mais que l'utilisateur n'est authentifiée 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 complétement différentes des valeurs qu'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'individu 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". Peut 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.

3.5.2 Binaires CGI

3.5.2.1 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 du binaire PHP, `/secret/doc.html` est utilisé par convention pour spécifier le nom du fichier qui doit être ouvert et interprété par le programe `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.

3.5.2.2 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).

3.5.2.3 −−enable−force−cgi−redirect

Cette option de compilation prévient quiconque d'appeler directement un script avec l'url `http://my.host/cgi−bin/php/secretdir/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−standart **`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.

3.5.2.4 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 repertoire "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".

3.5.2.5 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 .

3.5.3 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 des virus dans les fichiers des utilisateurs. Ou encore, de 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 restraindre l'accès à certains dossiers qui pourront être utilisés par PHP. Vous pouvez aussi des aires de restrictions Apache, pour restraindre les activités anonymes liées aux internautes.

3.5.4 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/password, 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 = $HTTP_POST_VARS['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 souhaiterez utiliser une autre forme d'authentification. Considérez ce qui pourrait se passer si les utilisateurs passent "../etc/" et "passwd" comme arguments! Le code serait éxécuté 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 manoeuvres :

- Limiter les permissions du l'utilisateur web PHP.
- Vérifier toutes les variables liées aux chemins et aux fichiers qui sont fournie.

Voici un script renforcé :

Une vérification renforcée

```
<?php
// Efface un fichier sur le disque où l'utilisateur à le droit d'aller
  $username = $HTTP_SERVER_VARS['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 éffacé!";
?>
```

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 essayez d'écrire un script renforcé :

Vérification de noms de fichier sécurisée

```
<?php
  $username = $HTTP_SERVER_VARS['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 stockages 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.

3.5.5 Rapport d'erreur

En terme de sécurité, il y a deux conséquence au rapport d'erreur. D'un coté, 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 :

Attacque 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 débugger 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 buggé, 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 débuggage, 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 débuggage. Si l'intrus peut comprendre votre

technique habituelle d'utilisation, il peut tenter une attaque frontale sur une page, en envoyant des chaînes de débuggage :

Exploiter des variables classiques de débuggage

```
<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 d'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 d'inactiver le rapport d'erreur, dès que le script est en production. La troisième est d'utiliser les fonctions de gestions 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 dangeureuses de variables. Vous testez votre code en béta−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 vérifée 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");
  }
?>
```

3.5.6 Utilisation des variables HTTP

Une fonctionnalité de PHP qui peut être utilisée pour améliorer la sécurité est de configurer PHP en inactivant 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 interessants.

Travailler avec register_globals actif

```
<?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 actif

```
<?php
  if($HTTP_COOKIE_VARS["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 ($HTTP_COOKIE_VARS['username'] &&
      !$HTTP_POST_VARS['username'] &&
      !$HTTP_GET_VARS['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", $HTTP_SERVER_VARS['REMOTE_ADDR']);
   echo "Problème de sécurité, l'administrateur est alerté.";
   exit;
  }
?>
```

Bien entendu, inactiver 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.

3.5.7 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).

3.5.8 Masquer PHP

Quelques astuces permettent de masquer PHP, et cela entrave les pirates qui recherchent des faiblesses dans votre système. En inactivant 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 type
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 légèr 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.

3.5.9 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.

4 Référence

4.1 La syntaxe de base

4.1.1 Le passage du HTML au PHP

Lorsque PHP commence à traiter un fichier, il ne fait qu'afficher le texte HTML qu'il rencontre. Si vous renommez un fichier .html en .php, il s'affichera exactement comme avant.

Si vous voulez insérer des commandes PHP dans votre fichier, vous devez indiquer à PHP le début d'une telle séquence, en passant en mode PHP. Il y a quatre moyens pour passer du mode HTML au mode PHP :

Le passage du HTML au PHP

```
1.  <? echo ("Ceci est un exemple d'affichage à l'écran en PHP, sous forme d'expression SGML.\n"); ?
    <?= expression ?>  Raccourci de "<? echo expression ?>"
2.  <?php echo("Si vous voulez afficher du XML ou du XHTML, faites comme ceci.\n"); ?>
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 deuxième méthode est généralement utilisée, car elle permet une implémentation aisée de PHP avec la prochaine génération de XHTML.

La première possibilité n'est valable que si vous l'avez activée. Soit en faisant appel à la fonction short_tags() (NdT : semble avoir disparu), soit en utilisant l'option d'exécution short_open_tag dans le fichier de configuration, soit en utilisant l'option de compilation −−enable−short−tags .

La quatrième possibilité est seulement disponible si vous l'avez activée en utilisant soit l'option d'exécution asp_tags , soit en utilisant l'option de compilation −−enable−asp−tags.

Note

Le support de la quatrième possibilité, ASP−style, a été ajouté dans la version 3.0.4.

La marque de fermeture d'un bloc (?>) comprend la nouvelle ligne suivante, s'il y en a une.

PHP vous permet d'utiliser des structures telles que :

Méthode avancée

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

```
  }
?>
```

Cela fonctionne comme on peut s'y attendre, car PHP traite le texte entre ?> et <?php comme une fonction `echo` , sans remplacer les variables éventuelles par leur valeur.

4.1.2 Le séparateur d'instruction

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'un instruction, et donc ajoute implicitement un point virgule. Les deux exemples suivants sont équivalents.

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

4.1.3 Commentaires

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

```
<?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
?>
```

Le premier type de commentaire ne commente que jusqu'à la fin de la ligne ou bien jusqu'à la fin du bloc, cela dépend du premier rencontré.

```
<h1>Ceci est un <?php echo "simple";?> exemple.</h1>
<p>
```

La ligne du dessus affichera 'Ceci est un simple exemple'.

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

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

4.2 Les types

4.2.1 Introduction

PHP supporte les 8 types basiques suivants :

PHP supporte quatres 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 souvant 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.

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 vrai TRUE ou FALSE .

Vous pouvez utiliser les constantes ' TRUE ' et ' FALSE ' pour spécifier une valeur de type `boolean` . Généralement, vous les créerez avec un des operateurs qui retourne une valeur `boolean` , pour le passer à une structure de contrôle .

```
<?php
  if ($action == "show_version"){
// == is an <link linkend="language.operators">operator</link>
// qui retourne un booléen
    echo "La version est 1.23";
  }
// ceci n'est pas nécessaire
  if ($show_separators == true){
```

```
    echo "<hr>\n";
  }
// ceci est plus pratique
  if ($show_separators){
    echo "<hr>\n";
  }
?>
```

Note
Le type booéen a été introduit en PHP 4.

4.2.2.1 Conversion en booléen

Reportez−vous au chapitre " Définition du type " pour plus d'informations sur les conversions.

Lors des conversions de valeurs de type `boolean` , les valeurs suivantes sont considérées comme fausse (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!

4.2.3 Entiers

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

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` .

```
<?php
  $a = 1234; # nombre entier en base 10
  $a = −123; # nombre entier négatif
  $a = 0123; # nombre entier en base 8, octale (équivalent à 83 en base 10)
  $a = 0x12; # nombre entier en base 16, hexadécimale
             # (équivalent à 18 en base 10)
?>
```

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.

Note
En PHP, il n'existe pas de type fraction. `1/2` se transforme en nombre à virgule flottante `0.5` .

4.2.3.1 Dépassement de capacité des entiers

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

```
<?php
  $large_number =  2147483647;
  var_dump($large_number);
// affiche : int(2147483647)

 var_dump( 0x80000000 );
 // affiche : float(2147483648)

var_dump( 25/7 );
// affiche : float(3.5714285714286)
?>
```

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` .

```
<?php
  $million = 1000000;
  $large_number =  50000 * $million;
  var_dump($large_number);
// affiche : float(50000000000)
?>
```

En PHP, il n'y a pas de division entière. `1/2` sera un nombre 1agrave; 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 que 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` .

4.2.3.2 Conversion en entiers

Pour explicitement convertir une valeur en `integer` , 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, un 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.

4.2.3.2.1 booléen

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

4.2.3.2.2 nombre à virgule flottante

Lors de conversion entre 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 `+/− 2.15e+9 = 2^31`), le résultat est indéfinit, 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.

```
<?php
  echo (int) ( (0.1+0.7) * 10 );
// affiche 7!
?>
```

Pour plus d'informations, reportez−vous aux alertes liées aux nombres à virgule flottante.

4.2.3.2.3 From strings

Reportez−vous à la section des conversions de chaînes .

4.2.3.2.4 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:

```php
<?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, `floor((0.1+0.7)*10)` retournera `7` au lieu de `8` car le résultat de la représentation interne est `7.9999999999...` . Tout ceci est lié au fait qu'il est impossible d'exprimer certaines fractions en un nombre fini de chiffres. Par exemple `1/3` s'écrira `0.3333333...` en mode décimal. Ne faites donc jamais confiance aux nombres à virgule jusqu'à leur dernière décimale, et ne comparez jamais ces nombres avec l'opérateur d'égalité. Si vous avez besoin d'une précision particulière, reportez−vous au traitement des nombres de grande taille avec les librairies BC ou GMP .

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 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 a pas à s'en faire pour cela.

4.2.5.1 Syntax

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 guillemets lui−même.

Note
En PHP 3, une alerte sera affichée si cela arrive avec un niveau de rapport d'erreur de `E_NOTICE` .

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

```
<?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
?>
```

4.2.5.1.2 Guillemets doubles

Si la chaîne est entourés de guillemets doubles, PHP va comprendre certaines séquences de caractères :

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

Si vous essayez d'utiliser l'anti−slash sur n'importe quelle autre séquence, l'anti−slash sera affiché dans votre chaîne.

Le plus important des chaînes à guillemets doubles est le fait que les variables qui s'y trouvent seront remplacées par leurs valeurs. 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 de "Here doc" (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 form−feed (`\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 Here doc 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 soit 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 $boisson";
 // Pas OK, car 's' peut entrer dans un nom de variable, et PHP recherche $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
 $fruits = array( 'fraise' => 'rouge' , 'banane' => 'jaune' );
 echo "Une banane est $fruits[banane].";
 // OK. Notez toutefois que pour cet exemple, cela fonctionne
 // différemment. Voyez <literal>$foo[bar]</literal> hors des chaînes
 echo "Ce carré est large de $carre->largeur mètres.";
 // OK
 echo "Ce carré est large de $carre->largeur00 mètres..";
 // pas OK
 // pour résoudre ce problème, voyez la syntaxe complexe.
?>
```

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 :

```
<?php
  $super = 'fantastique';
  echo "C'est { $super}";
// ne fonctionne pas,
// affiche "C'est { fantastique}"
  echo "C'est {$super}";
// fonctionne,
// affiche "C'est fantastique"
  echo "Ce carré a {$square->width}00 centimètres de large.";
  echo "Ceci fonctionne : {$tableau[4][3]}";
  echo "Ceci échoue : {$tableau[foo][3]}";
// pour la même raison que $tableau[bar] ne fonctionne pas hors d'une chaîne

  echo "Essayez plutôt comme ceci : {$tableau['foo'][3]}";
  echo "Vous pouvez même écrire {$objet->valeurs[3]->nom}";
  echo "Voici la valeur de la variable nommée $name: {${$name}}";

// cela fonctionne, mais c'est vivement déconseillé.
// Et pour finir, on peut combiner avec des fonctions
  $inv = 'Bordeaux';
  echo "Je reprendrai bien un verre de cet excellent {${ strrev('niv') }}, hips";

?>
```

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
/* Affectation d'une chaîne. */
  $str = "Ceci est une chaîne";
/* Concaténation. */
  $str = $str . " avec un peu plus de texte";
/* Une autre méthode de concaténation. */
  $str .= " et une nouvelle ligne à la fin.\n";
/*  Cette chaîne finira comme : '<p>Nombre: 9</p>' */
```

```
  $num = 9;
  $str = "<p>Nombre: $num</p>";
/* Celle-ci sera '<p>Nombre: $num</p>' */
  $num = 9;
  $str = '<p>Nombre: $num</p>';
/* Premier caractère d'une chaîne */
  $str = 'Ceci est un test.';
  $first = $str{0};
/* Dernier caractère d'une chaîne. */
  $str = 'This is still a 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 extendues pour les recherches et remplacements.

Il y a aussi les fonctions sur les URL , ainsi que des fonctions de chiffrage (mcrypt et mhash).

Finalement, si vous ne trouvez toujours pas votre bonheure, il y a les fonctions de types de caractères .

4.2.5.3 Conversion de type

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 caractère '.', '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.

```
<?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 Small Pigs";     // $foo est du type  integer (11)
  $foo = 1 + "10 Little Piggies"; // $foo est du type  integer (11)
  $foo = "10.0 pigs " + 1;        // $foo est du type  integer (11)
  $foo = "10.0 pigs " + 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 testez 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.

```
<?php
  echo "\$foo==$foo; type is " . gettype( $foo ) . "<br>\n";
?>
```

4.2.6 Les tableaux

Un tableau PHP est en fait une association ordonnée (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 C array

Un tableau `array` peut être créé avec la fonction `array` . Cette fonction prend en argument des structures `key => value` , séparées par des virgules.

Une clé `key` est soit un entier positif ou bien une chaîne. Si une clé est la représentation standard d'un entier positif, elle sera interprété comme tel. (i.e. `'8'` sera interprété comme `8` , tandis que `'08'` sera interprété comme `'08'`).

Une valeur peut être n'importe quoi.

4.2.6.1.1.1 Omettre des clés

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 spécifiez une qui a déjà été assigné, la nouvelle valeur écrasera la précédente.

```
array( Array value , ... ) // key is either string ou entier integer
positif // value peut être n'importe quoi
```

4.2.6.1.2 La syntaxe à crochets

Vous pouvez aussi modifier un tableau existant en lui assignant simplement des valeurs.

L'assignement 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` .

4.2.6.2 Fonctions pratiques

Il y a toute une panoplie de fonctions pratiques pour travailler avec les tableau : array−functions .

L'élément de langage foreach est spécifiquement dédiée aux tableau : elle permet de passer en revue simplement les valeurs d'un tableau.

4.2.6.3 Exemples

Le type tableau de PHP est très souple. Voici quelques exemples d'utilisation :

```php
<?php
// ceci
  $a = array( 'couleur' => 'rouge'
            , 'gout' => 'sucre'
            , 'forme' => 'rond'
            , 'nom'  => 'pomme'
            ,            4        // cette clé sera 0
          );
// est complétement 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
<?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
// remplis 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"
                                       )
                  , "nombres" => array ( 1
```

```
                                    , 2
                                    , 3
                                    , 4
                                    , 5
                                    , 6
                                    )
                  , "trous"   => array ( "premier"
                                    , 5 => "second"
                                    , "troisième"
                                    )
                );

?>
```

4.2.6.4 Attention aux tableaux

4.2.6.4.1 P $foo[bar]

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

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

Cela est mauvais, mais ca marche. Pourquoi est−ce mauvais? La raison est que PHP réclame une expression entre les crochets (comme indiqué dans la section sur la syntaxe des tableaux). Cela signifie que vous pouvez écrire quelque chose comme :

```
<?php
  echo $arr[ foo(true) ];
?>
```

Ceci est un exemple d'utilisation de retour de fonction dans un index de tableau. PHP reconnait aussi les constantes, et vous pouvez avoir déjà rencontré E_* .

```
<?php
  $error_descriptions[E_ERROR] = "Une erreur fatale est survenue";
  $error_descriptions[E_WARNING] = "PHP a généré une alerte";
  $error_descriptions[E_NOTICE] = "Ceci est juste une note grâcieuse";
?>
```

Notez que E_ERROR est aussi un identifiant vlaide, tout comme `bar` dans le premier exemple. Mais le dernier exemple revient à écrire :

```
<?php
  $error_descriptions[1] = "Une erreur fatale est survenue";
  $error_descriptions[2] = "PHP a généré une alerte";
```

```
  $error_descriptions[8] = "Ceci est juste une note grâcieuse";
?>
```

car E_ERROR égale 1 , etc.

Alors, comment se fait−tile que $foo[bar] fonctionne? C'est parce que bar est attendu comme une constante. Cependant, dans ce cas, aucune constante n'a pour nom bar . PHP suppose alors que vous voulez dire bar littéralement, c'est−à−dire la chaîne "bar" , mais que vous avez oublié les guillements.

4.2.6.4.1.1 Alors, pourquoi est−ce mal?

Dans le futur, l'équipe de développement peut vouloir ajouter une nouvelle constante et vous vous retrouverez dans la panade. Par exemple, vous ne pouvez déjà pas utiliser les constantes empty et default , car ce sont des mots reservés.

Et, pour en mettre une autre couche, cette syntaxe est tout simplement obsolète, et risque de ne plus fonctionner du tout un jour ou l'autre.

Note
Lorsque vous activez le rapport d'erreur avec error_reporting avec E_ALL , PHP générera des notes à chaque fois que cette syntaxe est utilisée. Essayez donc error_reporting(E_ALL); dans vos scripts, pour voir).

Note
Dans une chaîne à guillements doubles, une autre syntaxe est valide. Voyez la section sur Traitement des variables dans les chaînes pour plus de détails.

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 linstance de l'objet.

```
<?php
  class foo {
    function faire_foo () {
        echo "Faisant foo.";
    }
  }
  $bar = new foo;
  $bar−>faire_foo();
?>
```

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 ressource pour une liste exhaustive des fonctions créant et utilisant ces ressources.

Note
Le type ressource a été introduit en PHP 4.

4.2.8.1 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.

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

Note
La valeur NULL a été introduite en PHP 4.

4.2.10 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.

```
<?php
  $foo = "0";  // $foo est une chaîne de caractères (ASCII 48)
  $foo += 2;   // $foo est maintenant du type entier (2)
  $foo = $foo + 1.3;  // $foo est maintenant du type double (3.3)
  $foo = 5 + "10 Petits cochons"; // $foo est du type entier (15)
  $foo = 5 + "10 cochonnets";     // $foo est du type entier (15)
?>
```

Si les deux derniers exemples vous semblent obscurs ou si vous voulez forcer une variable a 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.

```
<?php
  echo "\$foo==$foo; le type est " . gettype( $foo ) . "<br>\n";
?>
```

Note

Le comportement de la conversion automatique est actuellement indéfinie.

```
<?php
  $a = 1;       // $a est un entier
  $a[0] = "f";  // $a devient un tableau, et $a[0] contient "f"
?>
```

Bien que dans l'exemple ci−dessus, il semble évident que $a va devenir un tableau, dont le premier élément est 'f', considérez l'exemple suivant :

```
<?php
  $a = "1";     // $a est une chaîne
  $a[0] = "f";  // Qu'est ce qu'une position dans une chaîne ? que se passe t il?
?>
```

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?

Pour cette raison, aussi bien pour PHP 3.0.12 que PHP 4.0b3−RC4, le résultat de la conversion automatique est considéré comme indéfinie. Des solutions sont en cours de discussion.

4.2.10.1 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").

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

Les conversions autorisées sont:

- (int), (integer) − type entier
- (bool), (boolean) − booléen
- (real), (double), (float) − type double
- (string) − ctype 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:

```
<?php
  $foo = (int) $bar;
  $foo = ( int ) $bar;
?>
```

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

- Conversion en booléen
- Conversion en entier

Attention
Pour transformer facilement une variable en chaîne, entourez la simplement de guillements doubles.

Lors de la conversion d'un tableau en chaîne, le résultat sera le mot `Array` (tableau, en anglais). Lors de la conversion d'un objet en chaîne, le résultat sera le mot `Object` (objet, en anglais). Dans les deux cas, une alerte sera affichée.

Lorsque vous transtypez un scalaire ou une chaîne en tableau, la variable verra son contenu affecté au premier élément du tableau.

```
<?php
  $var = 'ciao';
  $arr = (array) $var;
  echo $arr[0];  // affiche 'ciao'
?>
```

Lorsque vous transtypez un scalaire ou une chaîne en objet, la valeur de la variable sera transformée en attribut de l'objet. L'attribut s'appellera 'scalar':

```
<?php
  $var = 'ciao';
  $obj = (object) $var;
  echo $obj->scalar;  // affiche 'ciao'
?>
```

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 (ie : $x != $X).

Les noms de variables suivent les mêmes règles de nommage que les autres entités PHP. Un nom de variable valide doit commencer par une lettre ou un souligné (_), suivi de lettres, chiffres ou soulignés. Exprimé sous la forme d'une expression régulière, cela donne : '[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*'

Note
Dans nos propos, une lettre peut être une des lettres minuscules (a à z) ou majuscules (A à Z), et les caractères ASCII de 127 à 255 (0x7f-0xff).

```
<?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; 'ï' 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 recopiée 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 terme, "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 :

```
<?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.

```
<?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.
?>
```

4.3.2 Variables prédéfinies

PHP fourni 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
En PHP 4.2.0 et plus récent, le jeu de variables prédéfinies qui sont disponibles dans l'environnement global a changé. Les variables d'entrées et de serveurs ne sont plus **par défaut** placées dans le contexte d'exécution global. Elles sont placées dans les tableaux superglobaux . Vous pouvez toujours forcer l'ancien comportement de PHP en le configurant avec register_globals à `On` votre fichier `php.ini` . Pour plus d'informations et d'explications sur ce changement, voyez l'annonce de PHP 4.1.0 (en anglais).

Depuis la version 4.1.0, PHP fourni 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 peut particuliers, car ils ont automatiquement globaux : ils sont automatiquement disponibles dans tous les environnements d'exécution, sans avoir à utiliser le mot réservé `global` . Pour cette raison, ils sont dits 'auto−globaux' ou bien encore 'superglobaux' (il n'y a pas de mécanisme PHP pour créer de telles variables. Les superglobales sont listées ci−dessous. Cependant, pour connaître le détails de leur contenu, et une présentation approfondie sur les variables prédéfinies PHP, et leur nature, reportez vous à la section variables prédefinies .

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.

$_SESSION
: Les variables qui sont actuellement enregistrées dans la session attachée au script. which are currently registered to a script's session.C'est la nouvelle version de l'ancienne variable `$HTTP_SESSION_VARS` . Voir le chapitre Sessions handling functions pour plus d'informations.

4.3.3 Missing title

Attention
Cette section est la documentation qui avait cours jusqu'en PHP version 4.1.2. Elle est laissée ici pour assurer la transition avec les nouvelles versions, qui ont **abandonné définitivement** leur usage. Nous recommandons vivement au lecteur la section précédente . Pour assurer une meilleure compatibilité de vos scripts avec les nouvelles versions de PHP, n'utilisez plus ces variables.

Malgré ces données, voici une liste de variables prédéfinies, qui seront accessibles avec une installation ad hoc de PHP3, fonctionnant en module, sous Apache 1.3.6.

Pour la liste complète des variables prédéfinies (et d'autres informations pratiques) reportez−vous (et usez) de `phpinfo` .

Note
Cette liste n'est pas exhaustive et ne le sera pas. C'est simplement un aperçu des variables prédéfinies qui peuvent être accessibles dans les scripts.

4.3.3.1 Variables Apache

Ces variables sont créées par le serveur Apache . Si vous utilisez un autre serveur web, il n'est pas sur que celui−ci vous fournira les mêmes variables. Il peut ne pas les fournir, en fournir d'autres. Cependant, un bon nombre de ces variables font partie de l'interface CGI 1.1 , et on peut s'attendre à les retrouver.

Notez que peu d'entre elles seront accessibles lorsque PHP est appelé en ligne de commande, (et elles n'auront alors peut être pas de sens)

$GATEWAY_INTERFACE
: Numéro de révision de l'interface CGI du serveur : i.e. 'CGI/1.1'.

$SERVER_NAME
: Le nom du serveur hôte qui éxécute 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 : 'en'.

$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.

$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_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'.

4.3.3.2 Variables d'environnement

Ces variables sont importées dans l'espace de nom global de PHP, depuis l'environnement sous lequel PHP fonctionne. Beaucoup d'entre elles sont fournies par le shell qui exécute PHP et différents systèmes étant susceptibles de disposer de différents shells, une liste définitive est impossible à établir. Reportez-vous à la documentation de votre shell, pour connaître la liste des

variables d'environnement prédéfinies.

Les autres variables d'environment incluent les variables CGI, placées ici, quelquefois la méthode d'éxécution de PHP (CGI ou module).

4.3.3.3 Variables PHP

Ces variables sont créées par PHP lui_même. Les variables `$HTTP_*_VARS` ne sont disponibles que si l'option de configuration track_vars a été activée. Lorsque c'est le cas, ces variables existent toujours, même si ce sont des tableaux vides. Cela évite les usurpations mal intentionnées de ces variables.

Note
Depuis PHP 4.0.3, track_vars est toujours activé, quelle que soit la configuration.

Si la directive register_globals est activée, alors ces variables seront aussi disponibles comme variables globales du script : c'est−à−dire, indépendamment des tableaux `$HTTP_*_VARS` . Cette fonctionnalité doit être utilisée avec précautions, et de préférence, désactivée. Si `$HTTP_*_VARS` est sécurisé, les équivalents globaux peuvent être écrasés par les données d'entrée de l'utilisateur, avec des intrusions possibles. Si vous ne pouvez pas désactiver register_globals , vous devez prendre toutes les dispositions possibles pour vous assurer que les données utilisées sont sûres.

$argv
: Tableau des arguments passées 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).

$PHP_SELF
: Le nom du fichier du script en cour d'éxécution, par rapport au document root. Si PHP fonctionne en ligne de commande, cette variable n'est pas disponible.

$HTTP_COOKIE_VARS
: Un tableau associatif des variables passées au script courant via les HTTP cookies. Uniquement possible si le suivi des variables a été activé avec la directive générale track_vars ou avec la directive locale `<? php_track_vars ?>` .

$HTTP_GET_VARS
: Un tableau associatif des variables passées au script courant via les HTTP GET. Uniquement possible si le suivi des variables a été activé avec la directive générale track_vars ou avec la directive locale `<? php_track_vars ?>` .

$HTTP_POST_VARS
: Un tableau associatif des variables passées au script courant via les HTTP POST. Uniquement possible si le suivi des variables a été activé avec la directive générale track_vars ou avec la directive locale `<? php_track_vars ?>` .

$HTTP_POST_FILES
: Un tableau associatif contenant les informations sur les fichiers téléchargés avec la méthode HTTP POST. Reportez−vous au chapitre Téléchargement par méthode POST pour plus de détails sur le contenu de `$HTTP_POST_FILES` .

  `$HTTP_POST_FILES` n'est disponible que dans les versions 4.0.0 et plus récentes de PHP.

$HTTP_ENV_VARS
: Un tableau associatif des variables passées au script par l'environnement parent.

$HTTP_SERVER_VARS
: Un tableau associatif des variables passées au script par le serveur HTTP. Ces variables sont analogues aux variables décrites ci−dessus.

4.3.4 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:

```
<?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 :

```
<?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. Par exemple:

```
<?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:

```php
<?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.

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:

```php
<?php
function test() {
    $a = 0;
    echo $a;
    $a++;
}
?>
```

Cette fonction est un peu inutile car à chaque fois qu'elle est appelée, elle initialise $a à 0 et affiche "0". L'incrémentation de la variable ($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:

```php
<?php
function test() {
    static $a = 0;
    echo $a;
    $a++;
}
?>
```

Maintenant, à chaque fois que la fonction Test() est appelée, elle affichera une valeur de $a incrémentée de 1.

Les variables statiques sont essentielles lorsque vous faites des appels récursifs à une fonction. Une fonction récursive est une fonction qui s'appelle elle−même. Il faut faire attention lorsque vous écrivez une fonction récursive car il est facile de faire une boucle infinie. Vous devez vérifier que vous avez bien une condition qui permet de terminer votre récursivité. La fonction suivante compte récursivement jusqu'à 10:

```php
<?php
function test() {
    static $count = 0;
    $count++;
    echo $count;
    if ($count < 10) {
```

```
        test();
    }
    $count--;
}
?>
```

4.3.5 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 affecté avec l'instruction suivante:

```
<?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 "$$" précédent la variable. C'est−à−dire

```
<?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

```
<?php
echo "$a ${$a}";
?>
```

produira le même affichage que :

```
<?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 $$a[1], le parseur a besoin de savoir si vous parler de la variable qui a pour nom $a[1] ou bien si vous voulez l'index [1] de la variable $$a. La syntaxe pour résoudre cette ambiguïté est la suivante: ${$a[1]} pour le premier cas, et ${$a}[1] pour le deuxième.

4.3.6 Variables externes à PHP

4.3.6.1 Formulaires HTML (GET et POST)

Lorsqu'un formulaire est envoyé à un script PHP, toutes les variables du formulaire seront automatiquement disponibles dans le script. Par exemple, considérons le formulaire suivant:

Exemple avec un formulaire simple

```
<form action="foo.php" method="post">
    Nom: <input type="text" name="nom"><br>
    <input type="submit">
</form>
```

Lorsque ce formulaire est envoyé, le PHP va créer la variable `$nom` , qui contiendra la valeur que vous avez entrée dans le champs **Nom:** du formulaire.

Note

La directive de configuration magic_quotes_gpc affecte les valeurs issues des méthodes GET et POST ainsi aue des Cookies. Si cette directive est active, une valeur telle que `It's "PHP!"` sera automagiquement transformée en `It\'s \"PHP!\"` . L'échappement est nécessaire pour les insertions en base de données. Voyez `addslashes` , `stripslashes` et magic_quotes_sybase .

Le PHP permet aussi l'utilisation des tableaux dans le contexte de formulaire, mais seulement des tableaux à une seule dimension. Comme cela, vous pouvez rassembler des variables ou utiliser cette fonctionnalité pour récupérer les valeurs d'un choix multiple :

Variables complexes de formulaire

```
<form action="array.php" 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="medoc">Médoc
        <option value="chablis">Chablis
        <option value="riesling">Riesling
        </select>
    <input type="submit">
</form>
```

Si l'option "track_vars" est activée, soit par l'option de compilation track_vars , soit par la directive de configuration `<? php_track_vars ?>` , les variables transmises par les méthodes POST et GET pourront aussi être trouvées dans le tableau associatif global $HTTP_POST_VARS ou $HTTP_GET_VARS suivant la méthode utlisée.

4.3.6.1.1 Bouton "submit" sous forme d'image

Lorsque vous envoyez le résultat d'un formulaire, vous pouvez utiliser une image au lieu du bouton "submit" standard en utilisant un tag :

```
<input type=image src="image.gif" name="sub">
```

Lorsqu'un utilisateur clique sur l'image, le formulaire sera transmis au serveur avec deux variables de plus, sub_x et sub_y. Ces deux variables contiennent les coordonnées de l'endroit où l'utilisateur à cliqué. Les utilisateurs expérimentés remarqueront que les noms de variables sont transmis avec une virgule à la place du caractère "_", mais le PHP fait la conversion automatiquement.

4.3.6.2 HTTP Cookies

Le PHP supporte les cookies HTTP de manière totalement transparente, comme défini dans les Netscape's Spec . 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 du "header" 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 :

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

4.3.6.3 Variables d'environnement

Le PHP fait en sorte que les variables d'environnement soient accessibles directement comme des variables PHP normales.

```
<?php
echo $HOME;  /* Affiche la valeur de la variable d'environnement HOME,
                si celle−ci est affectée. */
?>
```

Même si le PHP crée les variables lors de l'utilisation des méthodes GET, POST et cookie, il est de temps en temps préférable de transmettre explicitement la valeur de la variable afin d'être sûr de la valeur. La fonction `getenv` peut être utilisée pour récupérer la valeur des variables d'environnement. Vous pouvez aussi affecter une variable d'environnement grâce à la fonction

`putenv` .

4.3.6.4 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 non−guillemetée (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).

4.3.6.5 Détermination du type des variables

Parceque le 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 leur nom le suggère, cette valeur ne peut jamais être modifiée durant l'exécution du script (les constantes magiques `__FILE__` et `__LINE__` sont les seules exception). 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).

Les constantes sont accessibles de manière globale.

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 `boolean` , `integer` , `double` et `string`).

Vous povuez 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, si vous obtenez le nom de cette constante dynamiquement (retour de fonction, par exemple). Utilisez la fonction `get_defined_constants` pour connaître la liste de toutes les fonctions 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.

Definir 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

Les constantes prédéfinies sont toujours disponibles. En voici la liste :

__FILE__ (insensible à la casse)
: Le nom du fichier qui est actuellement exécuté. Si cette constante est utilisée dans le cadre d'un fichier "inclus" (après utilisation de `require`), alors le nom du fichier inclus est renvoyée, et non le nom du fichier parent.

__LINE__ (insensible à la casse)
: Le numéro de la ligne qui est actuellement exécutée. Si cette constante est utilisée dans le cadre d'un fichier "inclus" (après utilisation de `require`), c'est la position dans le fichier inclus qui est renvoyé.

PHP_VERSION
: La chaîne de caractères de présentation de la version du PHP qui est actuellement utilisée. Par exemple '4.1.0'.

PHP_OS
: Nom du système d'exploitation qui est utilisé par la machine qui fait tourner le PHP. Parmi les valeurs possibles : "AIX", "Darwin" (MacOS), "Linux", "SunOS", "WIN32",

"WINNT". Note : cette liste n'est pas exhaustive.

TRUE
: La valeur vraie booléenne, TRUE .

FALSE
: La valeur faux booléenne, FALSE .

E_ERROR
: Dénote une erreur autre qu'une erreur d'analyse ("parse error") qu'il n'est pas possible de corriger.

E_WARNING
: Dénote un contexte dans lequel le PHP trouve que quelque chose ne va pas. Mais l'exécution se poursuit tout de même. Ces alertes−là peuvent être récupérées par le script lui−même. Un exemple serait une expression régulière (regexp) invalide dans la fonction `ereg` .

E_PARSE
: L'analyseur a rencontré une forme syntaxique invalide dans le script. Correction de l'erreur impossible.

E_NOTICE
: Quelque chose s'est produit, qui peut être ou non une erreur. L'exécution continue. Par exemple, le cas de guillemets doubles (") non refermés, ou bien la tentative d'accéder à une variable qui n'est pas encore affectée.

E_ALL
: Toutes les constantes E_* rassemblées en une seule. Si vous l'utilisez avec `error_reporting` , toutes les erreurs et les problèmes que PHP rencontrera seront notifiés.

Les constantes E_* sont généralement utilisées avec la fonction `error_reporting` .

Vous pouvez définir d'autres constantes en utilisant la fonction `define` .

Il est à noter que ce sont des constantes, et non pas des macros comme en C. Seulement les données scalaires peuvent être représentées par des constantes.

Définition de constantes

```
<?php
  define("CONSTANTE", "Bonjour le monde.");
  echo CONSTANTE;
// affiche "Bonjour le monde."
?>
```

Utilisation des constantes __FILE__ et __LINE__

```
<?php
  function rapport_erreur($file, $line, $message) {
    echo "Une erreur est survenue dans le fichier $file à la ligne $line: $message.";
  }
  rapport_erreur(__FILE__, __LINE__, "Il y a un problème!");
?>
```

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 morceau, 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 jeux 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; $b = 5;' (un point virgule marque la fin d'une instruction). Comme les assignations sont analysé de droite à gauche, vous pouvez aussi bien écrire '$b = $a = 5'.

Un autre bon exemple du langage orienté expression est la pré−incrémentation et la post−incrémentation, (ainsi que la décrémentation). Les utilisateurs de PHP/FI 2 et ceux de nombreux autres langages sont habitués à la notation "variable++" et "variable−−". Ce sont les opérateurs d'incrémentation et de décrémentation. En PHP/FI 2, l'instruction '$a++' n'a aucune valeur (c'est−à−dire que ce n'est pas une expression) et vous ne pouvez donc pas l'utiliser. PHP ajoute les possibilités d'incrémentation et de décrémentation comme c'est le cas dans le langage C. En PHP, comme en C, il y a deux types d'opérateurs d'incrémentation (pré−incrémentation et

post−incrémentation). Les deux types d'opérateur d'incrémentation jouent le même rôle (c'est−à−dire qu'ils incrémentent la variable). La différence vient de la valeur de l'opérateur d'incrémentation. L'opérateur de pré−incrémentation, qui s'écrit '++$variable', évalue la valeur incrémentée (PHP incrémente la variable avant de lire la valeur de cette variable, d'où le nom de 'pré−incrémentation'). L'opérateur de post−incrémentation, qui s'écrit '$variable++', évalue la valeur de la variable avant de l'incrémenter (PHP incrémente la variable après avoir lu sa valeur, d'où le nom de 'post−incrémentation').

Un type d'expression très commun est l'expression de comparaison. Ces expressions sont évaluées à 0 ou 1, autrement dit FALSE ou TRUE (respectivement). PHP supporte les opérateurs de comparaison > (plus grand que), => (plus grand ou égal), == (égal à), < (plus petit que), <= (plus petit ou égal). Ces expressions sont utilisées de manière courante dans les instructions conditionnelles, comme l'instruction `if` .

Pour le dernier exemple d'expression, nous allons parler des combinaisons d'opérateurs/assignation. Vous savez que si vous voulez incrémenter la variable $a d'une unité, vous devez simplement écrire '$a++'. Mais si vous voulez ajouter la valeur '3' à votre variable ? Vous pouvez écrire plusieurs fois '$a++', mais ce n'est pas la meilleure des méthodes. Un pratique plus courante est d'écrire '$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'incrémentation 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 vueà−dire dans d'autre langage, 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 troisème sous−expression qui est évaluée et qui constitue le résultat de l'expression.

Les exemples suivants devraient vous permettre de mieux comprendre la pré− et 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−incrémentation de la variable $a et assignation de
                       la valeur à la variable $c */
$e = $d = ++$b;     /* Pré−incrémentation, et assignation de la valeur aux
```

```
                        variables $d et $e  */
/* à ce niveau, les variables $d et $e sont égales à 6 */
$f = double($d++);  /* assignation du double de la valeur de $d à la variable $f ($f vaut 12),
                        puis incrémentation de la valeur de $d  */
$g = double(++$e);  /* assigne deux fois la valeur de $e
<B>après</B>

                        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 qu'elle 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 précédence 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 ("*") à une priorité supérieure par rapport à 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−associative	== != === !==
non−associative	< <= > >=
gauche	<< >>
gauche	+ − .
gauche	* / %
droite	! ~ ++ −− (int) (double) (string) (array) (object) @
droite	[
non−associative	new

4.6.2 Les opérateurs arithmétiques

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

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

L'opérateur de division ("/") retourne une valeur entière (le résultat d'une division entière) si les deux opérandes sont entiers (ou bien des chaînes converties en entier. 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.

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 :

```
<?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:

```
<?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 revéler 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.

Exemple	Nom	Résultat
$a & $b	ET (AND)	Les bits positionnés à 1 dans $a ET dans $b sont positionnés à 1.
$a | $b	OU (OR)	Les bits positionnés à 1 dans $a OU $b sont sont positionnés à 1.
$a ^ $b	Xor	Les bits positionnés à 1 dans $a OU dans $b sont positionnés à 1.
~ $a	NON (Not)	Les bits qui sont positionnés à 1 dans $a sont positionnés à 0, et vice versa.
$a << $b	Décalage à gauche	Décale les bits de $a $b fois sur la gauche (chaque décalage équivaut à une multiplication par 2).
$a >> $b	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

$a == $b	Egal	Vrai si $a est égal à $b.
$a === $b	Identique	Vrai si $a est égal à $b et qu'ils sont de même type (PHP 4 seulement).
$a != $b	Différent	Vrai si $a est différent de $b.
$a <> $b	Différent	Vrai si $a est différent de $b.
$a < $b	Plus petit que	Vrai si $a est plus petit strictement que $b.
$a > $b	Plus grand	Vrai si $a est plus grand strictement que $b.
$a <= $b	Inférieur ou égal	Vrai si $a est plus petit ou égal à $b.
$a >= $b	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
  (expr1) ? (expr2) : (expr3);
?>
```

Cette expression renvoie la valeur de l'expression `expr2` si l'expression `expr1` est vraie, et l'expression `expr3` si l'expression `expr1` est fausse.

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 quelquechose, 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.

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

4.6.8 Opérateurs d'incrémentation/Décrémentation

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

Exemple	Nom	Résultat
++$a	Pré−incrémente	Incrémente $a de 1, puis retourne $a.
$a++	Post−incrémente	Retourne $a, puis l'incrémente de 1.
−−$a	Pré−décrémente	Décrémente $a de 1, puis retourne $a.
$a−−	Post−décrémente	Retourne $a, puis décrémente $a de 1.

Voici un exempla 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";
?>
```

4.6.9 Les opérateurs logiques

Exemple	Nom	Résultat
$a and $b	ET (And)	Vrai si $a ET $b sont vrais.
$a or $b	OU (Or)	Vrai si $a OU $b est vrai
$a xor $b	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'assignations pour plus de détails.

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

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 :

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

Comme nous l'avons vu dans le paragraphe consacré aux expressions, `expr` est évaluée à sa vraie valeur. Si l'expression `expr` est TRUE , PHP exécutera l'instruction et si elle est FALSE , l'instruction sera ignorée.

L'exemple suivant affiche la phrase `a est plus grand que b` si $a est plus grand que $b :

```
<?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 :

```
<?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 oú 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:

```
<?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 aprè un if dans le cas oú le "premier" if est évalué comme FALSE . Mais, à la différence de l'expression else , il n'exécutera l'instruction que si l'expression condionnelle 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 :

```
<?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 s'imbriquent les uns dans 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.

L'expression elseif est exécutée seulement si le if précédent et tout autre elseif précédent est évalués comme FALSE , et que votre elseif est évalué à TRUE .

4.7.4 Syntaxe alternative

Le PHP propose une autre manière de rassembler des instructions à l'intérieur d'un bloc, pour les fonctions de contrôle if , while , for , foreach et switch . Dans chaque cas, le principe est de remplacer l'accolade d'ouverture par deux points (:) et l'accolade de fermeture par, respectivement, endif; , endwhile; , endfor; , ou endswitch; .

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

```
?>
  A vaut 5
<?php
 endif;
 ?>
```

Dans l'exemple ci−desssus, le block 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 `elsif` et un `else` utilisant cette autre syntaxe:

```
<?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;
?>
```

Allez voir 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 :

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

La signification d'une boucle `while` est très simple. Le 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 le 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:

```
```

```
<?php
while (expression): commandes ... endwhile;
?>
```

Les exemples suivants sont identiques, et affichent tous les nombres de 1 à 10:

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

```
<?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:

```
<?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:

```
<?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:

```
<?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 pensez qu'utiliser une expression vide dans une boucle `for` peut être utile parfois.

PHP supporte aussi la syntaxe alternative suivante pour les boucles `for` :

```
<?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:

```
<?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 le 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 supppression des erreurs @ .

Vous pouvez remarquer que les exemples suivants fonctionnent de manière identique :

```
<?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 :

```
<?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 :

```
<?php
/* exemple 1: valeurs seules */
$a = array (1, 2, 3, 17);
foreach ($a as $v) {
   print "Valeur courante de \$a: $v.\n";
}
/* exemple 2: valeurs (avec la clé correspondante) */
$a = array (1, 2, 3, 17);
$i = 0; /* pour l'illustration uniquement */
foreach($a as $v) {
    print "\$a[$i] => $v.\n";
}
/* exemple 3: clé et valeur */
$a = array (
    "un" => 1,
    "deux" => 2,
    "trois" => 3,
```

```
    "dix-sept" => 17
);
foreach($a as $k => $v) {
    print "\$a[$k] => $v.\n";
}
/* exemple 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 5: tableaux dynamique */
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 emboitées ont été interrompues.

```
<?php
$i = 0;
while ($i < 10) {
    if ($arr[$i] == "stop") {
        break;  /* Vous pouvez aussi écrire 'break 1;' ici. */
    }
    $i++;
}
/* Utilisation de l'argument optionnel. */
$i = 0;
while ( ++$i ) {
    switch ( $i ) {
    case 5:
        echo "à 5<br>\n";
        break 1;  /* Ne sort que du switch. */
    case 10:
        echo "à 10; quitting<br>\n";
        break 2;  /* Sort du switch et du 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.

`continue` accepte un argument numérique optionnel qui vous indiquera combien de structures emboitées ont été ignorées.

```
<?php
while (list ($cle, $valeur) = each ($arr)) {
   if (!($cle % 2)) { // évite les membres impairs
       continue;
   }
   fonction_quelconque($valeur);
}
$i = 0;
while ($i++ < 5) {
    echo "Dehors<br>\n";
    while (1) {
        echo "  Milieu<br>\n";
        while (1) {
             echo "  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` .

Les deux exemples suivants sont deux manières différentes d'écrire la même chose, l'une en utilisant une séries de `if` , et l'autre en utilisant l'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 :

```
<?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.

```
<?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 :

```
<?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).

```
<?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'éxécutions dans un bloc de code. La syntaxe de `declare` est similaire à la syntaxe des autres fonctions de contrôle :

```
<?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.

```
<PRE>
<?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);
    }
    $profile[] = microtime ();
  }
// Enregistre un gestionnaire de tick
  register_tick_function("profile");
// Initialise la fonction avant le bloc de déclaration
  profile();
// Exécute un bloc de code, et appele un tick toutes les deux secondes
  declare (ticks=2) {
    for ($x = 1; $x < 50; ++$x) {
        echo similar_text(md5($x), md5($x*$x)), "<br>";
    }
  }
?>
```

Pour voir le résultat :

```
<?php
// Affiche les données de la variable $profile
  print_r(profile(TRUE));
?>
</pre>
```

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 débugger, 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 require

La commande `require` se remplace elle−même par le contenu du fichier spécifié, comme les préprocesseurs C le font avec la commande `#include` .

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 .

`require` n'est pas vraiment une fonction PHP : c'est plus une instruction du langage. Elle ne fonctionne pas comme les fonctions standards. Par exemple, `require` est indépendante des structures de contrôle (cela ne sert à rien de la placer dans une condition, elle sera toujours exécutée). De plus, elle ne retourne aucune valeur. Lire une valeur retournée par un `require` retourne une erreur d'analyse.

Contrairement à `include` , `require` va **toujours** lire dans le fichier cible, même si la ligne n'est jamais exécutée. Si vous souhaitez une inclusion conditionnelle, utilisez `include` . La condition ne va jamais affecter `require` . Cependant, si la ligne de `require` n'est jamais exécutée, le code du fichier ne le sera jamais non plus.

Les boucles n'affectent pas le comportement de `require` . Même si le code contenu dans le fichier source est appelé dans la boucle, `require` n'est exécuté qu'une fois.

Cela signifie qu'on ne peut pas mettre un `require` dans une boucle, et s'attendre à ce qu'il inclue du code à chaque itération. Pour cela, il faut utiliser `include` .

```
<?php
  require('header.inc');
?>
```

Attention : `include` et `require` ajoutent le contenu du fichier cible dans le script lui−même. Elles n'utilisent pas le protocole HTTP ou tout autre protocole. Toute variable qui est dans le champs du script sera accessible dans le fichier d'inclusion, et vice−versa.

```
<?php
  require ("file.inc?varone=1&vartwo=2");
/* Ne fonctionne pas. */
```

```
  $varone = 1;
  $vartwo = 2;
  require ("file.inc");
/* $varone et $vartwo seront accessibles à file.inc */
?>
```

Ne vous laissez pas abuser par le fait que vous pouvez requérir ou inclure des fichiers via HTTP en utilisant la fonctionnalité de gestion des fichiers distants ce qui est au dessus reste vrai.

En PHP 3, il est possible d'exécuter une commande `return` depuis un fichier inclut, tant que cette commande intervient au niveau global du fichier inclus. Elle ne doit intervenir dans aucun bloc (entre accolade {}). En PHP 4, cette possibilité a été supprimée. Si vous en avez besoin, utilisez plutôt `include` .

4.7.14 include

La fonction `include` inclus et évalue le fichier spécifié en argument.

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 .

Cela se produit à chaque fois que la fonction `include` est rencontrée. Donc, vous pouvez utiliser la fonction `include` dans une boucle pour inclure un nombre infini de fois un fichier, ou même des fichiers différents.

```
<?php
$files = array ('premier.inc', 'second.inc', 'troisieme.inc');
for ($i = 0; $i < count($files); $i++) {
    include $files[$i];
}
?>
```

`include` diffère de `require` car le fichier inclus est ré−évaluée à chaque fois que la commande est exécutée, tandis que `require` est remplacée par le fichier cible lors de la première exécution, que son contenu soit utilisé ou non. De plus, cela se fait même s'il est placé dans une structure conditionnelle, comme dans un if).

Parceque la fonction `include` nécessite une construction particulière, vous devez l'inclure dans un bloc si elle est incluse dans une structure conditionnelle.

```
<?php

/*  Ceci est faux, et ne fonctionnera pas comme on l'attend. */
if ($condition)
    include($file);
else
    include($other);
/* Ceci est CORRECT. */
```

```
if ($condition) {
    include($file);
} else {
    include($other);
}
?>
```

En PHP 3, il est possible d'exécuter une commande `return` depuis un fichier inclus, tant que cette commande intervient au niveau global du fichier inclus. Elle ne doit intervenir dans aucun bloc (entre accolade {}). En PHP 4, cette possibilité a été supprimée. Cependant, PHP 4 vous autorise à retourner des valeurs d'un fichier inclus. Vous pouvez traiter `include` comme une fonction normale, qui retourne une valeur. Mais cela génère une erreur d'analyse en PHP 3.

`include` en PHP 3 et PHP 4

On suppose que le fichier `test.inc` existe, et est placé dans le même dossier que le fichier principal :

```
<?php
echo "Avant le retour<br>\n";
if (1) {
    return 27;
}
echo "Après le retour <br>\n";
?>
```

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 .

```
<?php
  include ("file.inc?varone=1&vartwo=2"); /* ne fonctionne pas. */
  $varone = 1;
  $vartwo = 2;
  include ("file.inc");
  /* $varone et $vartwo sont accessibles dans file.inc */
?>
```

Ne vous laissez pas abuser par le fait que vous pouvez requérir ou inclure des fichiers via HTTP en utilisant la fonctionnalité de gestion des fichiers distants ce qui est au dessus reste vrai.

Voir aussi `readfile` , `require` et `virtual` .

4.7.15 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.

Par exemple, si vous créez les deux fichiers d'inclusion `utils.inc` et `foolib.inc`

utils.inc

```
<?php
define(PHPVERSION, floor(phpversion()));
echo "LES GLOBALES SONT SYMPAS\n";
function goodTea() {
    return "Le Earl Grey est délicieux!";
}
?>
```

foolib.inc

```
<?php
require ("utils.inc");
function showVar($var) {
    if (PHPVERSION == 4) {
        print_r($var);
    } else {
        dump_var($var);
    }
}
// Une série de fonctions
?>
```

Puis, vous écrivez un script `cause_error_require.php`

cause_error_require.php

```
<?php
  require("foolib.inc");
/* Ceci génère une erreur*/
  require("utils.inc");
  $foo = array("1",array("complex","quaternion"));
  echo "Ce code requiert utils.inc une deuxième fois, car il est requis \n";
  echo "dans foolib.inc\n";
  echo "Utilisation de GoodTea: ".goodTea()."\n";
  echo "Affichage de foo: \n";
  showvar($foo);
?>
```

Lorsque vous exécutez le script ci−dessus, le résultat sera (sous PHP 4.01pl2):

```
GLOBALS ARE NICE
GLOBALS ARE NICE
Fatal error:  Cannot redeclare causeerror() in utils.inc on line 5
```

En modifiant `foolib.inc` et `cause_errror_require.php` pour qu'elles utilisent `require_once` au lieu de `require` et ne renommant pas le fichier en `avoid_error_require_once.php` , on obtient :

foolib.inc (corrigé)

```
<?php
  require_once("utils.inc");
  function showVar($var) {
?>
```

avoid_error_require_once.php

```
<?php
  require_once("foolib.inc");
  require_once("utils.inc");
  $foo = array("1",array("complexe","quaternion"));
?>
```

L'exécution de ce script, sous PHP 4.0.1pl2, donne :

```
LES GLOBALES SONT SYMPA
Ce code requiert utils.inc une deuxième fois, car il est requis
dans foolib.inc
Utilisation de GoodTea: Le Earl Grey est délicieux!
Affichage de foo:
Array
(
    [0] => 1
    [1] => Array
        (
            [0] => complexe
            [1] => quaternion
        )
)
```

Notez aussi que, de la même manière que les préprocesseurs traitent les `#include` , cette commande est exécutée au moment de la compilation, c'est−à−dire lorsque le script est analysée, et avant qu'il soit exécuté, et ne doit pas être utilisée pour insérer des données dynamiques liées à l'éxécution. Il vaut alors mieux utiliser `include_once` ou `include` .

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.

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

4.7.16 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.

Comme précisé dans la section `require_once` , 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.

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

4.8 Les fonctions

4.8.1 Les fonctions utilisateur

Une fonction peut être définie en utilisant la syntaxe suivante :

```
<?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.

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 défault 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 :

```
<?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 :

```
<?php
function add_some_extra(&$string) {
    $string .= ', et un peu plus.';
}
$str = 'Ceci est une chaîne';
add_some_extra($str);
echo $str;    // affiche 'Ceci est une chaîne, et un peu plus.'
?>
```

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:

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

```
<?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 :

```
<?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 :

```
<?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.

```
<?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.

```php
<?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.

```php
<?php
function &retourne_reference() {
    return $uneref;
}
$newref =&retourne_reference();
?>
```

4.8.4 old_function

L'instruction `old_function` vous permet de déclarer une fonction en utilisant une syntaxe du type PHP/FI2 (au détail près que vous devez remplacer l'instruction 'function' par 'old_function'.)

C'est une fonctionnalité obsolète et elle ne devrait être utilisée que dans le cadre de conversion de PHP/FI2 vers PHP 3

Attention
Les fonctions déclarées comme `old_function` ne peuvent pas être appelées à partir du code interne du PHP. Cela signifie, par exemple, que vous ne pouvez pas les utiliser avec des fonctions comme `usort` , `array_walk` , et `register_shutdown_function` . Vous pouvez contourner ce problème en écrivant une fonction d'encapsulation qui appellera la fonction `old_function` .

4.8.5 Fonctions−variable

PHP supporte le concept de fonctions variables. Cela signifie que si le nom d'une variable est suivi de parenthèses, PHP recherchera une fonction de même nom, et essaiera de l'exécuter. Cela peut servir, entre autre, pour faire des fonctions call−back, des tables de fonctions...

Les fonctions−variables ne peuvent pas fonctionner avec les éléments de langage comme les `echo` , `unset` , `isset` et `empty` . C'est une des différences majeures entre les fonctions PHP et les éléments de langage.

Exemple de fonction variable

```php
<?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' );
?>
```

4.9 Les classes et les objets

4.9.1 L 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 :

```
<?php
class Caddie {
// Eléments de notre panier
    var $items;

// Ajout de $num articles de type $artnr au panier
    function add_item ($artnr, $num) {
        $this->items[$artnr] += $num;
    }

// Suppression de $num articles du type $artnr du panier
    function remove_item ($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return
<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 valable 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 commençants par __ , pour sa propre magie. Il est vivement recommandé de ne pas utilser des noms de fonctions commençants 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.

```php
<?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.

```php
<?php
 $cart = new Caddie;
 $cart->add_item("10", 1);
 $another_cart = new Cart;
 $another_cart->add_item("0815", 3);
?>
```

L'instruction ci−dessus crée l'objet $cart de la class Caddie. La fonction add_idem() est appelée afin d'ajouter l'article numéro 10 dans le panier. 3 articles numéro 0815 sont ajoutés au cart $another_cart.

$cart et $another_cart disposent des fonctions add_item(), remove_item() et de la variable items. Ce sont des fonctions et variables distinctes. Vous pouvez vous représenter les objets comme des dossiers sur votre disque dur. Vous pouvez avoir deux fichiers "lisez−moi.txt" sur votre disque dur, tant qu'ils ne sont pas dans le même répertoire. De même que vous devez alors taper le chemin complet jusqu'au fichier, vous devez spécifier le nom complet de la méthode avant de l'employer : en termes PHP, le dossier racine est l'espace de nom global, et le séparateur de dossier est −> . Par exemple, les noms $cart−>items et $another_cart−>items représentent deux variables distinctes. Notez que le nom de la variable est alors $cart−>items, et non pas $cart−>$items : il n'y a qu'un seul signe $ dans un nom de variable.

```php
<?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

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 ça 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'.

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

L'exemple ci-desssus 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.

```php
<?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

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 a 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.

```
<?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 construteur 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.

```
<?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.

```php
<?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 fonctions 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 basse est appelé, s'il existe. L'exemple ci−dessus affichera 'Je suis le constructeur de A.
' en PHP 4.

```php
<?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 soudaiement devenir le constructeur de la classe B, bien qu'il n'ai 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 O ::

Attention
La documentation suivante n'est valable que pour PHP 4.

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.

```
<?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 proscire 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 faire 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 litéral de votre classe de base dans votre code, vous pouvez utiliser le mot réservé `parent` , qui représente votre classe de base (celle indiqué par `extends` , dans la déclaration de votre classe). En faisant cela, vous éviter 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.

```
<?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 sont 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 focntionnalité 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 pensez à inclure le fichier `classa.inc` dans toutes vos pages, et pas seulement page1.php et page2.php.

4.9.7 L __sleep __wakeup

`serialize` s'assure que votre classe a une méthode avec le nom magique `__sleep` . Si c'est le cas, cette fonction est appelée avant toute linéarisation. Elle peut alors nettoyer l'objet et on s'attend à ce qu'elle retourne un tableau avec la liste des noms de variables qui doivent être sauvées.

Le but de cette fonction `__sleep` est de fermer proprement toute connexion à une base de données, de valider les requêtes, de finaliser toutes les actions commencées. Cette fonction est aussi pratique si vous ave de très grands objets qui n'ont pas besoin d'être sauvé 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 rétablit toutes les connexions aux bases de données, et de recréer les variables qui n'ont pas été sauvées.

4.9.8 Références dans un constructeur

Créer des références dans un constructeur peut conduire à des résultats étranges. Ce tutorial 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('crée dans le constructeur');
    $bar1->echoName();
    $globalref[0]->echoName();
    /* affiche :
    crée dans le constructeur
    crée dans le constructeur
    crée 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 */
?>
```

Apparament, 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 du au fait que l'opérateur "new"ne retourne par de référence, mais retourne une copie.

Note
Il n'y a aucune perte de performance (puisque PHP 4 utilise un compteur de référence) 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 monstreux, auquel cas il est préférable de passer par des références).

Pour prouver ceci, regardez le code suivant :

```
<?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
*/
?>
```

4.10 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 repertoire, 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 foobar())`
- 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 Caractéristiques

5.1 Authentification HTTP avec PHP

Les fonctions d'authentification 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 authentification ("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'authentification. Actuellement, seule l'authentification simple ("Basic") est supportée. Reportez−vous à la fonction `header` pour plus d'informations.

Voici un exemple de script qui force l'authentification du client pour accéder à une page :

Exemple d'authentification HTTP

```
<?php
  if(!isset($PHP_AUTH_USER)) {
    Header("WWW−Authenticate: Basic realm=\"My Realm\"");
    Header("HTTP/1.0 401 Unauthorized");
    echo "Texte à envoyer si le client appuie sur le bouton d'annulation\n";
    exit;
  } else {
    echo "Bonjour $PHP_AUTH_USER.<P>"
    echo "Vous avez entré le mot de passe $PHP_AUTH_PW.<P>"
  }
?>
```

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'authentification (**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 authentification traditionnelle, les variables globales PHP_AUTH ne seront pas assignées si l'authentification externe a été activée pour cette page. Dans ce cas, la variable `$REMOTE_USER` peut être utilisée pour identifier l'utilisateur à l'extérieur.

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'authentification client s'ils recoivent une réponse 401. Cela permet de déconnecter un utilisateur, pour le forcer à ré−entrer 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.

Authentification 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 nom d'utilisateur valide et un mot de passe correct pour accéder à cet
   exit;
  }
  if(!isset($PHP_AUTH_USER) || ($SeenBefore == 1 && !strcmp($OldAuth, $PHP_AUTH_USER)) ) {
   authenticate();
  }
  else {
   echo "Bienvenue $PHP_AUTH_USER<BR>";
   echo "Old: $OldAuth";
   echo "<form action=\"$PHP_SELF\" method=post>\n";
   echo "<input type=hidden NAME=\"SeenBefore\" VALUE=\"1\">\n";
   echo "<input type=hidden NAME=\"OldAuth\" VALUE=\"$PHP_AUTH_USER\">\n";
   echo "<input type=submit value=\"Re Authenticate\">\n"
   echo "</FORM>\n";
  }
?>
```

Ce comportement n'est pas nécessaire par le standard d'authentification 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 authentifié donnera l'accès direct à la ressource. Cependant, l'utilisateur peut utiliser la touche '_' pour détruire les anciennes authentifications.

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.

5.2 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` .

5.3 Gestion des chargements de fichier

5.3.1 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.

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

```
<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>
```

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 :

En PHP 3, les variables suivantes seront définies dans le script de destination, en cas de téléchargement réussi, et en supposant que register_globals est activé dans le fichier `php.ini` . Si track_vars est activé, elles seront aussi disponibles dans le dossier `$HTTP_POST_VARS` . Notez que les noms des variables suivantes supposent que nom du fichier téléchargé est 'userfile', comme présenté dans l'exemple ci−dessus.

- `$userfile` − Le nom temporaire du fichier qui sera chargé sur la machine serveur.
- `$userfile_name` − Le nom du fichier original sur le système de l'envoyeur.
- `$userfile_size` − La taille du fichier envoyé en octets.
- `$userfile_type` − Le type MIME du fichier, si le navigateur a fourni cette information. Par exemple, `"image/gif"` .

Notez que "$userfile" prend la valeur qui est passée dans le champs INPUT de type TYPE=file. Dans l'exemple ci−dessus, nous avons choisi de l'appeler "userfile".

En PHP 4, le comportement est légèrement différent, car c'est la variable d'environnement `$HTTP_POST_FILES` , qui contiendra les informations sur les fichiers téléchargés. Ces informations sont disponibles dans si l'option track_vars est activée, mais track_vars est toujours activée dans les versions de PHP supérieures à la version 4.0.2.

Le contenu du tableau `$HTTP_POST_FILES` décrit ci−dessous. Notez que l'on suppose ici que le nom du fichier téléchargé est ' `userfile` ', comme présenté dans l'exemple ci−dessus :

`$HTTP_POST_FILES['userfile']['name']`
: Le nom du fichier original sur la machine source.

`$HTTP_POST_FILES['userfile']['type']`
: Le type MIME du fichier, si le navigateur a fourni cette information. Par exemple, `"image/gif"` .

`$HTTP_POST_FILES['userfile']['size']`
: La taille du fichier envoyé, en octets.

`$HTTP_POST_FILES['userfile']['tmp_name']`
: Le nom temporaire du fichier qui sera chargé sur la machine serveur.

Les fichiers seront enregistrés par défaut dans le dossier des fichiers temporaires, à moins qu'un autre dossier n'ait été fourni avec la directive de configuration upload_tmp_dir du fichier `php.ini` . Le dossier par défaut du serveur peut être modifié grâce à la variable d'environnement **`TMPDIR`** , de l'utilisateur qui exécute PHP. Sa modification avec `putenv` depuis un script PHP ne fonctionnera pas. Cette variable d'environnement peut aussi être utilisée pour s'assurer que d'autres opérations fonctionnent avec les fichiers téléchargés.

Validation de fichiers téléchargés
Les exemples suivants fonctionnent sur les versions de PHP 3 supérieures à la version 3.0.16, et supérieures à la version 4.0.2 pour PHP 4. Reportez−vous à la section des fonctions pour étudier `is_uploaded_file` et `move_uploaded_file` .

Note
Cela ne fonctionnera **PAS** avec les versions de PHP 4 supérieure à 4.0.2. Cela repose sur des fonctionnalités internes à PHP qui ont évolué après cette version.

Le script PHP qui reçoit le fichier chargé doit pouvoir gérer le fichier de manière appropriée. Vous pouvez utiliser la variable `$file_size` pour recaler tous les fichiers qui sont trop gros ou trop petits. Vous pouvez utiliser la variable `$file_type` pour recaler 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.

Le fichier sera automatiquement effacé du fichier temporaire à la fin du script, s'il n'a pas été déplacé ou renommé.

5.3.2 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 mégaoctets.

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 chargement.

5.3.3 Chargement multiples de fichiers

Il est possible de charger plusieurs fichiers en même temps, et de recevoir les informations adéquates organisées sous forme de tableau. Pour ce faire, il faut utiliser la même syntaxe d'envoi dans le code HTML que pour les sélections ou boîtes à cocher multiples.

Note
Le support du chargement multiple de fichier a été ajouté dans la version 3.0.10.

Chargement multiple de fichier
`<form action="file−upload.html" method="post" enctype="multipart/form−data">` `  Send these files: `

```
  <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 étéF envoyé, les tableaux `$userfile` , `$userfile_name` , et `$userfile_size` seront initialisés (ainsi que $HTTP_POST_VARS). Chaque tableau sera de type numérique, et contiendra les valeurs appropriées pour le chargement des fichiers.

Par exemple, supposons que les noms de fichier `/home/test/review.html` et `/home/test/xwp.out` soient envoyés. Dans ce cas, `$userfile_name[0]` va contenir `review.html` , et `$userfile_name[1]` contiendra `xwp.out` . Similairement, `$userfile_size[0]` contiendra la taille de `review.html` , etc...

`$userfile['name'][0]` , `$userfile['tmp_name'][0]` , `$userfile['size'][0]` , et `$userfile['type'][0]` sont aussi affectés.

5.3.4 Chargement par méthode PUT

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 webmestres 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 des authentifications d'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.

5.4 Utilisation des fichiers à distance

Aussi longtemps que le support de la fonction d'ouverture générique de fichiers ("URL fopen wrapper") est actif lorsque vous configurez PHP (il est inutile de passer explicitement l'option `--disable-url-fopen-wrapper` pour faire la configuration), vous pouvez utiliser des URLs (HTTP et FTP) avec la majorité des fonctions qui utilisent un nom de fichier comme paramètre, ceci incluant les expressions `require` et `include` .

Note
Vous ne pouvez pas utiliser les fichiers distants dans les expressions `include` et `require` sous Windows.

Par exemple, vous pouvez suivre l'exemple suivant pour ouvrir un fichier sur un serveur web distant, analyser les résultats pour extraire les informations dont vous avez besoin, et ensuite l'utiliser dans une requête de base de données, ou simplement éditer les informations dans le style de votre site.

Connaître le titre d'une page distante

```
<?php
  $file = fopen("http://www.php.net/", "r");
  if (!$file) {
    echo "<p>Impossible d'ouvrir le fichier distant.\n";
    exit;
  }
  while (!feof($file)) {
    $line = fgets($file, 1024);
    /* Cela ne fonctionne que si le titre est écrit sur une ligne.*/
    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
<?php
  $file = fopen("ftp://ftp.php.net/incoming/outputfile", "w");
  if (!$file) {
    echo "<p>Impossible d'ouvrir un fichier distant en écriture.\n";
    exit;
  }
  /* Ecriture des données. */
  fputs($file, "$HTTP_USER_AGENT\n");
  fclose($file);
?>
```

Note
Remarque: Vous pouvez avoir l'idée,à partir de l'exemple ci−dessus, d'utiliser la même technique pour écrire sur un log distant, mais comme mentionné ci−dessus vous ne pouvez qu'écrire sur un nouveau fichier en utilisant les fonctions `fopen` avec une URL. Pour faire des log distribués, nous vous conseillons de regarder la partie `syslog` .

5.5 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 de terminer le script, même si le client n'est plus là pour recevoir les informations. Cependant, par défaut, le script sera interrompu, et terminé 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 exécutez 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 expirer après un laps 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 champs de bits, avec les états. Si les deux états sont actifs, l'état retourné prendra la valeur 3.

5.6 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 interpreteur 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.

5.7 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 acteurs, notamment les fournisseurs d'hébergement, utilise le "Safe Mode".

Les directives de configuration qui contrôlent le safe mode sont :

```
safe_mode = Off
open_basedir =
safe_mode_exec_dir =
safe_mode_allowed_env_vars = PHP_
safe_mode_protected_env_vars = LD_LIBRARY_PATH
disable_functions =
```

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 a 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
```

éxécuter 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
```

Si vou 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
```

5.7.1 Functions désactivées par le Safe Mode

Voici une liste non−exhaustive des fonctions désactivées par le Safe Mode .

Fonction	Limitations
`dbmopen`	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
`dbase_open`	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
`filepro`	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..

filepro_rowcount	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
filepro_retrieve	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
Fonctions ifx_	restrictions sql_safe_mode, (!= Safe Mode)
Fonctions ingres_	restrictions sql_safe_mode, (!= Safe Mode)
Fonctions mysql_	restrictions sql_safe_mode, (!= Safe Mode)
pg_loimport	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
posix_mkfifo	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
putenv	Obéis aux directives safe_mode_protected_env_vars et safe_mode_allowed_env_vars. Voir aussi la documentation de putenv
move_uploaded_file	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
chdir	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
dl	Cette fonction est désactivée par le safe−mode
backtick operator	Cette fonction est désactivée par le safe−mode
shell_exec (équivalent fonctionnels des guillemets oblique)	Cette fonction est désactivée par le safe−mode
exec	Vous ne pouvez éxécuter que les programmes qui sont dans le dossier safe_mode_exec_dir. Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier.
system	Vous ne pouvez éxécuter que les programmes qui sont dans le dossier safe_mode_exec_dir. Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier.
passthru	Vous ne pouvez éxécuter que les programmes qui sont dans le dossier safe_mode_exec_dir. Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier.
popen	Vous ne pouvez éxécuter que les programmes qui sont dans le dossier safe_mode_exec_dir. Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier.
mkdir	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..

`rmdir`	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
`rename`	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..
`unlink`	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..
`copy`	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`)
`chgrp`	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
`chown`	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté..
`chmod`	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
`touch`	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..
`symlink`	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)

`link`	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.
`getallheaders`	En Safe Mode, les entêtes commencants par 'authorization' (sensible à la casse) ne seront pas retournés. Attention : cette protection est mise en défaut par la fonction `getallheaders` sur les serveurs AOL!
Toutes les fonctions qui utilisent les wrappers de fichiers.	??

5.8 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 enter le type `CLI SAPI` et les autres SAPI , qui sont expliqués dans ce chapitre.

Le `CLI SAPI` a été publié puor 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 entête HTTP n'est écrits dans le résultat.

  Bien que le `CGI SAPI` fournisse un moyen de supprimer les entêtes HTTP, il n'y a pas moyen d'activer les entêtes 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 `CLI SAPI`	Commentaire
html_errors	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 intérêts. Par conséquent, cette directive est forcée à FALSE .
implicit_flush	TRUE	Il est souhaitable que tout affichage en provenance de `print` , `echo` et consorts soit immédiatement affiché dans le terminal, et non pas placé dans un buffer quelconque. Vous pouvez toujours utiliser la bufferisation de sortie si vous voulez retarder un affichage, ou bien en manipuler ultimement le contenu.
max_execution_time	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.
register_argc_argv	TRUE	Les variables globales de PHP `$argc` (numbre d'arguments passés à l'application) et `$argv` (tableau contenant les arguments passés à l'application) sont toujours enregistré et rempli avec les données appropriées, lorsque vous utilisez le `CLI SAPI` .

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	Déscription
`STDIN`	Un descripteur de fichier déjà disponible vers `stdin` . Cela évite de l'ouvrir avec `$stdin = fopen('php://stdin', 'r');`
`STDOUT`	Un descripteur de fichier déjà disponible vers `stdout` . Cela évite de l'ouvrir avec `$stdout = fopen('php://stdout', 'w');`
`STDERR`	Un descripteur de fichier déjà disponible vers `stdout` . Cela évite de l'ouvrir avec `$stderr = fopen('php://stderr', 'w');`

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 monter 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
CGI SAPI supporte le même comportement de CLI SAPI , en lui passant l'option -C , lorsque vous l'invoquez en ligne de commmande.

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) exécute le script contenu dans le fichier `mon_script.php` . Vous pouvez choisir n'importe quel fichier, et ces fichiers ne sont pas tenus d'utiliser l'extension `.php` . N'importe quelle extension peut faire l'affaire.
2. Donner du code PHP à exécuter directement en ligne de commande.

   ```
   php -r 'print_r(get_defined_constants());'
   ```

   Une attention particulière doit alors être apportée aux variables d'environnement, qui seront remplacées, et aux guillements, 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.
3. 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 recevoirs. 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'atteidrez 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 voter 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
`-s`	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 entêtes. `<code> [...] </code>` **Note** Cette option ne fonctionne pas avec l'option `-r` .

−w	Affiche la source sans les commentaires et les espaces. Note Cette option ne fonctionne pas avec l'option `−r` .
−f	Analyse et exécute le fichier donné après l'option `−f` . Cette optione est optionnelle, et peut être omise. Le seul nom du fichier est suffisant.

−v	Affiche les versions de PHP, PHP SAPI, et Zend sur le terminal. Par exemple :
	
	`$ 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`

−c	Avec cette option, vous pouvez spécifier le nom du dossier dans lequel PHP va recherche le fichier `php.ini` , ou encore vous pouvez spécifier un fichier de configuration directement (qui n'a pas obligatoirement à être appelé `php.ini`) :
	
	`$ php −c /custom/directory/ mon_script.php`
	
	`$ php −c /custom/directory/custom−file.ini mon_script.php`

−a	Exécute PHP de manière interactive.

−d	Cette option permet de modifier n'importe quelle directive de configuration du fichier `php.ini` . La syntaxe est :
	
	`−d configuration_directive[=value]`
	
	Exemples:
	
	`# 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"`

−e	Génére des informations étendues pour le profilage et le débogage.
−z	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 `/etc/ld.so.conf` sous Linux). Passer un nom de fichier avec le chemin

	complet fera que PHP utilisera ce fichier, sans recherche dans les dossiers classiques. Un chemin de dossier relatif indiquera à PHP qu'il doit chercher uniquement les extensions dans ce dossier.
−l	Cette option permet de faire une vérification syntaxique sur le code PHP fourni. En cas de réussite, le message `No syntax errors detected in <filename>` (`Aucune d'erreur de syntaxe n'ont été détectées dans <nom_de_fichier>` est affiché sur la sortie standard, et le script shell retourne `0` . En cas d'erreur, le message `Errors parsing <filename>` (`Erreurs d'analyse dans le fichier <filename>`) est affiché, en plus des messages d'erreurs détectés par l'analyseur lui même. Le script shell retourne le code `255` . Cette option ne détecte pas les erreurs fatales (comme par exemple les fonctions non définies). Utilisez `−f` si vous voulez tester aussi ces erreurs. Note Cette option ne fonctionne pas avec l'option `−r` .

−m	Cette option liste les extensions PHP et Zend compilées et chargées :
	
	`$ php −m`
	`[PHP Modules]`
	`xml`
	`tokenizer`
	`standard`
	`session`
	`posix`
	`pcre`
	`overload`
	`mysql`
	`mbstring`
	`ctype`
	
	`[Zend Modules]`

−i	Cette option appelle la fonction `phpinfo` , et affiche le résultat. Si `PHP` ne fonctionne pas correctement, il est recommandé d'utiliser la commande `php −i` et de voir si il n'y a pas d'erreurs affichées avant ou après la table d'information. N'oubliez pas que le résutltat de cette option est au format `HTML` , et donc grand et illisible directement.

−r	Cette option permet l'exécution de `PHP` directement dans la ligne de commande. Les balises de `PHP` (`<?php` et `?>`) **ne sont pas** demandées, et coauseront 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 :
	
	`$ 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"]=>`
	`[...]`
	
	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.

-h	Avec cette option, vous pouvez obtenir des informations sur la liste des options supporté par votre PHP, avec des explications succintes.

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 facilements 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'entêtes 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 avnos 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 exécuétable, 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.

6 Index des fonctions

6.1 Apache

6.1.1 Introduction

Ces fonctions sont disponibles lorsque PHP est utilisé comme module Apache 1.x.

6.1.2 Installation

Pour savoir installer PHP pour Apache 1.x, reportez vous à Apache section , dans le chapitre d'installation.

6.1.3 Configuration à l'exécution

Le comportement du module PHP d'Apache est configurable dans le fichier 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 `.htaccess`
`php_flag engine off`

Nom	Défaut	Modifiable	Fonction
engine	On	PHP_INI_ALL	Active ou désactive l'analyse des fichiers PHP
child_terminate	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`
last_modified	Off	PHP_INI_ALL	Envoie la date de modification des scripts PHP dans l'entête HTTP Last−Modified:
xbit_hack	Off	PHP_INI_ALL	Analyse avec PHP les fichiers exécutables, indépendemment de leur extension.

Voici quelques détails sur les directives de configurations :

`engine booléen`
: Cette directive est utile uniquement pour le module Apache. Elle est utilisée par les sites qui souhaites activer ou désactiver PHP, au cas par cas, par dossier ou par dossier virtuels. En utilisant `engine off` au bon endroit dans le fichier `httpd.conf` , PHP peut être activé ou désactivé.

6.1.4 Types de ressources

Cette extension ne définit aucune ressource.

6.1.5 Constantes Prédefinies

Cette extension ne définit aucune constante.

- Introduction
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- apache_child_terminate
- apache_lookup_uri
- apache_note
- apache_request_headers
- apache_response_headers
- apache_setenv
- ascii2ebcdic
- ebcdic2ascii
- getallheaders
- virtual

6.1.7 apache_lookup_uri() : Effectue une requête partielle pour l'URI spécifiée et renvoie toutes les informations.

`object` **apache_lookup_uri** `( string filename )`
[lien librairie]

`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 le PHP est installé sous la forme d'un module Apache.

6.1.8 apache_note() : Affiche ou affecte le paramètre "apache request notes".

string **apache_note** (string note_name , *string note_value*)
[lien librairie]

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 .

6.1.9 apache_request_headers() : Lit tous les entêtes HTTP de la requête

array **apache_request_headers** (void)
[lien librairie]

apache_request_headers retourne un tableau associatif avec tous les entêtes 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'environment disponibles.

6.1.10 apache_response_headers() : Lit tous les entêtes de réponse HTTP

array **apache_response_headers** (void)
[lien librairie]

apache_response_headers retourne un tableau avec tous les entêtes de réponse Apache. Cette fonctionnalité n'est disponible qu'en PHP version 4.3.0 et plus récent.

Voir aussi getallheaders et headers_sent .

6.1.11 apache_setenv() : Modifie une variable de subprocess_env Apache

int **apache_setenv** (string variable , string value , *bool walk_to_top*)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.1.12 ascii2ebcdic() : Transforme une chaîne ASCII en EBCDIC

int **ascii2ebcdic** (string ascii_str)
[lien librairie]

ascii2ebcdic est une fonction spéficique à Apache, qui n'est disponible que sur les OS 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

6.1.13 ebcdic2ascii() : Transforme une chaîne EBCDIC en ASCII

int **ebcdic2ascii** (string ebcdic_str)
[lien librairie]

ebcdic2ascii est une fonction spéficique 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

6.1.14 getallheaders() : Récupère toutes les en−têtes des requêtes HTTP.

array **getallheaders** (void)
[lien librairie]

getallheaders renvoie un tableau associatif de toutes les en−têtes HTTP de la requête courante.

Note
Vous pouvez récupérer la valeur d'une variable d'une CGI en la lisant à partir des variables d'environnement, ce qui fonctionne aussi bien dans le cas d'une installation en module ou d'une

installation en CGI. Utilisez la fonction `phpinfo` pour avoir une liste de toutes les variables d'environnement disponibles.

Exemple avec `getallheaders`

```
<?php
  $headers = getallheaders();
  while (list($entete, $valeur) = each($headers)) {
    echo "$entete: $valeur<BR>\n";
  }
?>
```

Cet exemple est un exemple d'affichage de toutes les en−têtes de la requête courante.

Note

`getallheaders` ne fonctionne que si PHP est installé comme module `Apache` .

6.1.15 virtual() : Effectue une sous−requête Apache

`int` **virtual** (`string filename`)
[lien librairie]

`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 .shtml 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 une en−tête valide, c'est−à−dire, au minimum une 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.

6.2 Tableaux

6.2.1 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.

6.2.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.2.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.2.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.2.5 Types de ressources

Cette extension ne définit aucune ressource.

6.2.6 Constantes Prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

CASE_LOWER (integer)
: CASE_LOWER est utilisé avec `array_change_key_case` et sert à convertir tous les index d'un tableau en minuscules. C'est aussi le comportement par défaut de `array_change_key_case` .

CASE_UPPER (integer)
: CASE_UPPER est utilisé avec `array_change_key_case` et sert à convertir tous les index d'un tableau en majuscule.

6.2.7 Voir aussi

Voir aussi `is_array` , `explode` , `implode` , `split` , `preg_split` et `join` .

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- Voir aussi
- array_change_key_case
- array_chunk
- array_count_values
- array_diff
- array_fill
- array_filter
- array_flip
- 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

6.2.9 array_chunk() : Sépare un tableau en tableaux de taille inférieure

`array` **array_chunk** (`array input` , `int size` , *`bool preserve_keys`*)
[lien librairie]

`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, commencant 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
        )

)
```

6.2.10 array_count_values() : Compte le nombre de valeurs dans un tableau

`array` **array_count_values** (`array input`)
[lien librairie]

`array_count_values` retourne un tableau contenant les valeurs du tableau `input` comme clés et leurs fréquence comme valeur.

Exemple avec `array_count_values`

```php
<?php
  $array = array(1, "bonjour", 1, "monde", "bonjour");
  array_count_values($array);
  // retourne array(1=>2, "bonjour"=>2, "monde"=>1)
?>
```

Note

`array_count_values` a été ajoutée en PHP 4.0.

6.2.11 array_diff() : Calcule la différence entre deux tableaux

`array` **array_diff** (`array array1` , `array array2` , *`array ...`*)
[lien librairie]

`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
<?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_intersect` .

6.2.12 array_fill() : Remplis un tableau avec une même valeur

`array` **array_fill** `( int start_index , int num , mixed value )`
[lien librairie]

`array_fill` crée un tableau avec `num` entrées de la valeur `value` . Les index commecent à la valeur `start_index` .

Exemple avec `array_fill`

```php
<?php
  $a = array_fill(5, 6, 'banane');

/*
  $a est le tableau suivant

  $a[5]  = "banane";
  $a[6]  = "banane";
  $a[7]  = "banane";
  $a[8]  = "banane";
  $a[9]  = "banane";
  $a[10] = "banane";
*/
?>
```

6.2.13 array_filter() : Filtre les éléments d'un tableau

`array` **array_filter** `( array input , mixed callback )`
[lien librairie]

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 pair($var) {
   return ($var % 2 == 1);
  }
  function impair($var) {
   return ($var % 2 == 0);
  }
  $array1 = array("a"=>1, "b"=>2, "c"=>3, "d"=>4, "e"=>5);
  $array2 = array(6, 7, 8, 9, 10, 11, 12);
  $tableau_pair = array_filter($array1, "pair");
  $tableau_impair = array_filter($array2, "impair");
?>
```

Cet exemple montre comment extraire les nombres pairs dans $tableau_impair : ce dernier contient array ("a"=>1, "c"=>3, "e"=>5); , et les nombres impairs dans $tableau_pair : ce dernier contient array (6, 8, 10, 12); ,

Voir aussi array_map et array_reduce .

6.2.14 array_flip() : Remplace les clés par les valeurs, et les valeurs par les clés

array **array_flip** (array trans)
[lien librairie]

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
  $trans = array_flip ($trans);
  $original = strtr ($str, $trans);
?>
```

array_flip example : collision

```
<?php
  $trans = array ("a" => 1, "b" => 1, "c" => 2);
  $trans = array_flip ($trans);
// et $trans vaut : array(1 => "b", 2 => "c");
?>
```

Note

array_flip a été ajoutée en PHP 4.0.

6.2.15 array_intersect() : Calcule l'intersection de tableaux

array **array_intersect** (array array1 , array array2 , *array ...*)
[lien librairie]

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.

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!

Exemple avec array_intersect

```
<?php
  $array1 = array ("a" => "vert", "rouge", "bleu");
  $array2 = array ("b" => "vert", "jaune", "rouge");
  $result = array_intersect ($array1, $array2);
?>
```

$result contient array ("a" => "vert", "rouge"); .

Voir aussi array_diff .

6.2.16 array_key_exists() : Vérifie si une clé éxiste dans un tableau

bool **array_key_exists** (mixed key , array search)
[lien librairie]

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
<?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` et `array_search` .

6.2.17 array_keys() : Retourne toutes les clés d'un tableau

`array` **array_keys** (`array input` , *`mixed search_value`*)
[lien librairie]

`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
<?php
  $array = array(0 => 100, "couleur" => "rouge");
  array_keys($array);
// retourne array(0, "couleur")
  $array = array("bleu", "rouge", "vert", "bleu", "bleu");
  array_keys($array, "bleu");
//  retourne  array(0, 3, 4)
  $array = array( "couleur" => array("bleu", "rouge", "vert"),
                  "taille"  => array("petit", "moyen", "grand") );
  array_keys($array);
//  retourne array("couleur", "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
<?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` .

6.2.18 array_map() : Applique sur fonction sur des tableaux

`array` **array_map** (`mixed callback` , `array arr1` , *`array ...`*)
[lien librairie]

`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);
?>
```

Avec cet exemple, la variable $b contiendra `array (1, 8, 27, 64, 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);
// Affichera :
// 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
// )
  $d = array_map("map_espagnol", $a , $b);
  print_r($d);
// Affichera :
// Array
// (
//     [0] => Array
//         (
//             [1] => uno
//         )
//
```

```
//     [1] => Array
//         (
//             [2] => dos
//         )
//
//     [2] => Array
//         (
//             [3] => tres
//         )
//
//     [3] => Array
//         (
//             [4] => cuatro
//         )
//
//     [4] => Array
//         (
//             [5] => cinco
//         )
//
// )
?>
```

Généralement, lorsque vous utilisez plusieurs tableaux, ils doivent être de même longueur, car la fonction de callback est appliqué à un élément de chaque tableau. Si les tableaux sont de taille inégale, les plus petits seront complétés avec des éléments vides.

Une utilisation interessante de cette fonction est de construire des tableaux de tableaux, grâce à la fonction de callback `NULL` .

`array_map` − création d'un tableau de tableaux

```
<?php
  $a = array(1, 2, 3, 4, 5);
  $b = array("un", "deux", "trois", "quatre", "cinq");
  $c = array("uno", "dos", "tres", "cuatro", "cinco");
  $d = array_map(null, $a, $b, $c);
  print_r($d);
// affichera :
// Array
// (
//     [0] => Array
//         (
//             [0] => 1
//             [1] => un
//             [2] => uno
//         )
//
//     [1] => Array
//         (
//             [0] => 2
//             [1] => deux
//             [2] => dos
//         )
//
//     [2] => Array
//         (
//             [0] => 3
//             [1] => trois
//             [2] => tres
//         )
//
//     [3] => Array
```

```
//         (
//             [0] => 4
//             [1] => quatre
//             [2] => cuatro
//         )
//
//     [4] => Array
//         (
//             [0] => 5
//             [1] => cinq
//             [2] => cinco
//         )
//
// )
?>
```

Voir aussi `array_filter` et `array_reduce` .

6.2.19 array_merge_recursive() : Combine plusieurs tableaux ensembles, récursivement

`array` **array_merge_recursive** (`array array1` , `array array2` , *`array ...`*)
[lien librairie]

`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` .

6.2.20 array_merge() : Rassemble plusieurs tableaux

`array` **array_merge** (`array array1` , `array array2` , *`array ...`*)
[lien librairie]

`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);`
`  $array2 = array ("a", "b", "couleur" => "vert", "forme" => "trapézoïde");`
`  array_merge ($array1, $array2);`
`?>`

Note
`array_merge` a été ajoutée dans PHP 4.0.

6.2.21 array_multisort() : Tri multi−dimensionnel

`boolean` **array_multisort** (`array ar1` , *`mixed arg`* , *`mixed ...`* , *`array ...`*)
[lien librairie]

`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.

`array_multisort` retourne TRUE en cas de succès, FALSE sinon.

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 multidimensionnel

```
<?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).

6.2.22 array_pad() : Complète un tableau jusqu'à la longueur spécifiée, avec une valeur.

`array` **array_pad** (`array input` , `int pad_size` , `mixed pad_value`)
[lien librairie]

`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
?>
```

6.2.23 array_pop() : Dépile un élément de la fin d'un tableau

`mixed` **array_pop** (`array array`)
[lien librairie]

`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` .

Exemple avec `array_pop`

```
<?php
  $stack = array("orange", "pomme", "framboise");
  $fruit = array_pop($stack);
?>
```

Voir aussi `array_push` , `array_shift` et `array_unshift` .

Note

`array_pop` a été ajoutée en PHP 4.0.

6.2.24 array_push() : Empile un ou plusieurs éléments à la fin d'un tableau

`int` **array_push** (`array array` , `mixed var` , *`mixed ...`*)
[lien librairie]

`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;
?>
```

repeté pour chaque `var` .

`array_push` retourne le nouveau nombre d'éléments du tableau.

Exemple avec `array_push`

```
<?php
  $stack = array (1, 2);
  array_push($stack, "+", 3);
?>
```

Cet exemple fait que $stack a 4 éléments: 1, 2, "+", et 3.

Voir aussi `array_pop` , `array_shift` et `array_unshift` .

Note
`array_push` a été ajoutée en PHP 4.0.

6.2.25 array_rand() : Prend une ou plusieurs valeurs, au hasard dans un tableau

`mixed` **array_rand** (`array input` , *`int num_req`*)
[lien librairie]

`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` spéficie 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.

N'oubliez pas d'appeler `srand` pour initialiser le générateur de nombres aléatoires.

Exemple avec `array_rand`

```
<?php
  srand ((double) microtime() * 10000000);
  $input = array ("Neo", "Morpheus", "Trinitée", "Cypher", "Tank");
  $rand_keys = array_rand ($input, 2);
  print $input[$rand_keys[0]]."\n";
  print $input[$rand_keys[1]]."\n";
?>
```

6.2.26 array_reduce() : Réduit itérativement un tableau

`mixed` **array_reduce** (`array input` , `mixed callback` , *`int initial`*)
[lien librairie]

`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 `intial` 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 et array_map .

6.2.27 array_reverse() : Renverse l'ordre des éléments d'un tableau

array **array_reverse** (array array , *boolean preserve_keys*)
[lien librairie]

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);
?>
```

Au final, $result et $result_keyed contiennent tous les deux array ("rouge", "vert"), 4.0, "php" , dans cet ordre. Mais $result_keyed[0] vaut toujours "php" .

Note

array_reverse a été ajoutée en PHP 4.0 Beta 3.

Note

Le second paramètre preserve_keys a été ajouté en PHP 4.0.3.

6.2.28 array_search() : Recherche dans un tableau la clé associée à une valeur

mixed **array_search** (mixed needle , array haystack , boolean strict)
[lien librairie]

array_search recherche needle dans haystack et retourne la clé associée s'il la trouve, ou FALSE sinon.

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 .

Voir aussi in_array .

6.2.29 array_shift() : Dépile un élément au début d'un tableau

`mixed` **array_shift** (`array array`)
[lien librairie]

`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` .

Exemple avec `array_shift`

```
<?php
  $args = array("−v", "−f");
  $opt = array_shift($args);
?>
```

Cet exemple aura pour résultat que $args ne contiendra plus que "−f", et $opt contient "−v".

Voir aussi `array_unshift` , `array_push` et `array_pop` .

Note

`array_shift` a été ajoutée en PHP 4.0.

6.2.30 array_slice() : Extrait une portion de tableau

`array` **array_slice** (`array array` , `int offset` , *`int length`*)
[lien librairie]

`array_slice` retourne une série d' élément 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.

Exemple avec `array_slice`

```
<?php
  $input = array("a", "b", "c", "d", "e");
  $output = array_slice($input, 2);      // retourne "c", "d", et "e"
// les trois exemples suivants sont équivalents
  $output = array_slice($input, 2, 2);  // retourne "c", "d"
  $output = array_slice($input, 2, −1);  // retourne "c", "d"
// Equivalent à :
  $offset = 2; $length = −1;
  $output = array_slice($input, 2, count($input) − $offset + $length);
// 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` .

Note
`array_slice` a été ajoutée en PHP 4.0.

6.2.31 array_splice() : Efface et remplace une portion de tableau

`array` **array_splice** (`array input` , `int offset` , *`int length`* , *`array replacement`*)
[lien librairie]

`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(`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 :

```
<?php
  array_push($input, $x, $y)     array_splice($input, count($input), 0, array($x, $y))
  array_pop($input)              array_splice($input, −1)
  array_shift($input)            array_splice($input, 0, 1)
  array_unshift($input, $x, $y)  array_splice($input, 0, 0, array($x, $y))
  $a[$x] = $y                    array_splice($input, $x, 1, $y)
?>
```

`array_splice` retourne le tableau des éléments supprimés.

Exemples avec `array_splice`

```
<?php
// cas simple
  $input = array("rouge", "vert", "bleu", "jaune");
  array_splice($input, 2);
// $input est array("rouge", "vert")
// nombre d'éléments négatif
  $input = array("rouge", "vert", "bleu", "jaune");
  array_splice($input, 1, −1);
// $input est array("rouge", "jaune")
// avec un élément de remplacement
  $input = array("rouge", "vert", "bleu", "jaune");
```

```
  array_splice($input, 1, count($input), "orange");
// $input est array("rouge", "orange")
//  cas complexe
  $input = array("rouge", "vert", "bleu", "jaune");
  array_splice($input, -1, 1, array("noir", "marron"));
// $input est array("rouge", "vert",
//          "bleu", "noir", "marron")
?>
```

Voir aussi `array_slice` .

Note
`array_splice` a été ajoutée en PHP 4.0.

6.2.32 array_sum() : Calcule la somme des valeurs du tableau

`mixed` **array_sum** `( array arr )`
[lien librairie]

`array_sum` retourne la somme des valeurs du tableau, sous forme d'un entier ou d'un nombre à virgule flottante.

Exemple avec `array_sum`

```
<?php
  $a = array(2,4,6,8);
  echo "somme(a) = ".array_sum($a)."\n";
// affiche : somme(a) = 20
  $b = array("a"=>1.2,"b"=>2.3,"c"=>3.4);
  echo "somme(b) = ".array_sum($b)."\n";
// affiche : somme(b) = 6.9
?>
```

6.2.33 array_unique() : Dédoublonne un tableau

`array` **array_unique** `( array array )`
[lien librairie]

`array_unique` prend le tableau `array` et retourne un nouveau tableau, complètement dédoublonné.

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!

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,"3",3,"4",4,4);
  $result = array_unique($input);
  print_r($result);
// Cela va afficher :
//Array
//(
//  [0] => 3
//  [1] => 3
//  [2] => 4
//  [3] => 4
//)
?>
```

6.2.34 array_unshift() : Empile un ou plusieurs éléments au début d'un tableau

`int` **array_unshift** (`array array` , `mixed var` , *`mixed ...`*)
[lien librairie]

`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
  $queue = array("p1", "p3");
  array_unshift($queue, "p4", "p5", "p6");
?>
```

Le résultat de cet exemple est que $queue aura 5 éléments, à savoir: "p4", "p5", "p6", "p1", et "p3".

Voir aussi `array_shift` , `array_push` , et `array_pop` .

Note

`array_unshift` a été ajoutée en PHP 4.0.

6.2.35 array_values() : Retourne les valeurs d'un tableau

`array` **array_values** (`array input`)
[lien librairie]

`array_values` retourne les valeurs du tableau `input` .

Exemples avec `array_values`

```
<?php
  $array = array("taille" => "XL", "couleur" => "or");
  array_values($array);    // // retourne array("XL", "or")
?>
```

Note

`array_values` a été ajoutée en PHP 4. Ci−dessous, voici une implémentation pour ceux qui utilisent toujours PHP 3.

Implémentation de `array_values` pour les utilisateurs PHP 3

```
<?php
  function array_values($arr){
    $t = array();
    while (list($k, $v) = each($arr)){
        $t[] = $v;
    }
    return $t;
  }
?>
```

6.2.36 array_walk() : Exécute une fonction sur chacun des membres d'un tableau.

`int` **array_walk** (`array arr` , `string func` , `mixed userdata`)
[lien librairie]

`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 `str2lower` , il faut absolument passer par une fonction utilisateur.

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`

```
<?php
  $fruits = array ("d"=>"citron", "a"=>"orange", "b"=>"banane", "c"=>"pomme");
  function test_alter (&$item1, $key, $prefix) {
    $item1 = "$prefix: $item1";
  }
  function test_print ($item2, $key) {
    echo "$key. $item2<br>\n";
  }
  array_walk ($fruits, 'test_print');
  reset ($fruits);
  array_walk ($fruits, 'test_alter', 'fruit');
  reset ($fruits);
  array_walk ($fruits, 'test_print');
?>
```

Voir aussi `each` et `list` .

6.2.37 array() : Crée un tableau

`array` **array** (*`mixed ...`*)
[lien librairie]

`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 litéraux, 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"),
    "numbres" => array (1, 2, 3, 4, 5, 6),
    "trous"   => array ("premier", 5 => "deuxième", "troisième")
  );
?>
```

Index automatique d'un tableau avec `array`

```php
<?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
<?php
  $firstquarter  = array(1 => 'Janvier', 'Février', 'Mars');
  print_r($firstquarter);
?>
```

qui affichera :

```
Array
(
[1] => 'Janvier'
[2] => 'Février'
[3] => 'Mars'
)
```

Voir aussi `list` .

6.2.38 arsort() : Trie un tableau en ordre inverse

`void` **arsort** (`array array`)
[lien librairie]

`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
<?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.

Voir aussi `array_multisort` , `asort` , `krsort` , `ksort` , `natsort` , `natcasesort` , `rsort` , `sort` , `uasort` , `uksort` et `usort` .

6.2.39 asort() : Trie un tableau en ordre

`void` **asort** `( array  array )`
[lien librairie]

`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`

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

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.

Voir aussi `array_multisort` , `arsort` , `krsort` , `ksort` , `natsort` , `natcasesort` , `rsort` , `sort` , `uasort` , `uksort` et `usort` .

6.2.40 compact() : Crée un tableau contenant les variables et leur valeur

`array` **compact** (`string|array varname` , *`mixed ...`*)
[lien librairie]

`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 = "San Francisco";
  $etat = "CA";
  $evenement = "SIGGRAPH";
  $location_vars = array("ville", "etat");
  $result = compact("evenement", $location_vars);
?>
```

Voir aussi `extract` .

Note

`compact` a été ajoutée en PHP 4.0.

6.2.41 count() : Compte le nombre d'éléments d'un tableau

`int` **count** (`mixed var`)
[lien librairie]

`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`).

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.

Exemple avec `count`

```
<?php
  $a[0] = 1;
  $a[1] = 3;
  $a[2] = 5;
  $result = count ($a);
  //$result == 3
?>
```

Voir aussi `sizeof` , `isset` et `is_array` .

6.2.42 current() : Transforme une variable en tableau

`mixed` **current** (`array array`)
[lien librairie]

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` .

Voir aussi `end` , `next` , `prev` et `reset` .

6.2.43 each() : Retourne chaque paire clé/valeur d'un tableau

`array` **each** (`array array`)
[lien librairie]

`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, et **1** et **value** contiennent la valeur.

Si le pointeur interne de fichier est au−delà de la fin du tableau, `each` retourne FALSE .

Exemples avec `each`

```
<?php
  $foo = array("bob", "fred", "jussi", "jouni", "egon", "marliese");
  $bar = each($foo);
?>
```

`each` est utilisé conjointement avec `list` pour étudier tous les éléments d'un tableau; par exemple, $HTTP_POST_VARS:

Affichage de $HTTP_POST_VARS avec `each`

```
<?php
  echo "Valeurs transmises par la méthode POST<BR>";
  reset($HTTP_POST_VARS);
  while(list ($key, $val) = each ($HTTP_POST_VARS)) {
    echo "$key => $val<BR>";
```

```
  }
?>
```

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 .

6.2.44 end() : Positionne le pointeur de tableau en fin de tableau

end array array ()
[lien librairie]

end déplace le pointeur interne du tableau array jusqu'au dernier élément.

Voir aussi current , each , end , next et reset .

6.2.45 extract() : Importe les variables dans la table des symboles

int **extract** (array var_array , *int extract_type* , *string prefix*)
[lien librairie]

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.

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.

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 ou EXTR_PREFIX_INVALID. 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

<A HREF="function.wddx-deserialize.php"><TT>wddx_deserialize</TT></a>
 */
  $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 va afficher
bleu, large, 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.

6.2.46 in_array() : Indique si une valeur appartient à un tableau

boolean **in_array** (mixed needle , array haystack , *boolean strict*)
[lien librairie]

in_array recherche needle dans haystack et retourne TRUE s'il s'y trouve, ou FALSE sinon.

Le troisième paramètre strict est optionnel. S'il vaut TRUE alors in_array vérifiera aussi que le types du paramètre needle correspond à la valeur trouvée dans haystack .

Exemple avec in_array

```
<?php
  $os = array("Mac", "NT", "Irix", "Linux");
  if (in_array("Irix", $os))
       print "Irix trouve";
?>
```

in_array avec le paramètre strict

```
<?php
  $a = array('1.10', 12.4, 1.13);
  if (in_array('12.4', $a,
<TT>TRUE</TT>
))
    echo "'12.4' trouvé avec une recherche stricte\n";
  if (in_array(1.13, $a,
<TT>TRUE</TT>
```

```
))
    echo "1.13 trouvé avec une recherche stricte\n";
?>
```

L'affichage sera :

```
1.13 trouvé avec une recherche stricte
```

Note
`in_array` a été ajoutée en PHP 4.0.

Voir aussi `array_search` .

6.2.47 key() : Retourne une clé d'un tableau associatif

`mixed` **key** `( array array )`
[lien librairie]

`key` retourne l'index de la clé courante dans un tableau.

Voir aussi `current` et `next`

6.2.48 krsort() : Trie un tableau en sens inverse et suivant les clés

`int` **krsort** `( array array )`
[lien librairie]

`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
  $fruits = array("d"=>"papaye","a"=>"orange","b"=>"banane","c"=>"ananas");
  ksort($fruits);
  for(reset($fruits); $key = key($fruits); next($fruits)) {
    echo "fruits[$key] = ".$fruits[$key]."\n";
  }
?>
```

Cet exemple va afficher :

```
fruits[d] = citron
fruits[c] = ananas
fruits[b] = banane
fruits[a] = orange
```

Voir aussi `array-multisort` , `arsort` , `asort` , `ksort` , `natsort` , `natcasesort` , `rsort` , `sort` , `uasort` , `uksort` et `usort` .

6.2.49 ksort() : Trie un tableau suivant les clés

`int` **ksort** (`array array`)
[lien librairie]

`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 :

```
fruits[a] = orange
fruits[b] = banane
fruits[c] = ananas
fruits[d] = citron
```

Vous pouvez modifier le comportement du tri avec les options `sort_flags` . Pour plus de détails, voyez `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.

6.2.50 list() : Transforme une liste de variables en tableau

`void` **list** (`void`)
[lien librairie]

Tout comme `array` , `list` n'est pas une véritable fonction, mais une construction syntaxique, qui permet d'assigner une série de variables en une seule ligne.

Exemple avec `list`

```
<?php
<table>
 <tr>
  <th>Nom de l'employé</th>
  <th>Salaire</th>
 </tr>
<?php
  $result = mysql_query($conn, "SELECT id, name, salary FROM employees");
  while (list($id, $name, $salary) = mysql_fetch_row ($result)) {
    print (" <tr>\n".
           "  <td><a href=\"info.php3?id=$id\">$name</a></td>\n".
```

```
          "  <td>$salaire</td>\n".
          " </tr>\n");
  }
?>
</table>
?>
```

Voir aussi each et array .

6.2.51 natcasesort() : Tri d'un tableau avec l'algorithme à "ordre naturel" insensible à la casse

void **natcasesort** (array array)
[lien librairie]

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's Natural Order String Comparison .

Voir aussi array−multisort , arsort , asort , krsort , ksort , natsort , rsort , sort , uasort , uksort , usort , strnatcmp et strnatcasecmp .

6.2.52 natsort() : Tri d'un tableau avec l'algorithme à "ordre naturel"

void **natsort** (array array)
[lien librairie]

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 "\nTri par Ordre Naturel\n";
  print_r($array2);
?>
```

L'exemple ci−dessous génère l'affichage suivant :

```
Tri Standard
```

```
Array
(
    [0] => img1.png
    [1] => img10.png
    [2] => img12.png
    [3] => img2.png
)
Tri par Ordre Naturel
Array
(
    [3] => img1.png
    [2] => img2.png
    [1] => img10.png
    [0] => img12.png
)
?>
```

Pour plus de détails, rendez−vous sur le site de Martin Pool Natural Order String Comparison .

Voir aussi `array−multisort` , `arsort` , `asort` , `krsort` , `ksort` , `natsort` , `natcasesort` , `rsort` , `sort` , `uasort` , `uksort` , `usort` , `strnatcmp` et `strnatcasecmp` .

6.2.53 next() : Avance le pointeur interne d'un tableau

`mixed` **next** `( array array )`
[lien librairie]

`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` .

Voir aussi `current` , `end` , `prev` et `reset` .

6.2.54 pos() : Retourne l'élément courant d'un tableau

`mixed` **pos** `( array array )`
[lien librairie]

`pos` est une fonction alias de `current` .

Voir aussi `end` , `next` , `prev` et `reset` .

6.2.55 prev() : Recule le pointeur courant de tableau

`mixed` **prev** `( array array )`
[lien librairie]

`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.

Voir aussi `current` , `end` `next` et `reset` .

6.2.56 range() : Crée un tableau contenant un intervalle d'éléments

`array` **range** `( int low , int high )`
[lien librairie]

`range` retourne un tableau contenant tous les entiers depuis `low` jusqu'à `high` , inclus.

Voir aussi `shuffle` (pour un exemple d'utilisation).

6.2.57 reset() : Remet le pointeur interne de tableau au début

`mixed` **reset** `( array array )`
[lien librairie]

`reset` replace le pointeur de tableau `array` au premier élément.

`reset` retourne la valeur du premier élément.

Voir aussi `current` , `each` , `next` , `prev` et `reset` .

6.2.58 rsort() : Trie en ordre inverse

`void` **rsort** `( array array )`
[lien librairie]

`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.

Voir aussi `array_multisort` , `arsort` , `asort` , `krsort` , `ksort` , `natsort` , `natcasesort` , `sort` , `uasort` , `uksort` et `usort` .

6.2.59 shuffle() : Mélange les éléments d'un tableau

`void` **shuffle** `( array array )`
[lien librairie]

`shuffle` mélange les éléments d'un tableau.

Exemple avec `shuffle`

```
<?php
  $numbers = range (1,20);
  srand (time());
  shuffle ($numbers);
  while (list(, $number) = each ($numbers)) {
    echo "$number ";
}
?>
```

Voir aussi `array_multisort` , `arsort` , `asort` , `krsort` , `ksort` , `natsort` , `natcasesort` , `rsort` , `sort` , `uasort` , `uksort` et `usort` .

6.2.60 sizeof() : Retourne le nombre d'élément d'un tableau

`int` **sizeof** `( array array )`
[lien librairie]

`sizeof` retourne le nombre d'élément d'un tableau.

Voir aussi `count` .

6.2.61 sort() : Trie le tableau

`void` **sort** `( array array )`
[lien librairie]

`sort` trie le tableau `array` . Les éléments seront triés du plus petit au plus grand.

Exemple avec `sort`

```php
<?php
  $fruits = array("papaye","orange","banane","ananas");
  sort($fruits);
  for(reset($fruits); $key = key($fruits); next($fruits)) {
    echo "fruits[$key] = ".$fruits[$key]."\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.

Voir aussi `array_multisort` , `arsort` , `asort` , `krsort` , `ksort` , `natsort` , `natcasesort` , `rsort` , `uasort` , `uksort` et `usort` .

6.2.62 uasort() : Trie d'un tableau en utilisant une fonction de comparaison définie par l'utilisateur.

`void` **uasort** (`array array` , `function cmp_function`)
[lien librairie]

`uasort` trie un tableau 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 est définie par l'utilisateur.

6.2.63 uksort() : Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur

`void` **uksort** (`array array` , `function cmp_function`)
[lien librairie]

`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
<?php
  function mycompare($a, $b) {
    if ($a == $b) return 0;
    return ($a > $b) ? -1 : 1;
  }
  $a = array(4 => "quatre", 3 => "trois", 20 => "vingt", 10 => "dix");
  uksort($a, mycompare);
  while(list($key, $value) = each($a)) {
    echo "$key: $value\n";
  }
?>
```

Cet exemple affichera:

```
20: vingt
10: dix
4: quatre
3: trois
```

Voir aussi `array-multisort` , `arsort` , `asort` , `krsort` , `ksort` , `natsort` , `natcasesort` , `rsort` , `sort` , `uasort` et `usort` .

6.2.64 usort() : Trie un tableau en utilisant une fonction de comparaison définie par l'utilisateur

`void` **usort** (`array array` , `function cmp_function`)
[lien librairie]

`usort` va trier un tableau 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.

Exemple avec `usort`

```php
<?php
  function cmp($a,$b) {
    if ($a == $b) return 0;
    return ($a < $b) ? -1 : 1;
  }
  $tableau = array(3,2,5,6,1);
  usort($a, "cmp");
  while(list($cle,$valeur) = each($tableau)) {
    echo "$cle: $valeur\n";
  }
?>
```

Cet exemple va afficher :

```
0: 6
1: 5
2: 3
3: 2
4: 1
```

Note

Evidemment dans ce cas trivial, `rsort` serait plus approprié.

Attention

Les bibliothèques de tri rapides sur lesquelles reposent PHP peuvent le conduire à un plantage, si la fonction de comparaison ne retourne pas une valeur cohérente.

Voir aussi `array-multisort` , `arsort` , `asort` , `krsort` , `ksort` , `natsort` , `natcasesort` , `rsort` , `sort` , `uasort` et `uksort` .

6.3 Aspell

Les fonctions Aspell vous permettent de vérifier l'orthographe d'un mot, et d'offrir des suggestions de corrections. Plusieurs langues sont disponibles, comme le français, l'anglais, l'allemand, le suédois ou le danois.

Note
Aspell fonctionne avec de très vielles versions (jusqu'à la version .27.* ou presque) de la librairie Aspell. Ce module, et ces versions d'Aspell ne sont plus supportées. Si vous voulez exploiter des capacités de vérifications orthographiques en PHP, utilisez plutôt pspell . Ce module utilise la librairie pspell qui fonctionne avec les nouvelles versions de Aspell.

Vous avez besoin de la librairie Aspell, disponible à : http://aspell.sourceforge.net/ .

- aspell_check_raw
- aspell_check
- aspell_new
- aspell_suggest

6.3.2 aspell_check() : Vérifie un mot [Obsolète]

`boolean` **aspell_check** (`resource dictionary_link` , `string word`)
[lien librairie]

`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.";
  }
?>
```

6.3.3 aspell_new() : Charge un nouveau dictionnaire [Obsolète]

`resource` **aspell_new** (`string master` , `string personal`)
[lien librairie]

`aspell_new` ouvre un nouveau dictionaire, 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");
?>
```

6.3.4 aspell_suggest() : Suggère l'orthographe d'un mot [Obsolète]

`array` **aspell_suggest** `( resource dictionary_link , string word )`
[lien librairie]

`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>";
    }
  }
?>
```

6.4 Nombres de grande taille

6.4.1 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.

6.4.2 Pré−requis

Depuis PHP 4.0.4, libbcmath est distribuée avec PHP . Vous n'avez besoin d'aucune librairie externe pour cette extension.

6.4.3 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 a été n' pas été activée lors de la compilation.

6.4.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
bcmath.scale	0	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

`bcmath.scale` *`integer`*
: Le nombre de décimales pour toutes les fonctions bcmath.

6.4.5 Types de ressources

Cette extension ne définit aucune ressource.

6.4.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources

- Constantes Prédefinies
- bcadd
- bccomp
- bcdiv
- bcmod
- bcmul
- bcpow
- bcscale
- bcsqrt
- bcsub

6.4.8 bccomp() : Compare deux nombres de grande taille.

`int` **bccomp** (`string left_operand` , `string right_operand` , *`int scale`*)
[lien librairie]

`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 égales. Si l'opérande `left_operand` est plus grande que l'opérande `right_operand` , le résultat est 1. Si l'opérande `left_operand` est plus petite que l'opérande `right_operand` , le résultat est −1.

6.4.9 bcdiv() : Divise deux nombres de grande taille.

`string` **bcdiv** (`string left_operand` , `string right_operand` , *`int scale`*)
[lien librairie]

`bcdiv` divise l'opérande `left_operand` par l'opérande `right_operand` et renvoie le résultat. Le paramètre optionnel `scale` définit le nombre de chiffres après la virgule dans le résultat.

Voir aussi `bcmul` .

6.4.10 bcmod() : Retourne le reste d'une division entre nombre de grande taille.

`string` **bcmod** (`string left_operand` , `string modulus`)
[lien librairie]

`bcmod` retourne le reste de la division entre `left_operand` en utilisant `modulus` .

Voir aussi `bcdiv` .

6.4.11 bcmul() : Multiplie deux nombres de grande taille.

`string` **bcmul** (`string left_operand` , `string right_operand` , *`int scale`*)
[lien librairie]

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 .

6.4.12 bcpow() : Elève un nombre à la puissance n−ième.

string **bcpow** (string x , string y , *int scale*)
[lien librairie]

bcpow élève x à la puissance y . Le paramètre optionnel scale définit le nombre de chiffres après la virgule dans le résultat.

Voir aussi bcsqrt .

6.4.13 bcscale() : Détermine le nombre de décimales par défaut

string **bcscale** (int scale)
[lien librairie]

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 .

6.4.14 bcsqrt() : Renvoie la racine carrée d'un nombre de grande taille.

string **bcsqrt** (string operand , *int scale*)
[lien librairie]

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 .

6.4.15 bcsub() : Soustrait un nombre de grande taille à un autre.

string **bcsub** (string left_operand , string right_operand , *int scale*)
[lien librairie]

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.

Voir aussi bcadd .

6.5 Compression Bzip2

Ce module utilise les fonctions de la librairie bzip2 , de Julian Seward pour écrire et lire des fichier bzip2 (.bz2) de manière transparente.

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.

6.5.1 Exemple de compression bzip2

Cet exemple ouvre un fichier temporaire, et écrit une ligne de test, puis il en affiche le contenu.

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);
?>
```

- Exemple de compression bzip2
- bzclose
- bzcompress
- bzdecompress
- bzerrno
- bzerror
- bzerrstr
- bzflush
- bzopen
- bzread
- bzwrite

6.5.3 bzcompress() : Compresse une chaîne avec bzip2

`string` **bzcompress** (`string source` , *`int blocksize`* , *`int workfactor`*)
[lien librairie]

`bzcompress` compresse la chaîne `source` et retourne les données ainsi encodée.

Le paramètre optionnel `blocksize` spécifie la taille de bloc utilisé 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 ressource pour ce 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` .

6.5.4 bzdecompress() : Décompresse une chaîne bzip2

`string` **bzdecompress** (`string source` , *`int small`*)
[lien librairie]

`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` .

6.5.5 bzerrno() : Retourne le numéro d'erreur bzip2

`int` **bzerrno** (`resource bz`)
[lien librairie]

`bzerrno` retourne le numéro d'erreur du fichier bz2 représenté par le pointeur `bz` .

Voir aussi `bzerror` et `bzerrstr` .

6.5.6 bzerror() : Retourne le numéro et le message d'erreur bzip2 dans un tableau

`array` **bzerror** (`int bz`)
[lien librairie]

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 .

6.5.7 bzerrstr() : Retourne le message d'erreur bzip2

string **bzerrstr** (resource bz)
[lien librairie]

bzerrstr retourne le message d'erreur du fichier bz2 représenté par le pointeur bz .

Voir aussi bzerrno et bzerror .

6.5.8 bzflush() : Force l'écriture de toutes les données compressées

int **bzflush** (resource bz)
[lien librairie]

bzflush vide les buffers d'écriture du fichier représenté par bz .

bzflush retourne TRUE en cas de succès, et FALSE sinon.

Voir aussi bzread et bzwrite .

6.5.9 bzopen() : Ouvre un fichier compressé avec bzip2

resource **bzopen** (string filename , string mode)
[lien librairie]

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` .

6.5.10 bzread() : Lecture binaire d'un fichier bzip2

`string` **bzread** (`resource bz` , *`int length`*)
[lien librairie]

`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` .

6.5.11 bzwrite() : Ecriture binaire dans un fichier bzip2

`int` **bzwrite** (`resource bz` , `string data` , *`int length`*)
[lien librairie]

`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` .

6.6 Calendrier

6.6.1 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 (en anglais), reportez−vous à http://genealogy.org/~scottlee/cal−overview.html . Les traductions issues de ces pages seront mises entre guillemets.

6.6.2 Installation

Pour faire fonctionner ces fonctions avec PHP , vous devez le compiler avec l'option `−−enable−calendar` .

6.6.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.6.4 Types de ressources

Cette extension ne définit aucune ressource.

6.6.5 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` (nombre ˆ virgule flottante)
`CAL_JULIAN` (nombre ˆ virgule flottante)
`CAL_JEWISH` (nombre ˆ virgule flottante)
`CAL_FRENCH` (nombre ˆ virgule flottante)
`CAL_NUM_CALS` (nombre ˆ virgule flottante)
`CAL_DOW_DAYNO` (nombre ˆ virgule flottante)
`CAL_DOW_SHORT` (nombre ˆ virgule flottante)
`CAL_DOW_LONG` (nombre ˆ virgule flottante)
`CAL_MONTH_GREGORIAN_SHORT` (nombre ˆ virgule flottante)
`CAL_MONTH_GREGORIAN_LONG` (nombre ˆ virgule flottante)
`CAL_MONTH_JULIAN_SHORT` (nombre ˆ virgule flottante)
`CAL_MONTH_JULIAN_LONG` (nombre ˆ virgule flottante)
`CAL_MONTH_JEWISH` (nombre ˆ virgule flottante)
`CAL_MONTH_FRENCH` (nombre ˆ virgule flottante)

The following constants are available since PHP 4.3.0 :

CAL_EASTER_DEFAULT (nombre ˆ virgule flottante)
CAL_EASTER_ROMAN (nombre ˆ virgule flottante)
CAL_EASTER_ALWAYS_GREGORIAN (nombre ˆ virgule flottante)
CAL_EASTER_ALWAYS_JULIAN (nombre ˆ virgule flottante)

- Introduction
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- 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

6.6.7 cal_from_jd() : Convertit le nombre de jours Julien en un calendrier et retourne des détails

`array` **cal_from_jd** (`int jd` , `int calendar`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.6.8 cal_info() : Retourne des détails sur un calendrier

`array` **cal_info** (`int calendar`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.6.9 cal_to_jd() : Convertit un calendrier en nombre de jours Julien

`int` **cal_to_jd** (`int calendar` , `int month` , `int day` , `int year`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.6.10 easter_date() : Retourne un timestamp UNIX pour Pâques, à minuit

`int` **easter_date** (`int year`)
[lien librairie]

`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.

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`

```
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.

6.6.11 easter_days() : Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée.

`int` **easter_days** (`int year`)
[lien librairie]

`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.

`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` .

6.6.12 FrenchToJD() : Convertit une date du calendrier français républicain en nombre de jours du calendrier Julien.

`int` **frenchtojd** (`int month` , `int day` , `int year`)
[lien librairie]

`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 September 1792 à 22 September 1806 en calendrier grégorien). Cela couvre plus que la durée d'existence de ce calendrier.

6.6.13 GregorianToJD() : Convertit une date grégorienne en nombre de jours du calendrier Julien.

`int` **gregoriantojd** (`int month` , `int day` , `int year`)
[lien librairie]

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");
?>
```

6.6.14 JDDayOfWeek() : Retourne le numéro du jour de la semaine.

`mixed` **jddayofweek** (`int julianday` , `int mode`)
[lien librairie]

`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).

6.6.15 JDMonthName() : Retourne le nom du mois.

`string` **jdmonthname** (`int julianday` , `int mode`)
[lien librairie]

`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

6.6.16 JDToFrench() : Convertit le nombre de jours du calendrier Julien en date du calendrier français républicain

`string` **jdtofrench** (`int juliandaycount`)
[lien librairie]

`jdtofrench` convertit le nombre de jours du calendrier julien en date du calendrier français républicain.

6.6.17 JDToGregorian() : Convertit le nombre de jours du calendrier Julien en date grégorienne.

`string` **jdtogregorian** (`int julianday`)
[lien librairie]

`jdtogregorian` convertit le nombre de jours du calendrier Julien en une chaîne contenant une date du calendrier grégorien, au format "mois/jour/année".

6.6.18 JDToJewish() : Convertit le nombre de jours du calendrier Julien en date du calendrier juif.

`string` **jdtojewish** (`int julianday`)
[lien librairie]

`jdtojewish` convertit le nombre de jours du calendrier Julien en date du calendrier juif.

6.6.19 JDToJulian() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien.

`string` **jdtojulian** (`int julianday`)
[lien librairie]

`jdtojulian` convertit le nombre de jours du calendrier Julien en une chaîne contenant la date du calendrier Julien, au format "mois/jour/année".

6.6.20 jdtounix() : Convertit un nombre de jours Julien en timestamp UNIX

`int` **jdtounix** (`int jday`)
[lien librairie]

`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 `jdtounix` .

Note
`jdtounix` n'est disponible qu'à partir de la version PHP 4.0RC1.

6.6.21 JewishToJD() : Convertit une date du calendrier juif en nombre de jours du calendrier Julien.

`int` **jewishtojd** (`int month` , `int day` , `int year`)
[lien librairie]

Bien qu'il soit possible de manipuler des dates à partir de l'an 1 (3761 avant JC), 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 commencait quand une nouvelle lune était observée.

6.6.22 JulianToJD() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien.

`int` **juliantojd** (`int month` , `int day` , `int year`)
[lien librairie]

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 JC, et ses détails ne furent finalisés qu'au plus tôt en 8 après JC, et probablement pas avant le 4ème 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.

6.6.23 unixtojd() : Convertit un timestamp UNIX en nombre de jours Julien

`int` **unixtojd** (*`int timestamp`*)
[lien librairie]

`unixtojd` retourne le nombre de jours Juliens du timestamp UNIX `timestamp` (nombre de secondes depuis le 1/1/1970), ou pour le jour courant si `timestamp` est omis.

Voir aussi `jdtounix` .

`unixtojd` n'est disponible qu'à partir de la version PHP 4.0RC1.

Voir aussi `easter_date` .

6.7 Paiement CCVS

Ces fonctions font l'interface avec les API CCVS, vous permettant de travailler directement avec CCVS depuis vos scripts PHP. CCVS est la solution apportée par RedHat au problème de l'intermédiaire, lors du traitement de transactions de cartes de crédit. Il vous permet travailler directement avec les maisons de crédits, via votre boîte *nix et un modem. En utilisant le module CCVS pour PHP, vous pouvez effectuer des transactions avec les cartes de crédits, directement depuis vos scripts PHP via CCVS. La suite vous montrera comment procéder.

Pour activer le support CCVS de PHP, commencez par vérifier votre installation CCVS. Vous devez configurer PHP avec l'option `--with-ccvs` . Si vous utilisez cette option sans spécifier le chemin de votre installation, PHP essaiera de la trouver à sa position par défaut (/usr/local/ccvs). Si CCVS est installé dans un autre dossier, lancez la configuration avec : `--with-ccvs=$ccvs_path` , où $ccvs_path est le chemin de votre installation CCVS. Notez bien que CCVS requiert que `$ccvs_path/lib` et `$ccvs_path/include` existent, et qu'ils contiennent respectivement `cv_api.h` et `libccvs.a` 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 CCVS (i.e. si vous avez installé CCVS avec l'utilisateur 'ccvs', vos scripts PHP doivent tourner aussi en 'ccvs').

Plus de détails sur CCVS sont disponibles à http://www.redhat.com/products/ccvs .

Cette documentation est en chantier. Jusqu'à sa finalisation, RedHat entretient une version légèrement démodée mais bien pratique à http://www.redhat.com/products/ccvs/support/CCVS3.3docs/ProgPHP.html .

- 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

6.7.2 ccvs_auth() : Perform credit authorization test on a transaction

`string` **ccvs_auth** (`string session` , `string invoice`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.3 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.4 ccvs_count() : Find out how many transactions of a given type are stored in the system

`int` **ccvs_count** (`string session` , `string type`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.5 ccvs_delete() : Delete a transaction

`string` **ccvs_delete** (`string session` , `string invoice`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.6 ccvs_done() : Terminate CCVS engine and do cleanup work

`string` **ccvs_done** (`string sess`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.7 ccvs_init() : Initialize CCVS for use

`string` **ccvs_init** (`string name`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.8 ccvs_lookup() : Look up an item of a particular type in the database

`string` **ccvs_lookup** `( string session , string invoice , int inum )`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.9 ccvs_new() : Create a new, blank transaction

`string` **ccvs_new** `( string session , string invoice )`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.10 ccvs_report() : Return the status of the background communication process

`string` **ccvs_report** `( string session , string type )`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.11 ccvs_return() : Transfer funds from the merchant to the credit card holder

`string` **ccvs_return** `( string session , string invoice )`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.12 ccvs_reverse() : Perform a full reversal on an already−processed authorization

`string` **ccvs_reverse** `( string session , string invoice )`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.13 ccvs_sale() : Transfer funds from the credit card holder to the merchant

`string` **ccvs_sale** (`string session` , `string invoice`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.14 ccvs_status() : Check the status of an invoice

`string` **ccvs_status** (`string session` , `string invoice`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.15 ccvs_textvalue() : Get text return value for previous function call

`string` **ccvs_textvalue** (`string session`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.7.16 ccvs_void() : Perform a full reversal on a completed transaction

`string` **ccvs_void** (`string session` , `string invoice`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.8 Support COM pour Windows

6.8.1 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.

6.8.2 Pré−requis

Les fonctions COM sont uniquement disponibles sur les versions Windows de PHP. Elles ont été ajoutées à partir de PHP 4.0.

6.8.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.8.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
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

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.8.5 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` (nombre ˆ virgule flottante)
`CLSCTX_INPROC_HANDLER` (nombre ˆ virgule flottante)
`CLSCTX_LOCAL_SERVER` (nombre ˆ virgule flottante)
`CLSCTX_REMOTE_SERVER` (nombre ˆ virgule flottante)
`CLSCTX_SERVER` (nombre ˆ virgule flottante)
`CLSCTX_ALL` (nombre ˆ virgule flottante)

VT_NULL (nombre ^ virgule flottante)
VT_EMPTY (nombre ^ virgule flottante)
VT_UI1 (nombre ^ virgule flottante)
VT_I2 (nombre ^ virgule flottante)
VT_I4 (nombre ^ virgule flottante)
VT_R4 (nombre ^ virgule flottante)
VT_R8 (nombre ^ virgule flottante)
VT_BOOL (nombre ^ virgule flottante)
VT_ERROR (nombre ^ virgule flottante)
VT_CY (nombre ^ virgule flottante)
VT_DATE (nombre ^ virgule flottante)
VT_BSTR (nombre ^ virgule flottante)
VT_DECIMAL (nombre ^ virgule flottante)
VT_UNKNOWN (nombre ^ virgule flottante)
VT_DISPATCH (nombre ^ virgule flottante)
VT_VARIANT (nombre ^ virgule flottante)
VT_I1 (nombre ^ virgule flottante)
VT_UI2 (nombre ^ virgule flottante)
VT_UI4 (nombre ^ virgule flottante)
VT_INT (nombre ^ virgule flottante)
VT_UINT (nombre ^ virgule flottante)
VT_ARRAY (nombre ^ virgule flottante)
VT_BYREF (nombre ^ virgule flottante)
CP_ACP (nombre ^ virgule flottante)
CP_MACCP (nombre ^ virgule flottante)
CP_OEMCP (nombre ^ virgule flottante)
CP_UTF7 (nombre ^ virgule flottante)
CP_UTF8 (nombre ^ virgule flottante)
CP_SYMBOL (nombre ^ virgule flottante)
CP_THREAD_ACP (nombre ^ virgule flottante)

6.8.6 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 Prédefinies
- Voir aussi
- COM
- VARIANT
- com_addref
- com_get
- com_invoke
- com_isenum
- com_load_typelib
- com_load
- com_propget
- com_propput
- com_propset

- com_release
- com_set

6.8.8 VARIANT() : classe VARIANT

```
$vVar = new VARIANT ($var)
```

6.8.9 com_addref() : Incrémente le compteur de références

`void` **com_addref** (`void`)
[lien librairie]

`com_addref` incrémente le compteur de références.

6.8.10 com_get() : Lit la valeur d'un propriété d'un composant COM

`mixed` **com_get** (`resource com_object` , `string property`)
[lien librairie]

`com_get` retourne la valeur de la propriété `property` du composant COM `com_object` .
`com_get` retourne FALSE en cas d'erreur.

6.8.11 com_invoke() : Appelle une méthode d'un composant

`mixed` **com_invoke** (`resource com_object` , `string function_name` , *`mixed function parameters, ...`*)
[lien librairie]

`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.

6.8.12 com_isenum() : Récupère un IEnumVariant

`void` **com_isenum** (`object com_module`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.8.13 com_load_typelib() : Charge une librairie de type (Typelib)

`void` **com_load_typelib** (`string typelib_name` , *`int case_insensitive`*)
[lien librairie]

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.8.14 com_load() : Crée une référence sur un composant COM

`resource` **com_load** (`string module_name` , *`string server_name`*)
[lien librairie]

`com_load` crée une nouvelle référence au composant COM `module_name` , et retourne une référence dessus. `com_load` retourne FALSE en cas d'échec.

6.8.15 com_propget() : Lit la valeur d'un propriété d'un composant COM

`mixed` **com_propget** (`resource com_object` , `string property`)
[lien librairie]

`com_propget` est un alias de `com_get` .

6.8.16 com_propput() : Modifie une propriété d'un composant COM

`void` **com_propput** (`resource com_object` , `string property` , `mixed value`)
[lien librairie]

`com_propput` est un alias de `com_set` .

6.8.17 com_propset() : Modifie une propriété d'un composant COM

`void` **com_propset** (`resource com_object` , `string property` , `mixed value`)
[lien librairie]

`com_propset` est un alias de `com_propput` .

6.8.18 com_release() : Décrémente le compteur de références

`void` **com_release** (`void`)
[lien librairie]

`com_release` décrémente le compteur de références.

6.8.19 com_set() : Modifie une propriété d'un composant COM

`void` **com_set** (`resource com_object` , `string property` , `mixed value`)
[lien librairie]

`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.

6.9 Classes et Objets

6.9.1 Introduction

6.9.1.1 About

Ces fonctions permettent de gérer les classes et les objets. Vous pouvez notamment connaître le nom de la classe d'un objet, ses membres et ses méthodes, et tous les objets parents (les classes qui sont étendues par la classe d'un objet).

6.9.1.2 Exemple d'utilisation

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.

classes.inc

```
<?php
// classe de base, avec ses membres et ses méthodes
class Legume {
    var $mangeable;
    var $couleur;
    function legume( $mangeable, $couleur="green" ) {
        $this->mangeable = $mangeable;
        $this->couleur = $couleur;
    }
    function est_mangeable() {
        return $this->mangeable;
    }
    function quelle_couleur() {
        return $this->couleur;
    }
} // fin de la classe Legume
// extend la classe de base
class Epinard extends Legume {
    var $cuit =
<TT>FALSE</TT>
;
    function Epinard() {
        $this->Legume(
<TT>TRUE</TT>
, "green" );
    }
    function cuisine() {
        $this->cuit =
<TT>TRUE</TT>
;
    }
    function est_cuit() {
        return $this->cuit;
    }
} // fin de la classe Epinard
?>
```

Lorsqu'on instantie deux objets de ces classes, et qu'on affiche leurs informations, on affiche aussi leur héritage. On définit ici des utilitaires qui servent essentiellement à afficher ces informations proprement.

test_script.php

```
<pre>
<?php
  include "classes.inc";
// 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 "\tfunction $method()\n";
  }
  function class_parentage($obj, $class) {
    global $$obj;
    if (is_subclass_of($$obj, $class)) {
        echo "L'objet $obj belongs to class ".get_class($$obj);
        echo " est une sous-classe de $class\n";
    } else {
        echo "L'objet $obj n'est pas une sous classe $class\n";
    }
  }
// instantie 2 objets
  $legume = new Legume(TRUE,"blue");
  $feuilles = new Epinard();
// affiche les informations sur ces objets
  echo "legume: CLASS ".get_class($legume)."\n";
  echo "feuilles: CLASS ".get_class($feuilles);
  echo ", PARENT ".get_parent_class($feuilles)."\n";
// affiche les propriétés du légume
  echo "\nlégume: Propriétés \n";
  print_vars($legume);
// et les méthodes de "feuilles"
  echo "\nfeuilles: Methods\n";
  print_methods($feuilles);
  echo "\nParentée:\n";
  class_parentage("feuilles", "Epinard");
  class_parentage("feuilles", "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 Spinach
L'objet feuilles est une sous-classe de Legume
```

- Introduction
- 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

6.9.3 call_user_method() : Appelle une méthode utilisateur d'un objet

`mixed` **call_user_method** (`string method_name` , `object obj` , *`mixed parameter`* , *`mixed ...`*)
[lien librairie]

Appelle la méthode `method_name` depuis 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` .

```
<?php
class Pays {
    var $NOM;
    var $TLD;
    function Pays($nom, $tld) {
        $this->NOM = $nom;
        $this->TLD = $tld;
    }
    function print_info($prestr="") {
        echo $prestr."Pays: ".$this->NOM."\n";
        echo $prestr."Nom de domaine: ".$this->TLD."\n";
    }
}
$unPays = new Pays("Pérou","pe");
echo "* Appel de la méthode directement\n";
$unPays->print_info();
echo "\n* Appel de la méthode indirectement\n";
call_user_method ("print_info", $unPays, "\t");
?>
```

Voir aussi `call_user_func_array` . `call_user_func` et `call_user_method_array` .

6.9.4 class_exists() : Vérifie qu'une classe a été définie

`boolean` **class_exists** (`string class_name`)
[lien librairie]

`class_exists` retourne TRUE si la classe `class_name` a été définie, et FALSE sinon.

6.9.5 get_class_methods() : Retourne les noms des méthodes d'une classe.

`array` **get_class_methods** (`string class_name`)
[lien librairie]

`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 myclass {
    // 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_class));
foreach ($class_methods as $method_name) {
    echo "$method_name\n";
}
?>
```

Va afficher :

```
maclass
mafunc1
mafunc2
```

Voir aussi `get_class_vars` et `get_object_vars`

6.9.6 get_class_vars() : Retourne les valeurs par défaut des attributs d'une classe.

`array` **get_class_vars** (`string class_name`)
[lien librairie]

`get_class_vars` retourne un tableau contenant les valeurs par défaut des attributs de la classe `class_name` .

Note
Les variables de classe qui ne sont pas encore initialisées ne seront pas retournées par `get_class_vars` .

Exemple `get_class_vars`

```
<?php
class maclasse {
    var $var1; // Pas de valeur par défaut
    var $var2 = "xyz";
    var $var3 = 100;
    // constructeur
    function maclasse() {
        return(TRUE);
    }
}
$ma_classe = new maclasse();
$class_vars = get_class_vars(get_class($ma_classe));
foreach ($class_vars as $name => $value) {
    echo "$name : $value\n";
}
?>
```

va afficher :

```
var2 : xyz
var3 : 100
```

6.9.7 get_class() : Retourne la classe d'un objet

`string` **get_class** (`object obj`)
[lien librairie]

`get_class` retourne la classe de l'objet `obj` .

Voir aussi `get_parent_class` et `is_subclass_of`

6.9.8 get_declared_classes() : Liste toutes les classes définies

`array` **get_declared_classes** (`void`)
[lien librairie]

`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`).

6.9.9 get_object_vars() : Retourne un tableau associatif des propriétés d'un objet

`array` **get_object_vars** (`object obj`)
[lien librairie]

`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.Si des variables déclarées dans la classe de l'objet `obj` , n'ont pas été assignées, elles ne seront pas retournées dans le tableau.

Exemple avec `get_object_vars`

```
<?php
class Point2D {
    var $x, $y;
    var $nom;
    function Point2D($x, $y) {
        $this->x = $x;
        $this->y = $y;
    }
    function donne_nom($nom) {
        $this->nom = $nom;
    }
    function LitPoint() {
        return array("x" -> $this->x,
                     "y" -> $this->y,
                     "nom" -> $this->nom);
    }
}
$p1 = new Point2D(1.233, 3.445);
print_r(get_object_vars($p1));
// "$nom" est déclaré, mais non défini
// Array
// (
//     [x] -> 1.233
//     [y] -> 3.445
// )
$p1->setnom("point #1");
print_r(get_object_vars($p1));
// Array
// (
//     [x] -> 1.233
//     [y] -> 3.445
//     [nom] -> point #1
// )
?>
```

Voir aussi `get_class_methods` et `get_class_vars`

6.9.10 get_parent_class() : Retourne le nom de la classe d'un objet

`string` **get_parent_class** (`object obj`)
[lien librairie]

get_parent_class retourne le nom de la classe de l'objet obj .

Voir aussi get_class et is_subclass_of

6.9.11 is_a() : Retourne TRUE si un objet a pour parent une classe donnée

bool **is_a** (object object , string class_name)
[lien librairie]

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 .

6.9.12 is_subclass_of() : Détermine si un objet est une sous−classe

boolean **is_subclass_of** (object obj , string superclass)
[lien librairie]

is_subclass_of retourne TRUE si l'objet obj est une sous−classe de superclass , FALSE sinon.

Voir aussi get_class et get_parent_class

6.9.13 method_exists() : Vérifie que la méthode existe pour une classe.

boolean **method_exists** (object object , string method_name)
[lien librairie]

method_exists retourne TRUE si la méthode method_name a été définie pour la classe object , et sinon, retourne FALSE .

6.10 ClibPDF

6.10.1 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.

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 convertit 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);
  }

  $ltime = getdate();

  /* draw hour hand */
  cpdf_save($pdf);
  cpdf_rotate($pdf, -(($ltime['minutes']/60.0) + $ltime['hours'] - 3.0) * 30.0);
  cpdf_moveto($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, -(($ltime['seconds']/60.0) + $ltime['minutes'] - 15.0) * 6.0);
  cpdf_moveto($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, -(($ltime['seconds'] - 15.0) * 6.0));
  cpdf_moveto($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);
?>
```

6.10.2 Voir aussi

Voir aussi la documentation de PDFlib .

- Introduction
- 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_restore
- cpdf_rlineto
- cpdf_rmoveto
- cpdf_rotate
- cpdf_save_to_file
- cpdf_save
- cpdf_scale
- cpdf_set_char_spacing
- cpdf_set_creator
- cpdf_set_current_page
- 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_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

6.10.4 cpdf_add_outline() : Ajoute un signet à la page courante.

`void` **cpdf_add_outline** (`resource pdf_document` , `string text`)
[lien librairie]

`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);
?>
```

6.10.5 cpdf_arc() : Dessine un arc de cercle.

`void` **cpdf_arc** (`resource pdf_document` , `float x-koor` , `float y-koor` , `float radius` , `float start` , `float end` , `int mode`)
[lien librairie]

`cpdf_arc` dessine un arc de cercle, dont le centre est au point (`x-koor` , `y-koor`) 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` .

6.10.6 cpdf_begin_text() : Démarre une section de texte.

`void` **cpdf_begin_text** (`resource pdf_document`)
[lien librairie]

`cpdf_begin_text` démarre une section de texte. Elle doit être terminée avec `cpdf_end_text` .

Affichage de texte

```
<?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` .

6.10.7 cpdf_circle() : Dessine un cercle.

`void` **cpdf_circle** (`resource pdf_document` , `float x-koor` , `float y-koor` , `float radius` , `int mode`)
[lien librairie]

`cpdf_circle` dessine un cercle de centre (`x-koor` , `y-koor`) 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 .

6.10.8 cpdf_clip() : Aligne les dessins sur le chemin courant.

void **cpdf_clip** (resource pdf_document)
[lien librairie]

cpdf_clip aligne les dessins sur le chemin courant.

6.10.9 cpdf_close() : Ferme un fichier PDF.

void **cpdf_close** (resource pdf_document)
[lien librairie]

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 .

6.10.10 cpdf_closepath_fill_stroke() : Remplit le chemin, dessine le bord et ferme le chemin.

void **cpdf_closepath_fill_stroke** (resource pdf_document)
[lien librairie]

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 .

6.10.11 cpdf_closepath_stroke() : Ferme le fichier et dessine une ligne le long du chemin.

void **cpdf_closepath_stroke** (resource pdf_document)
[lien librairie]

cpdf_closepath_stroke est une combinaison de cpdf_closepath et cpdf_stroke .

Voir aussi cpdf_closepath et cpdf_stroke .

6.10.12 cpdf_closepath() : Ferme le chemin.

void **cpdf_closepath** (resource pdf_document)
[lien librairie]

cpdf_closepath ferme le chemin courant.

6.10.13 cpdf_continue_text() : Imprime le texte à la ligne suivante.

void **cpdf_continue_text** (resource pdf_document , string text)
[lien librairie]

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 .

6.10.14 cpdf_curveto() : Dessine une courbe.

void **cpdf_curveto** (resource pdf_document , float x1 , float y1 , float x2 , float y2 , float x3 , float y3 , int mode)
[lien librairie]

cpdf_curveto dessine une courbe de Bezier, entre le point courant et le point (x3 , y3), en utilisant les points de contrôle (x1 , y1) et (x2 , y2).

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 .

6.10.15 cpdf_end_text() : Termine une section de texte.

void **cpdf_end_text** (resource pdf_document)
[lien librairie]

cpdf_end_text termine une section de texte, commencée avec cpdf_begin_text .

Affichage de texte

<?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_begin_text .

6.10.16 cpdf_fill_stroke() : Remplit le chemin, et dessine le bord.

void **cpdf_fill_stroke** (resource pdf_document)
[lien librairie]

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 .

6.10.17 cpdf_fill() : Remplit le chemin courant.

void **cpdf_fill** (resource pdf_document)
[lien librairie]

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 .

6.10.18 cpdf_finalize_page() : Termine une page.

void **cpdf_finalize_page** (resource pdf_document , int page_number)
[lien librairie]

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 .

6.10.19 cpdf_finalize() : Termine un document.

void **cpdf_finalize** (resource pdf_document)
[lien librairie]

cpdf_finalize termine un document. Vous devez toujours appeler cpdf_close après.

Voir aussi cpdf_close .

6.10.20 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)
[lien librairie]

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 .

6.10.21 cpdf_import_jpeg() : Ouvre une image JPEG.

`int` **cpdf_import_jpeg** (`resource pdf_document` , `string file_name` , `float x-koor` , `float y-koor` , `float angle` , `float width` , `float height` , `float x-scale` , `float y-scale` , `int mode`)
[lien librairie]

`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-koor` , `y-koor`). 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` .

6.10.22 cpdf_lineto() : Dessine une ligne.

`void` **cpdf_lineto** (`resource pdf_document` , `float x-koor` , `float y-koor` , `int mode`)
[lien librairie]

`cpdf_lineto` dessine une ligne entre le point courant et le point de coordonnées (`x-koor` , `y-koor`).

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` .

6.10.23 cpdf_moveto() : Fixe le point courant.

`void` **cpdf_moveto** (`resource pdf_document` , `float x-koor` , `float y-koor` , `int mode`)
[lien librairie]

`cpdf_moveto` fixe le point courant aux coordonnées (`x-koor` , `y-koor`).

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.

6.10.24 cpdf_newpath() : Commence un nouveau chemin

`void` **cpdf_newpath** (`int pdf_document`)
[lien librairie]

`cpdf_newpath` commence un nouveau chemin dans le document `pdf_document` .

6.10.25 cpdf_open() : Ouvre un nouveau document PDF.

`resource` **cpdf_open** (`int compression` , `string filename`)
[lien librairie]

`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'envoie 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_text` .

6.10.26 cpdf_restore() : Restaure un environnement.

`void` **cpdf_restore** (`resource pdf_document`)
[lien librairie]

`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

`<?php`
`cpdf_save($pdf);`
`// plein de transformations, translations, ...`
`cpdf_restore($pdf)`
`?>`

Voir aussi `cpdf_save` .

6.10.27 cpdf_rlineto() : Dessine une ligne, relativement.

`void` **cpdf_rlineto** (`resource pdf_document` , `float x−koor` , `float y−koor` , `int mode`)
[lien librairie]

`cpdf_rlineto` dessine une ligne entre le point courant et le point de coordonnées relatives (`x−koor` , `y−koor`).

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 .

6.10.28 cpdf_rmoveto() : Fixe le point courant relativement.

void **cpdf_rmoveto** (resource pdf_document , float x-koor , float y-koor , int mode)
[lien librairie]

cpdf_rmoveto fixe le point courant aux coordonnées (x-koor , y-koor), 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 .

6.10.29 cpdf_rotate() : Effectue une rotation.

void **cpdf_rotate** (resource pdf_document , float angle)
[lien librairie]

cpdf_rotate effectue une rotation, d'un angle de angle degrés.

6.10.30 cpdf_save_to_file() : Ecrit un document PDF dans un fichier.

void **cpdf_save_to_file** (resource pdf_document , string filename)
[lien librairie]

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 .

6.10.31 cpdf_save() : Sauve l'environnement courant.

void **cpdf_save** (resource pdf_document)
[lien librairie]

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 .

6.10.32 cpdf_scale() : Modifie l'échelle.

`void` **cpdf_scale** (`resource pdf_document` , `float x-scale` , `float y-scale`)
[lien librairie]

`cpdf_scale` modifie l'échelle dans les deux directions.

6.10.33 cpdf_set_char_spacing() : Fixe l'espacement des caractères.

`void` **cpdf_set_text_pos** (`resource pdf_document` , `float space`)
[lien librairie]

`cpdf_set_char_spacing` fixe l'espacement des caractères.

Voir aussi `cpdf_set_word_spacing` et `cpdf_set_leading` .

6.10.34 cpdf_set_creator() : Fixe le créateur d'un document PDF.

`void` **cpdf_set_creator** (`string creator`)
[lien librairie]

`cpdf_set_creator` fixe le créateur d'un document PDF.

Voir aussi `cpdf_set_subject` , `cpdf_set_title` et `cpdf_set_keywords` .

6.10.35 cpdf_set_current_page() : Fixe la page courante.

`void` **cpdf_set_current_page** (`resource pdf_document` , `int page_number`)
[lien librairie]

`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` .

6.10.36 cpdf_set_font() : Sélectionne la police courante et sa taille.

`void` **cpdf_set_font** (`resource pdf_document` , `string font name` , `float size` , `string encoding`)
[lien librairie]

`cpdf_set_font` selectionne 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.

6.10.37 cpdf_set_horiz_scaling() : Fixe l'échelle horizontale du texte.

`void` **cpdf_set_horiz_scaling** (`resource pdf_document` , `float scale`)
[lien librairie]

`cpdf_set_horiz_scaling` fixe l'échelle horizontale du texte à `scale` %.

6.10.38 cpdf_set_keywords() : Fixe les mot clés d'un document PDF.

`void` **cpdf_set_keywords** (`string keywords`)
[lien librairie]

`cpdf_set_keywords` fixe les mots−clé d'un document PDF.

Voir aussi `cpdf_set_title` , `cpdf_set_creator` et `cpdf_set_subject` .

6.10.39 cpdf_set_leading() : Fixe la distance entre deux lignes.

`void` **cpdf_set_leading** (`resource pdf_document` , `float distance`)
[lien librairie]

`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` .

6.10.40 cpdf_set_page_animation() : Fixe l'animation de la transition entre les pages.

`void` **cpdf_set_page_animation** (`resource pdf_document` , `int transition` , `float duration`)
[lien librairie]

`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.

6.10.41 cpdf_set_subject() : Fixe le sujet d'un document PDF.

`void` **cpdf_set_subject** (`string subject`)
[lien librairie]

`cpdf_set_subject` fixe le sujet d'un document PDF.

Voir aussi `cpdf_set_title` , `cpdf_set_creator` et `cpdf_set_keywords` .

6.10.42 cpdf_set_text_matrix() : Fixe la matrice du texte.

`void` **cpdf_set_text_matrix** (`resource pdf_document` , `array matrix`)
[lien librairie]

`cpdf_set_text_matrix` fixe la matrice du texte, qui décrit la transformation appliquée à police.

6.10.43 cpdf_set_text_pos() : Fixe la position du texte.

`void` **cpdf_set_text_pos** (`resource pdf_document` , `float x-koor` , `float y-koor` , `int mode`)
[lien librairie]

`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` .

6.10.44 cpdf_set_text_rendering() : Détermine le rendu du texte.

`void` **cpdf_set_text_rendering** (`resource pdf_document` , `int mode`)
[lien librairie]

`cpdf_set_text_rendering` détermines 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.

6.10.45 cpdf_set_text_rise() : Fixe l'élévation du texte.

`void` **cpdf_set_text_rise** (`resource pdf_document` , `float value`)
[lien librairie]

`cpdf_set_text_rise` fixe l'élévation du texte à `value` unités.

6.10.46 cpdf_set_title() : Fixe le titre d'un document PDF.

`void` **cpdf_set_title** (`string title`)
[lien librairie]

`cpdf_set_title` fixe le titre d'un document PDF.

Voir aussi `cpdf_set_subject` , `cpdf_set_creator` et `cpdf_set_keywords` .

6.10.47 cpdf_set_word_spacing() : Fixe l'espacement des mots.

`void` **cpdf_set_word_spacing** (`resource pdf_document` , `float space`)
[lien librairie]

`cpdf_set_word_spacing` fixe l'espacement des caractères.

Voir aussi `cpdf_set_char_spacing` et `cpdf_set_leading` .

6.10.48 cpdf_setdash() : Fixe le motif de pointillé.

`void` **cpdf_setdash** (`resource pdf_document` , `float white` , `float black`)
[lien librairie]

`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.

6.10.49 cpdf_setflat() : Fixe la platitude (flatness).

`void` **cpdf_setflat** (`resource pdf_document` , `float value`)
[lien librairie]

`cpdf_setflat` fixe la platitude (flatness), entre 0 et 100.

6.10.50 cpdf_setgray_fill() : Modifie le niveau de gris comme couleur de remplissage.

`void` **cpdf_setgray_fill** (`resource pdf_document` , `float value`)
[lien librairie]

`cpdf_setgray_fill` remplace le niveau de gris, couleur de remplissage courante, par `value` .

Voir aussi `cpdf_setrgbcolor_fill` .

6.10.51 cpdf_setgray_stroke() : Choisit un niveau de gris comme couleur de dessin.

`void` **cpdf_setgray_stroke** (`resource pdf_document` , `float gray_value`)

[lien librairie]

`cpdf_setgray_stroke` remplace le niveau de gris, couleur de dessin courante, par `value` .

Voir aussi `cpdf_setrgbcolor_stroke` .

6.10.52 cpdf_setgray() : Modifie un niveau de gris comme couleur de dessin et de remplissage.

`void` **cpdf_setgray** (`resource pdf_document` , `float gray_value`)
[lien librairie]

`cpdf_setgray_stroke` remplace le niveau de gris, couleur de dessin et de remplissage, par `value` .

Voir aussi `cpdf_setrgbcolor_stroke` et `cpdf_setrgbcolor_fill` .

6.10.53 cpdf_setlinecap() : Fixe le paramètre linecap.

`void` **cpdf_setlinecap** (`resource pdf_document` , `int value`)
[lien librairie]

`cpdf_setlinecap` fixe le paramètre linecap à une valeur `value` entre 0 et 2. 0 = butt end, 1 = round, 2 = projecting square.

6.10.54 cpdf_setlinejoin() : Fixe le paramètre linejoin.

`void` **cpdf_setlinejoin** (`resource pdf_document` , `long value`)
[lien librairie]

`cpdf_setlinejoin` fixe le paramètre linejoin à une valeur `value` , entre 0 et 2. 0 = miter, 1 = round, 2 = bevel.

6.10.55 cpdf_setlinewidth() : Fixe la largeur de ligne.

`void` **cpdf_setlinewidth** (`resource pdf_document` , `float width`)
[lien librairie]

`cpdf_setlinewidth` fixe la largeur de ligne à la valeur de `width` .

6.10.56 cpdf_setmiterlimit() : Fixe le paramètre miter limit.

`void` **cpdf_setmiterlimit** (`resource pdf_document` , `float value`)
[lien librairie]

`cpdf_setmiterlimit` fixe le paramètre "miter limit" à une valeur supérieure ou égale à 1.

6.10.57 cpdf_setrgbcolor_fill() : Choisit une couleur rgb comme couleur de remplissage.

`void` **cpdf_setrgbcolor_fill** (`resource pdf_document` , `float red_value` , `float green_value` , `float blue_value`)
[lien librairie]

`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` .

6.10.58 cpdf_setrgbcolor_stroke() : Choisit une couleur rgb comme couleur de dessin.

`void` **cpdf_setrgbcolor_stroke** (`resource pdf_document` , `float red_value` , `float green_value` , `float blue_value`)
[lien librairie]

`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` .

6.10.59 cpdf_setrgbcolor() : Choisit une couleur rgb comme couleur de dessin et de remplissage.

`void` **cpdf_setrgbcolor** (`resource pdf_document` , `float red_value` , `float green_value` , `float blue_value`)
[lien librairie]

`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` .

6.10.60 cpdf_show_xy() : Affiche un texte à une position.

`void` **cpdf_show_xy** (`resource pdf_document` , `string text` , `float x-coor` , `float y-coor` , *`int mode`*)
[lien librairie]

`cpdf_show_xy` imprime la chaîne `text` , à la position de coordonnées (`x-koor` , `y-koor`). 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 `cpdf_show_xy` est identique à `cpdf_text` sans les options.

Voir aussi cpdf_text .

6.10.61 cpdf_show() : Imprime un texte à la position courante.

void **cpdf_show** (resource pdf_document , string text)
[lien librairie]

cpdf_show imprime la chaîne text , à la position courante.

Voir aussi cpdf_text , cpdf_begin_text et cpdf_end_text .

6.10.62 cpdf_stringwidth() : Retourne la taille de la chaîne.

float **cpdf_stringwidth** (resource pdf_document , string text)
[lien librairie]

cpdf_stringwidth retourne la taille de la chaîne text . Une police doit avoir déjà été choisie.

Voir aussi cpdf_set_font .

6.10.63 cpdf_stroke() : Dessine une ligne le long du chemin.

void **cpdf_stroke** (resource pdf_document)
[lien librairie]

cpdf_stroke dessine une ligne le long du chemin.

Voir aussi cpdf_closepath et cpdf_closepath_stroke .

6.10.64 cpdf_text() : Imprime un texte avec des options.

void **cpdf_text** (resource pdf_document , string text , float x−coor , float y−coor , *int mode* , *float orientation* , *int alignmode*)
[lien librairie]

cpdf_text imprime le text text à la position de coordonnées (x−koor , y−koor). 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 optionel 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 .

6.10.65 cpdf_translate() : Modifie l'origine du système de coordonnées.

`void` **cpdf_translate** (`resource pdf_document` , `float x-koor` , `float y-koor` , `int mode`)
[lien librairie]

`cpdf_translate` modifie l'origine du système de coordonnées en plaçant l'origine aux coordonnées (`x-koor` , `y-koor`).

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.

6.11 Crack

6.11.1 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 minusculles, et effectue des recherches dans le dictionnaire CrackLib. CrackLib donne aussi des conseil pour reforcer vos mots de passe.

6.11.2 Pré−requis

Plus d'informations concernant la librairie CrackLib sont disponibles à http://www.users.dircon.co.uk/~crypto/ .

6.11.3 Installation

Pour utiliser cette fonction, vous devez compiler PHP avec l'extension Crack, en utilisant l'option de compilation `−−with−crack[=DIR]` .

6.11.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
crack.default_dictionary	NULL	PHP_INI_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.11.5 Types de ressources

Cette extension ne définit aucune ressource.

6.11.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

6.11.7 Exemples

Cet exemple montre coment ouvrir un dictionnaire CrackLib, tester un mot de passe, lire le diagnostic et refermer le dictionnaire.

CrackLib example

```
<?php
// Ouverture du dictionnaire CrackLib
```

```
$dictionary = crack_opendict('/usr/local/lib/pw_dict')
     or die('Unable to open CrackLib dictionary');

// Vérification du mot de passe
$check = crack_check($dictionary, 'gx9A2s0x');

// Lecture du diagnostic
$diag = crack_getlastmessage();
echo $diag; // 'strong password'

// Fermeture du dictionnaire
crack_closedict($dictionary);
?>
```

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

6.11.9 crack_closedict() : Referme le dictionnaire CrackLib

`bool` **crack_closedict** (*`resource dictionary`*)
[lien librairie]

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 dictionnarie courant est fermé.

6.11.10 crack_getlastmessage() : Retourne le message de diagnostic

`string` **crack_getlastmessage** (`void`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 diagnotic obtenu lors de la dernière vérification.

6.11.11 crack_opendict() : Ouvre un nouveau dictionnaire CrackLib dictionary

`resource` **crack_opendict** `( string dictionary )`
[lien librairie]

`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 dictionnarie 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` .

6.12 CURL

6.12.1 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.

6.12.2 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.

6.12.3 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` .

6.12.4 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` (nombre ˆ virgule flottante)
`CURLOPT_FILE` (nombre ˆ virgule flottante)
`CURLOPT_INFILE` (nombre ˆ virgule flottante)
`CURLOPT_INFILESIZE` (nombre ˆ virgule flottante)
`CURLOPT_URL` (nombre ˆ virgule flottante)
`CURLOPT_PROXY` (nombre ˆ virgule flottante)
`CURLOPT_VERBOSE` (nombre ˆ virgule flottante)
`CURLOPT_HEADER` (nombre ˆ virgule flottante)
`CURLOPT_HTTPHEADER` (nombre ˆ virgule flottante)
`CURLOPT_NOPROGRESS` (nombre ˆ virgule flottante)
`CURLOPT_NOBODY` (nombre ˆ virgule flottante)
`CURLOPT_FAILONERROR` (nombre ˆ virgule flottante)
`CURLOPT_UPLOAD` (nombre ˆ virgule flottante)
`CURLOPT_POST` (nombre ˆ virgule flottante)
`CURLOPT_FTPLISTONLY` (nombre ˆ virgule flottante)
`CURLOPT_FTPAPPEND` (nombre ˆ virgule flottante)
`CURLOPT_NETRC` (nombre ˆ virgule flottante)

CURLOPT_FOLLOWLOCATION (nombre ˆ virgule flottante)
CURLOPT_FTPASCII (nombre ˆ virgule flottante)
CURLOPT_PUT (nombre ˆ virgule flottante)
CURLOPT_MUTE (nombre ˆ virgule flottante)
CURLOPT_USERPWD (nombre ˆ virgule flottante)
CURLOPT_PROXYUSERPWD (nombre ˆ virgule flottante)
CURLOPT_RANGE (nombre ˆ virgule flottante)
CURLOPT_TIMEOUT (nombre ˆ virgule flottante)
CURLOPT_POSTFIELDS (nombre ˆ virgule flottante)
CURLOPT_REFERER (nombre ˆ virgule flottante)
CURLOPT_USERAGENT (nombre ˆ virgule flottante)
CURLOPT_FTPPORT (nombre ˆ virgule flottante)
CURLOPT_LOW_SPEED_LIMIT (nombre ˆ virgule flottante)
CURLOPT_LOW_SPEED_TIME (nombre ˆ virgule flottante)
CURLOPT_RESUME_FROM (nombre ˆ virgule flottante)
CURLOPT_COOKIE (nombre ˆ virgule flottante)
CURLOPT_SSLCERT (nombre ˆ virgule flottante)
CURLOPT_SSLCERTPASSWD (nombre ˆ virgule flottante)
CURLOPT_WRITEHEADER (nombre ˆ virgule flottante)
CURLOPT_SSL_VERIFYHOST (nombre ˆ virgule flottante)
CURLOPT_COOKIEFILE (nombre ˆ virgule flottante)
CURLOPT_SSLVERSION (nombre ˆ virgule flottante)
CURLOPT_TIMECONDITION (nombre ˆ virgule flottante)
CURLOPT_TIMEVALUE (nombre ˆ virgule flottante)
CURLOPT_CUSTOMREQUEST (nombre ˆ virgule flottante)
CURLOPT_STDERR (nombre ˆ virgule flottante)
CURLOPT_TRANSFERTEXT (nombre ˆ virgule flottante)
CURLOPT_RETURNTRANSFER (nombre ˆ virgule flottante)
CURLOPT_QUOTE (nombre ˆ virgule flottante)
CURLOPT_POSTQUOTE (nombre ˆ virgule flottante)
CURLOPT_INTERFACE (nombre ˆ virgule flottante)
CURLOPT_KRB4LEVEL (nombre ˆ virgule flottante)
CURLOPT_HTTPPROXYTUNNEL (nombre ˆ virgule flottante)
CURLOPT_FILETIME (nombre ˆ virgule flottante)
CURLOPT_WRITEFUNCTION (nombre ˆ virgule flottante)
CURLOPT_READFUNCTION (nombre ˆ virgule flottante)
CURLOPT_PASSWDFUNCTION (nombre ˆ virgule flottante)
CURLOPT_HEADERFUNCTION (nombre ˆ virgule flottante)
CURLOPT_MAXREDIRS (nombre ˆ virgule flottante)
CURLOPT_MAXCONNECTS (nombre ˆ virgule flottante)
CURLOPT_CLOSEPOLICY (nombre ˆ virgule flottante)
CURLOPT_FRESH_CONNECT (nombre ˆ virgule flottante)
CURLOPT_FORBID_REUSE (nombre ˆ virgule flottante)
CURLOPT_RANDOM_FILE (nombre ˆ virgule flottante)
CURLOPT_EGDSOCKET (nombre ˆ virgule flottante)
CURLOPT_CONNECTTIMEOUT (nombre ˆ virgule flottante)
CURLOPT_SSL_VERIFYPEER (nombre ˆ virgule flottante)
CURLOPT_CAINFO (nombre ˆ virgule flottante)
CURLOPT_COOKIEJAR (nombre ˆ virgule flottante)
CURLOPT_SSL_CIPHER_LIST (nombre ˆ virgule flottante)
CURLOPT_BINARYTRANSFER (nombre ˆ virgule flottante)
CURLCLOSEPOLICY_LEAST_RECENTLY_USED (nombre ˆ virgule flottante)
CURLCLOSEPOLICY_LEAST_TRAFFIC (nombre ˆ virgule flottante)
CURLCLOSEPOLICY_SLOWEST (nombre ˆ virgule flottante)

CURLCLOSEPOLICY_CALLBACK (nombre ˆ virgule flottante)
CURLCLOSEPOLICY_OLDEST (nombre ˆ virgule flottante)
CURLINFO_EFFECTIVE_URL (nombre ˆ virgule flottante)
CURLINFO_HTTP_CODE (nombre ˆ virgule flottante)
CURLINFO_HEADER_SIZE (nombre ˆ virgule flottante)
CURLINFO_REQUEST_SIZE (nombre ˆ virgule flottante)
CURLINFO_TOTAL_TIME (nombre ˆ virgule flottante)
CURLINFO_NAMELOOKUP_TIME (nombre ˆ virgule flottante)
CURLINFO_CONNECT_TIME (nombre ˆ virgule flottante)
CURLINFO_PRETRANSFER_TIME (nombre ˆ virgule flottante)
CURLINFO_SIZE_UPLOAD (nombre ˆ virgule flottante)
CURLINFO_SIZE_DOWNLOAD (nombre ˆ virgule flottante)
CURLINFO_SPEED_DOWNLOAD (nombre ˆ virgule flottante)
CURLINFO_SPEED_UPLOAD (nombre ˆ virgule flottante)
CURLINFO_FILETIME (nombre ˆ virgule flottante)
CURLINFO_SSL_VERIFYRESULT (nombre ˆ virgule flottante)
CURLINFO_CONTENT_LENGTH_DOWNLOAD (nombre ˆ virgule flottante)
CURLINFO_CONTENT_LENGTH_UPLOAD (nombre ˆ virgule flottante)
CURLE_OK (nombre ˆ virgule flottante)
CURLE_UNSUPPORTED_PROTOCOL (nombre ˆ virgule flottante)
CURLE_FAILED_INIT (nombre ˆ virgule flottante)
CURLE_URL_MALFORMAT (nombre ˆ virgule flottante)
CURLE_URL_MALFORMAT_USER (nombre ˆ virgule flottante)
CURLE_COULDNT_RESOLVE_PROXY (nombre ˆ virgule flottante)
CURLE_COULDNT_RESOLVE_HOST (nombre ˆ virgule flottante)
CURLE_COULDNT_CONNECT (nombre ˆ virgule flottante)
CURLE_FTP_WEIRD_SERVER_REPLY (nombre ˆ virgule flottante)
CURLE_FTP_ACCESS_DENIED (nombre ˆ virgule flottante)
CURLE_FTP_USER_PASSWORD_INCORRECT (nombre ˆ virgule flottante)
CURLE_FTP_WEIRD_PASS_REPLY (nombre ˆ virgule flottante)
CURLE_FTP_WEIRD_USER_REPLY (nombre ˆ virgule flottante)
CURLE_FTP_WEIRD_PASV_REPLY (nombre ˆ virgule flottante)
CURLE_FTP_WEIRD_227_FORMAT (nombre ˆ virgule flottante)
CURLE_FTP_CANT_GET_HOST (nombre ˆ virgule flottante)
CURLE_FTP_CANT_RECONNECT (nombre ˆ virgule flottante)
CURLE_FTP_COULDNT_SET_BINARY (nombre ˆ virgule flottante)
CURLE_PARTIAL_FILE (nombre ˆ virgule flottante)
CURLE_FTP_COULDNT_RETR_FILE (nombre ˆ virgule flottante)
CURLE_FTP_WRITE_ERROR (nombre ˆ virgule flottante)
CURLE_FTP_QUOTE_ERROR (nombre ˆ virgule flottante)
CURLE_HTTP_NOT_FOUND (nombre ˆ virgule flottante)
CURLE_WRITE_ERROR (nombre ˆ virgule flottante)
CURLE_MALFORMAT_USER (nombre ˆ virgule flottante)
CURLE_FTP_COULDNT_STOR_FILE (nombre ˆ virgule flottante)
CURLE_READ_ERROR (nombre ˆ virgule flottante)
CURLE_OUT_OF_MEMORY (nombre ˆ virgule flottante)
CURLE_OPERATION_TIMEOUTED (nombre ˆ virgule flottante)
CURLE_FTP_COULDNT_SET_ASCII (nombre ˆ virgule flottante)
CURLE_FTP_PORT_FAILED (nombre ˆ virgule flottante)
CURLE_FTP_COULDNT_USE_REST (nombre ˆ virgule flottante)
CURLE_FTP_COULDNT_GET_SIZE (nombre ˆ virgule flottante)
CURLE_HTTP_RANGE_ERROR (nombre ˆ virgule flottante)
CURLE_HTTP_POST_ERROR (nombre ˆ virgule flottante)
CURLE_SSL_CONNECT_ERROR (nombre ˆ virgule flottante)

CURLE_FTP_BAD_DOWNLOAD_RESUME (nombre ˆ virgule flottante)
CURLE_FILE_COULDNT_READ_FILE (nombre ˆ virgule flottante)
CURLE_LDAP_CANNOT_BIND (nombre ˆ virgule flottante)
CURLE_LDAP_SEARCH_FAILED (nombre ˆ virgule flottante)
CURLE_LIBRARY_NOT_FOUND (nombre ˆ virgule flottante)
CURLE_FUNCTION_NOT_FOUND (nombre ˆ virgule flottante)
CURLE_ABORTED_BY_CALLBACK (nombre ˆ virgule flottante)
CURLE_BAD_FUNCTION_ARGUMENT (nombre ˆ virgule flottante)
CURLE_BAD_CALLING_ORDER (nombre ˆ virgule flottante)
CURLE_HTTP_PORT_FAILED (nombre ˆ virgule flottante)
CURLE_BAD_PASSWORD_ENTERED (nombre ˆ virgule flottante)
CURLE_TOO_MANY_REDIRECTS (nombre ˆ virgule flottante)
CURLE_UNKNOWN_TELNET_OPTION (nombre ˆ virgule flottante)
CURLE_TELNET_OPTION_SYNTAX (nombre ˆ virgule flottante)
CURLE_OBSOLETE (nombre ˆ virgule flottante)
CURLE_SSL_PEER_CERTIFICATE (nombre ˆ virgule flottante)

6.12.5 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_setopt
- curl_version

6.12.7 curl_errno() : Retourne un entier représentant le dernier message d'erreur.

`int` **curl_errno** (`resource ch`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.12.8 curl_error() : Retourne une chaîne contenant le dernier message d'erreur

`string` **curl_error** (`resource ch`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.12.9 curl_exec() : Exécute une session CURL

`boolean` **curl_exec** (`resource ch`)
[lien librairie]

Cette fonction doit être appelée après l'initialisation et le paramètrage d'une session CURL. Son but est simplement d'éxécuter la session `ch` .

6.12.10 curl_getinfo() : Lit des informations concernant un transfert particulier

`string` **curl_getinfo** (`resource ch` , `int opt`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.12.11 curl_init() : Initialise une session CURL

`resource` **curl_init** (*`string url`*)
[lien librairie]

`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 récupèration d'une page web.

```
<?php
$ch = curl_init();
curl_setopt ($ch, CURLOPT_URL, "http://www.zend.com/");
curl_setopt ($ch, CURLOPT_HEADER, 0);
curl_exec ($ch);
curl_close ($ch);
?>
```

Voir aussi : `curl_close` , `curl_setopt` .

6.12.12 curl_setopt() : Modifie une option de transfert CURL

`boolean` **curl_setopt** (`resource ch` , `string option` , `mixed value`)
[lien librairie]

`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`) :

- `CURLOPT_INFILESIZE` : Lorsque vous téléchargez un fichier sur un site distant, cette option sert à indiquer à PHP la taille maximale du fichier attendu.
- `CURLOPT_VERBOSE` : Choisissez une valeur non nulle pour que CURL vous affiche tous les événements.
- `CURLOPT_HEADER` : Choisissez une valeur non nulle pour que CURL inclut l'en−tête dans la valeur de retour.
- `CURLOPT_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 débuggage.

- `CURLOPT_NOBODY` : Choisissez une valeur non nulle pour que le corps du transfert ne soit pas inclus dans la valeur de retour.
- `CURLOPT_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.
- `CURLOPT_UPLOAD` : Choisissez une valeur non nulle pour que PHP prépare un chargement.
- `CURLOPT_POST` : Choisissez une valeur non nulle pour que PHP fasse un HTTP POST. Un POST est un encodage normal "application/x−www−from−url", utilisé couramment par les formulaires HTML.
- `CURLOPT_FTPLISTONLY` : Choisissez une valeur non nulle pour que PHP ne fasse que lister les noms d'un dossier FTP.
- `CURLOPT_FTPAPPEND` : Choisissez une valeur non nulle pour que PHP concatène le fichier distant, plutôt que de l'écraser.
- `CURLOPT_NETRC` : Choisissez une valeur non nulle pour que PHP scanne votre fichier ~./netrc et utilise votre nom de compte et mot de passe sur le site distant que vous souhaitez contacter.
- `CURLOPT_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).
- `CURLOPT_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.

- CURLOPT_MUTE : Choisissez une valeur non nulle pour que PHP soit totalement silencieux concernant toutes les fonctions CURL.
- CURLOPT_TIMEOUT : Passez un entier "long" comme paramètre qui représente le temps maximum d'exécution de la fonction CURL.
- CURLOPT_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 considèrera qu'elle est trop lente, et annulera le transfert.
- CURLOPT_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 considèrera que la connexion est trop lente, et l'annulera.
- CURLOPT_RESUME_FROM : Passez un entier "long", qui représente l'offset, en octets, à partir duquel vous voulez commencer le transfert.
- CURLOPT_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.
- CURLOPT_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.
- CURLOPT_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

- CURLOPT_URL : L'URL que PHP va récupérer. Vous pouvez aussi choisir cette valeur lors de l'appel à curl_init .
- CURLOPT_USERPWD : Passez une chaîne de caractères au format [nom]:[mot de passe], pour que PHP l'utilise lors de la connexion.
- CURLOPT_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.
- CURLOPT_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.
- CURLOPT_POSTFIELDS : Passez une chaîne de caractères qui contient toutes les données à passer lors d'une opération de HTTP POST.
- CURLOPT_REFERER : Passez une chaîne de caractères qui contient l'en−tête de "REFERER", utilisé lors d'une requête HTTP.
- CURLOPT_USERAGENT : Passez une chaîne de caractères qui contient l'en−tête "user−agent" utilisé dans une requête HTTP.
- CURLOPT_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.
- CURLOPT_COOKIE : Passez une chaîne de caractères qui contiendra le contenu du cookie, à transmettre dans l'en−tête HTTP.
- CURLOPT_SSLCERT : Passez une chaîne de caractères qui contiendra le nom de fichier du certificat, au format PEM.
- CURLOPT_SSLCERTPASSWD : Passez une chaîne de caractères qui contient le mot de passe nécessaire pour utiliser le certificat CURLOPT_SSLCERT.
- CURLOPT_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.

- CURLOPT_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` :

- CURLOPT_FILE : Le fichier de sortie de votre transfert. Par défaut, STDOUT.
- CURLOPT_INFILE : Le fichier d'entrée de votre transfert.
- CURLOPT_WRITEHEADER : Le fichier de destination de l'en−tête de la sortie du transfert.
- CURLOPT_STDERR : Le fichier d'erreurs.

6.12.13 curl_version() : Retourne la version courante de CURL

`string` **curl_version** (`void`)
[lien librairie]

`curl_version` retourne une chaîne avec la version courante de la librairie CURL.

6.13 Paiement Cybercash

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.

- cybercash_base64_decode
- cybercash_base64_encode
- cybercash_decr
- cybercash_encr

6.13.2 cybercash_base64_encode() :

`string` **cybercash_base64_encode** (`string inbuff`)
[lien librairie]

6.13.3 cybercash_decr() :

`array` **cybercash_decr** (`string wmk` , `string sk` , `string inbuff`)
[lien librairie]

`cybercash_decr` retourne un tableau associatif, contenant les éléments "errcode" et, si "errcode" vaut `FALSE` , "outbuff" (string), "outLth" (long) et "macbuff" (string).

6.13.4 cybercash_encr() :

`array` **cybercash_encr** (`string wmk` , `string sk` , `string inbuff`)
[lien librairie]

`cybercash_encr` retourne un tableau associatif, contenant les éléments "errcode" et, si "errcode" vaut `FALSE` , "outbuff" (string), "outLth" (long) et "macbuff" (string).

6.14 CyberMUT : Crédit Mutuel

Cette extension vous permet de traiter des transactions de cartes de crédits, avec le système due Crédit Mutuel : CyberMUT (http://www.creditmutuel.fr/centre_commercial/vendez_sur_internet.html).

CynerMUT est un système de paiement français, proposé par le Crédit Mutuel. Si vous n'êtes pas résidants français, ces fonctions ne vous seront pas utiles.

Cette extension n'est disponible que si PHP a été compilé par l'option `--with-cybermut[=DIR]` , où DIR est le dossier qui contient les fichiers `libcm-mac.a` et `cm-mac.h` . Vous aurez besoin du SDK approprié, qui vous est fournit après vous êtes inscrit à CyberMUT (via le web, ou à votre agence la plus proche).

L'utilisation de ces fonctions est presque identique aux fonctions originales, hormis le fait que les fonctions `cybermut_creerformulairecm` et `cybermut_creerreponsecm` , qui sont retournées directement par des fonctions PHP, au lieu d'être passées par référence aux fonctions originales.

Ces fonctions ont été ajoutée en 4.0.6.

Note
Ces fonctions ne font que fournis un moyen d'utiliser le SDK CyberMUT. Lisez attentivement le "CyberMUT Developers Guide" pour plus de détails sur les parmètres nécessaires.

- cybermut_creerformulairecm
- cybermut_creerreponsecm
- cybermut_testmac

6.14.2 cybermut_creerreponsecm() : Génère un accusé de réception de confirmation de paiement

`string` **cybermut_creerreponsecm** (`string phrase`)
[lien librairie]

`cybermut_creerreponsecm` retourne une chaîne contenant le message d'accusé de reception.

Le paramètre vaut "OK" si le message de confirmation du paiement a été correctement indentifié par `cybermut_testmac` . Tout autre chaîne doit être considéré comme une erreur de traitement.

Voir aussi `cybermut_creerformulairecm` et `cybermut_testmac` .

6.14.3 cybermut_testmac() : Vérifie le message de confirmation

`bool` **cybermut_testmac** (`string code_MAC` , `string version` , `string TPE` , `string cdate` , `string montant` , `string ref_commande` , `string texte_libre` , `string code-retour`)
[lien librairie]

`cybermut_testmac` s'assure qu'il n'y a pas de données parasites dans le message de confirmation recu. Attention aux paramètres `code-retour` and `texte-libre` , qui ne peuvent

pas être utilisés directement, car ils contienent des tirets dans leur nom. Vous devez utilisez la syntaxe suivante :

```
<?php
  $code_retour=$HTTP_GET_VARS["code-retour"];
  $texte_libre=$HTTP_GET_VARS["texte-libre"];
?>
```

Deuxième étape de paiement (équivalent à cgi2.c)

```
<?php
// Assurez vous que l'option Enable Track Vars est active.
// Dossier qui contient les clés de paiement
putenv("CMKEYDIR=/var/creditmut/cles");
// Numéro de version
$VERSION="1.2";
$texte_libre = $HTTP_GET_VARS["texte-libre"];
$code_retour = $HTTP_GET_VARS["code-retour"];
$mac_ok = creditmut_testmac($MAC,$VERSION,$TPE,$date,$montant,$reference,$texte_libre,$code_retour);
if ($mac_ok) {
  //
  // Gestion d'un paiement réussi
  //
  //
  $result=creditmut_creerreponsecm("OK");
  } else {
  $result=creditmut_creerreponsecm("Document Falsifié");}
?>
```

Voir aussi `cybermut_creerformulairecm` et `cybermut_creerreponsecm` .

6.15 Cyrus IMAP

6.15.1 Introduction

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.15.2 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 (nombre ˆ virgule flottante)
CYRUS_CONN_INITIALRESPONSE (nombre ˆ virgule flottante)
CYRUS_CALLBACK_NUMBERED (nombre ˆ virgule flottante)
CYRUS_CALLBACK_NOLITERAL (nombre ˆ virgule flottante)

- Introduction
- Constantes Prédefinies
- cyrus_authenticate
- cyrus_bind
- cyrus_close
- cyrus_connect
- cyrus_query
- cyrus_unbind

6.15.4 cyrus_bind() : Ajoute une fonction de callback sur une connexion Cyrus IMAP

`bool` **cyrus_bind** (`resource connection` , `array callbacks`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.15.5 cyrus_close() : Ferme la connexion à un serveur Cyrus IMAP

`bool` **cyrus_close** (`resource connection`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.15.6 cyrus_connect() : Se connecte à un serveur Cyrus IMAP

`resource` **cyrus_connect** (*`string host`* , *`string port`* , *`int flags`*)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.15.7 cyrus_query() : Envoye une requête à un serveur Cyrus IMAP

`bool` **cyrus_query** (`resource connection` , `string query`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.15.8 cyrus_unbind() : Supprime une fonction de callback sur une connexion Cyrus IMAP

`bool` **cyrus_unbind** (`resource connection` , `string trigger_name`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.16 Caractères

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.

Attention
Ces fonctions ont été ajoutée en PHP 4.0.4, et leur nom peut changer dans un futur proche. Les suggestions actuelles sont : `ctype_issomething()` au lieu de `ctype_somthing` ou encore d'en faire une partie `ext/standard` et utiliser ainsi leur nom en langage C, même si cela peut conduire à des confusions entre `isset` et `is_sometype` .

- ctype_alnum
- ctype_alpha
- ctype_cntrl
- ctype_digit
- ctype_graph
- ctype_lower
- ctype_print
- ctype_punct
- ctype_space
- ctype_upper
- ctype_xdigit

6.16.2 ctype_alpha() : Vérifie qu'un caractère est alphabétique

`boolean` **ctype_alpha** `( string c )`
[lien librairie]

6.16.3 ctype_cntrl() : Vérifie qu'un caractère est un caractère de contrôle

`boolean` **ctype_cntrl** `( string c )`
[lien librairie]

6.16.4 ctype_digit() : Vérifie qu'un caractère est numérique

`boolean` **ctype_digit** `( string c )`
[lien librairie]

6.16.5 ctype_graph() : Vérifie qu'un caractère est imprimable (sauf " ", espace)

`boolean` **ctype_graph** `( string c )`
[lien librairie]

6.16.6 ctype_lower() : Vérifie qu'un caractère est en minuscule

`boolean` **ctype_lower** `( string c )`
[lien librairie]

6.16.7 ctype_print() : Vérifie qu'un caractère est imprimable

`boolean` **ctype_print** `( string c )`
[lien librairie]

6.16.8 ctype_punct() : Vérifie qu'un caractère est imprimable, sans être ni un espace, ni un caractère alpha−numérique

`boolean` **ctype_punct** `( string c )`
[lien librairie]

6.16.9 ctype_space() : Vérifie qu'un caractère est caractère blanc (espace, tabulation...)

`boolean` **ctype_space** `( string c )`
[lien librairie]

6.16.10 ctype_upper() : Vérifie qu'un caractère est en majuscule

`boolean` **ctype_upper** `( string c )`
[lien librairie]

6.16.11 ctype_xdigit() : Vérifie qu'un caractère représente un nombre héxadécimal

`boolean` **ctype_xdigit** `( string c )`
[lien librairie]

6.17 DBA

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).

Le comportement de certaines fonctions dépends de la base de données utilisée. Par exemple `dba_optimize` et `dba_sync` n'auront pas le même effet d'une base à l'autre.

Lors de l'utilisation de la fonction `dba_open` ou de `dba_popen` , une des librairies suivante doit être fournie comme argument. La liste complète des librairies supportées par votre configuration est disponible avec la fonction `phpinfo` . (Pour ajouter le support de l'une de ces librairies, ajouter l'option de configuration `--with-XXXX`).

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. (`--with-dbm`).
`ndbm`	ndbm est un nouveau type de dbm plus flexible. Il a cependant la majorité des limitations du genre. (`--with-ndbm`).
`gdbm`	gdbm est la base dbm GNU. (`--with-gdbm`).
`db2`	db2 est DB2 de Sleepycat Software. 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". (`--with-db2`).
`db3`	DB3 est le DB3 de Sleepycat Software. (`--with-db3`).
`cdb`	cdb est "un package rapide, robuste, léger, pour créer et lire des bases de données constantes". C'est l'auteur de qmail qui l'a écrit, et elle est disponible ici. Puisque c'est une base constante, elle ne supporte que la lecture. (--with-cdb).

Exemple DBA

```
<?php
$id = dba_open("/tmp/test.db", "n", "db2");
if(!$id) {
    echo "dba_open a échoué\n";
    exit;
}
dba_replace("key", "Ceci est un exemple!", $id);
if(dba_exists("key", $id)) {
    echo dba_fetch("key", $id);
    dba_delete("key", $id);
}
dba_close($id);
?>
```

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);
?>
```

- 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

6.17.2 dba_delete() : Efface une entrée.

bool **dba_delete** (string key , resource handle)
[lien librairie]

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

6.17.3 dba_exists() : Vérifie qu'une clé existe.

`boolean` **dba_exists** (`string key` , `resource handle`)
[lien librairie]

`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` .

6.17.4 dba_fetch() : Lit les données liées à une clé.

`string` **dba_fetch** (`string key` , `resource handle`)
[lien librairie]

`dba_fetch` lit les données spécifiée par la clé `key` dans la base identifiée par `handle` .

`key` est la clé dont on veut lire les données.

`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.

Voir aussi `dba_exists` , `dba_delete` , `dba_insert` et `dba_replace` .

6.17.5 dba_firstkey() : Lit la première clé.

`string` **dba_firstkey** (`resource handle`)
[lien librairie]

`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 .

6.17.6 dba_handlers() : Liste les gestionnaires dba disponibles

`array` **dba_handlers** (`void`)
[lien librairie]

`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'.

6.17.7 dba_insert() : Insère une entrée.

`boolean` **dba_insert** (`string key` , `string value` , `resource handle`)
[lien librairie]

`dba_insert` insère l'entrée décrite par la clé `key` et la valeur `value` dans la base spécifiée par `handle` . Si une entrée aveec 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` .

6.17.8 dba_list() : Liste tous les fichiers de bases de données ouverts

`array` **dba_list** (`void`)
[lien librairie]

`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` .

6.17.9 dba_nextkey() : Lit la clé suivante.

`string` **dba_nextkey** (`resource handle`)
[lien librairie]

`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` .

6.17.10 dba_open() : Ouvre une base de données.

`int` **dba_open** (`string path` , `string mode` , `string handler` , *`mixed ...`*)
[lien librairie]

`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 .

6.17.11 dba_optimize() : Optimise une base.

boolean **dba_optimize** (resource handle)
[lien librairie]

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 .

6.17.12 dba_popen() : Ouvre une connexion persistante à une base de données.

int **dba_popen** (string path , string mode , string handler , *mixed ...*)
[lien librairie]

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.

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.

Voir aussi dba_open et dba_close .

6.17.13 dba_replace() : Remplace ou insère une entrée.

boolean **dba_replace** (string key , string value , resource handle)
[lien librairie]

dba_replace remplaces 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 .

6.17.14 dba_sync() : Synchronise une base de données.

boolean **dba_sync** (resource handle)
[lien librairie]

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 réussi, ou échoue, respectivement.

Voir aussi dba_optimize .

6.18 Dates et heures

6.18.1 checkdate() : Valide une date/heure.

`int` **checkdate** (`int month` , `int day` , `int year`)
[lien librairie]

`checkdate` retourne `TRUE` si la date représentée par le jour `day` , le mois `month` et l'année `year` est valide, et sinon `FALSE` . Notez bien que l'ordre des arguments n'est pas l'ordre français. La date est considérée comme valide si :

- L'année est comprise entre entre 1 et 32767 inclus. (pour les versions antérieures à PHP 4.0.3, les années inférieures à 1 étaient aussi valides).
- Le mois est compris entre 1 et 12 inclus
- Le jour est compris dans l'intervalle de dates du mois. Les années bissextiles sont prises en compte.

- checkdate
- date
- getdate
- gettimeofday
- gmdate
- gmmktime
- gmstrftime
- localtime
- microtime
- mktime
- strftime
- strtotime
- time

6.18.3 getdate() : Retourne la date/heure

`array` **getdate** (*`int timestamp`*)
[lien librairie]

`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
<?php
  $aujourdhui = getdate();
  $mois = $aujourdhui['month'];
  $mjour = $aujourdhui['mday'];
  $annee = $aujourdhui['year'];
  echo "$mjour/$mois/$annee";
?>
```

6.18.4 gettimeofday() : Retourne l'heure actuelle

`array` **gettimeofday** (`void`)
[lien librairie]

`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

6.18.5 gmdate() : Formate une date/heure GMT/CUT.

`string` **gmdate** (`string format` , *`int timestamp`*)
[lien librairie]

`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
<?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));
?>
```

Voir aussi `date` , `strftime` , `mktime` et `gmmktime` .

6.18.6 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`*)
[lien librairie]

Identique à `mktime` hormis le fait que les paramètres passés sont GMT.

6.18.7 gmstrftime() : Formate une date/heure GMT/CUT en fonction des paramétrages locaux.

`string` **gmstrftime** (`string format` , *`int timestamp`*)
[lien librairie]

`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` .

6.18.8 localtime() : Lit l'heure locale

`array` **localtime** (*`int timestamp`* , *`boolean is_associative`*)
[lien librairie]

`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?

6.18.9 microtime() : Retourne le timestamp UNIX actuel avec microsecondes.

`string` **microtime** (`void`)
[lien librairie]

`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++){
    //ne rien faire, pendant un millier de fois...
  }
  $time_end = getmicrotime();
  $time = $time_end - $time_start;
  echo "Rien fait durant $time secondes";
?>
```

Voir aussi `time` .

6.18.10 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`*)
[lien librairie]

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

`is_dst` 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 `mktime`

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

`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

```
<?php
  $lastday = mktime (0,0,0,3,0,2000);
  echo strftime ("Le dernier jour de Février 2000 est: %d", $lastday);
  $lastday = mktime (0,0,0,4,-31,2000);
  echo strftime ("Le dernier jour de Février 2000 est: %d", $lastday);
?>
```

Voir aussi `date` et `time` .

6.18.11 strftime() : Formate une date/heure locale avec les options locales.

`string` **strftime** (`string format` , *`int timestamp`*)
[lien librairie]

`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.

Exemple `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 est 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.

Voir aussi `strftime` , `setlocale` , `mktime` et le groupe de spécifications de strftime() .

6.18.12 strtotime() : Transforme un texte anglais en timestamp

`int` **strtotime** (`string time` , *`int now`*)
[lien librairie]

`strtotime` essaye de lire une date au format anglais dans la chaîne `time` , et de la transformer en timestamp UNIX.

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";
?>
```

6.18.13 time() : Retourne le timestamp UNIX actuel.

`int` **time** (`void`)
[lien librairie]

`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` .

6.19 dBase

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 PostgresSQL 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 organiseurs Windows. L'import et l'export de données sont les deux uniques fonctionnalités qui recommendent l'utilisation de dBase.

- dbase_add_record
- dbase_close
- dbase_create
- dbase_delete_record
- dbase_get_record_with_names
- dbase_get_record
- dbase_numfields
- dbase_numrecords
- dbase_open
- dbase_pack
- dbase_replace_record

6.19.2 dbase_close() : Ferme une base dBase.

`boolean` **dbase_close** (`resource dbase_identifier`)
[lien librairie]

`dbase_close` ferme la base associée à `dbase_identifier` .

6.19.3 dbase_create() : Crée une base de données dBase.

`int` **dbase_create** (`string filename` , `array fields`)
[lien librairie]

`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 :

L
: Boolean (booléen). Pas de longueur ou de précision pour ces valeurs.

M

Memo. (Note importante : les Memos ne sont pas supportés par PHP.) Elles n'ont pas de longueur ou de précision.

D

Date (enregistrée au format 'YYYYMMDD'). Elles n'ont pas de longueur ou de précision.

N

Number (nombre). Possède une longueur et un précision (le nombre de chiffres après la virgule).

C

String (chaîne).

Si la base de données a été créée, un identifiant de base dbase_identifier 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>";
?>
```

6.19.4 dbase_delete_record() : Efface un enregistrement dans une base dBase.

`boolean` **dbase_delete_record** (`resource dbase_identifier` , `int record`)
[lien librairie]

`dbase_delete_record` marque l'enregistrement `record` pour l'effacement, dans la base `dbase_identifier` . Pour effacer réellement l'enregistrement, il faut utiliser aussi `dbase_pack` .

6.19.5 dbase_get_record_with_names() : Lit un enregistrement dans une base, sous la forme d'un tableau associatif.

`array` **dbase_get_record_with_names** (`resource dbase_identifier` , `int record`)
[lien librairie]

`dbase_identifier` 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. (Les dates sont laissées au format chaîne).

6.19.6 dbase_get_record() : Lit un enregistrement dans une base dBase.

`array` **dbase_get_record** (`resource dbase_identifier` , `int record`)
[lien librairie]

`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. (Les dates sont laissées au format chaîne).

6.19.7 dbase_numfields() : Compte le nombre de champs d'une base dBase.

`int` **dbase_numfields** (`resource dbase_identifier`)
[lien librairie]

`dbase_numfields` retourne le nombre de champs (colonnes) de la base de données `dbase_identifier` . 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";
  }
?>
```

6.19.8 dbase_numrecords() : Compter le nombre d'enregistrements dans une base dBase.

`int` **dbase_numrecords** (`resource dbase_identifier`)
[lien librairie]

`dbase_numrecords` retourne le nombre d'enregistrements (lignes) dans la base `dbase_identifier` . 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)` .

6.19.9 dbase_open() : Ouverture d'une base dBase.

`int` **dbase_open** (`string filename` , `int flags`)
[lien librairie]

`flags` est un flag, comme pour la fonction open(). (Typiquement; 0 signifie lecture seule, 1 signifie écriture seule, et 2 écriture/lecture).

`dbase_open` retourne un identifiant de base de données, ou FALSE si la base n'a pas pu être sélectionnée.

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..

6.19.10 dbase_pack() : Compacte une base dBase.

`boolean` **dbase_pack** (`resource dbase_identifier`)
[lien librairie]

`dbase_pack` compacte la base de données `dbase_identifier` (effacement définitif de tous les enregistrements marqués pour l'effacement, avec la fonction `dbase_delete_record`).

6.19.11 dbase_replace_record() : Remplace un enregistrement dans une base dBase.

`boolean` **dbase_replace_record** (`resource dbase_identifier` , `array record` , `int dbase_record_number`)
[lien librairie]

`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`).

6.20 DBM

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).

Présentation de dbm

```
<?php
  $dbm = dbmopen("dernier", "w");
  if (dbmexists($dbm, $userid)) {
    $last_seen = dbmfetch($dbm, $userid);
  } else {
    dbminsert($dbm, $userid, time());
  }
  faire_quelquechose();
  dbmreplace($dbm, $userid, time());
  dbmclose($dbm);
?>
```

- dblist
- dbmclose
- dbmdelete
- dbmexists
- dbmfetch
- dbmfirstkey
- dbminsert
- dbmnextkey
- dbmopen
- dbmreplace

6.20.2 dbmclose() : Ferme une base de données dbm.

`boolean` **dbmclose** (`resource dbm_identifier`)
[lien librairie]

`dbmclose` déverrouille et ferme la base de données `dbm_identifier` .

6.20.3 dbmdelete() : Efface une valeur.

`boolean` **dbmdelete** (`resource dbm_identifier` , `string key`)
[lien librairie]

`dbmdelete` efface la valeur de la clé `key` , dans la base `dbm_identifier` .

`dbmdelete` retourne FALSE si la clé n'existe pas dans cette base.

6.20.4 dbmexists() : Indique si une valeur existe.

`boolean` **dbmexists** (`resource dbm_identifier` , `string key`)
[lien librairie]

`dbmexists` retourne TRUE s'il y a une valeur associée à la clé `key` .

6.20.5 dbmfetch() : Lit une valeur.

`string` **dbmfetch** (`resource dbm_identifier` , `string key`)
[lien librairie]

`dbmfetch` retourne la valeur associée à la clé `key` .

6.20.6 dbmfirstkey() : Lit la première clé.

`string` **dbmfirstkey** (`resource dbm_identifier`)
[lien librairie]

`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.

6.20.7 dbminsert() : Insère une valeur.

`int` **dbminsert** (`resource dbm_identifier` , `string key` , `string value`)
[lien librairie]

`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` .)

6.20.8 dbmnextkey() : Lit la clé suivante.

`string` **dbmnextkey** (`resource dbm_identifier` , `string key`)
[lien librairie]

`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.

```
<?php
  $cle = dbmfirstkey($dbm_id);
  while ($cle){
    echo "$cle = " . dbmfetch($dbm_id, $cle) . "\n";
    $cle = dbmnextkey($dbm_id, $cle);
  }
?>
```

6.20.9 dbmopen() : Ouvre une base de données dbm

`resource` **dbmopen** (`string filename` , `string flags`)
[lien librairie]

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 librairies internes, et Berkeley db crée le fichier `filename.db` . Notez que PHP dispose de son propre système de verrouillage des fichiers, qui s'additionne à celui éventuellement utilisé par les librairies. PHP n'efface jamais les fichiers `.lck` qu'il crée. Il les utilise comme inode fixe, sur lequel faire le verrouillage. Pour plus d'informations sur les fichiers dbm, reportez−vous à vos pages de manuel Unix (man) , ou bien chargez gdbm : `ftp://prep.ai.mit.edu/pub/gnu` .

6.20.10 dbmreplace() : Remplace une valeur.

`int` **dbmreplace** (`resource dbm_identifier` , `string key` , `string value`)
[lien librairie]

`dbmreplace` remplace la valeur courante par la valeur `value` pour la clé `key` , dans une base `dbm_identifier` .

`dbmreplace` crée la clé, si elle n'existe pas dans la base.

6.21 dbx

6.21.1 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.

6.21.2 Pré−requis

Pour pouvoir utiliser une base de données avec le module dbx, le module doit être soit linké, 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/ .

6.21.3 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` .

6.21.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
dbx.colnames_case	"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 de détails sur les 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` .

6.21.5 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.

6.21.6 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.

`DBX_MYSQL` (nombre ˆ virgule flottante)
`DBX_ODBC` (nombre ˆ virgule flottante)
`DBX_PGSQL` (nombre ˆ virgule flottante)
`DBX_MSSQL` (nombre ˆ virgule flottante)
`DBX_FBSQL` (nombre ˆ virgule flottante)
`DBX_OCI8` (nombre ˆ virgule flottante) CVS only
`DBX_SYBASECT` (nombre ˆ virgule flottante)
`DBX_PERSISTENT` (nombre ˆ virgule flottante)
`DBX_RESULT_INFO` (nombre ˆ virgule flottante)
`DBX_RESULT_INDEX` (nombre ˆ virgule flottante)
`DBX_RESULT_ASSOC` (nombre ˆ virgule flottante)
`DBX_COLNAMES_UNCHANGED` (nombre ˆ virgule flottante) CVS only
`DBX_COLNAMES_UPPERCASE` (nombre ˆ virgule flottante) CVS only
`DBX_COLNAMES_LOWERCASE` (nombre ˆ virgule flottante) CVS only
`DBX_CMP_NATIVE` (nombre ˆ virgule flottante)
`DBX_CMP_TEXT` (nombre ˆ virgule flottante)
`DBX_CMP_NUMBER` (nombre ˆ virgule flottante)
`DBX_CMP_ASC` (nombre ˆ virgule flottante)
`DBX_CMP_DESC` (nombre ˆ virgule flottante)

- 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_query
- dbx_sort

6.21.8 dbx_compare() : Compare deux lignes afin de les trier

`int` **dbx_compare** (`array row_a` , `array row_b` , `string column_key` , *`int flags`*)
[lien librairie]

`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` .

6.21.9 dbx_connect() : Ouvre une connexion à une base de données

`resource` **dbx_connect** (`string module` , `string host` , `string database` , `string username` , `string password` , *`int persistent`*)
[lien librairie]

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 utiliser ce module à vos risques et périls.

`dbx_connect` retourne une resource 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 persitante.

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` .

```
<?php
$link = dbx_connect("mysql", "localhost", "base de données", "utilisateur", "mot de passe");
mysql_close($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
$link = dbx_connect("odbc", "", "base de données", "utilisateur", "mot de passe", DBX_PERSISTENT)
    or die ("Impossible de se connecter");
print ("Connexion réussie");
dbx_close($link);
?>
```

Note
Reportez−vous aussi à la documentation de la base de données que vous utilisez.

Voir aussi `dbx_close` .

6.21.10 dbx_error() : Rapporte le message d'erreur du dernier appel de fonction

`string` **dbx_error** (`resource link_identifier`)
[lien librairie]

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 utiliser ce module à vos risques et périls.

`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 des connexions sur différents modules, la dernière erreur du module est retourné (le module est alors représenté par `link_identifier`). 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.

6.21.11 dbx_escape_string() : Protège une chaîne de caractères pour l'utiliser dans une requête

`string` **dbx_escape_string** (`object link_identifier` , `string text`)
[lien librairie]

`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 .

6.21.12 dbx_query() : Envoie une requête et lit tous les résultats

resource **dbx_query** (resource link_identifier , string sql_statement , *long flags*)
[lien librairie]

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 utiliser ce module à vos risques et périls.

dbx_query retourne une ressource 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. Le paramètre flags sert à contrôler la quantité d'informations retournée. Il peut être n'importe quelle combinaisons par OR des constantes : DBX_RESULT_INFO, DBX_RESULT_INDEX, DBX_RESULT_ASSOC. DBX_RESULT_INFO fournit des informations sur les colonnes, comme les noms des champs et leur type. DBX_RESULT_INDEX retourne le résultat dans un tableau indexé (par exemple, data[2][3], où 2 est le numéro de ligne et 3 est le numéro de colonne), dont la première colonne est indexée à 0. DBX_RESULT_ASSOC associe les noms de colonnes avec leurs indices. Notez que DBX_RESULT_INDEX est toujours retourné, indépendamment de la valeur de flags . Si DBX_RESULT_ASSOC est spécifié, DBX_RESULT_INFO est aussi retourné, même s'il n'a pas été spécifié. Ce qui signifie que les seules combinaisons envisageables sont DBX_RESULT_INDEX, DBX_RESULT_INDEX | DBX_RESULT_INFO et DBX_RESULT_INDEX | DBX_RESULT_INFO | DBX_RESULT_ASSOC. La dernière combinaison est la valeur par défaut de flags . Les résultats associés sont en fait des références, ce qui fait que modifier data[0][0] , revient à modifier data[0]['fieldnameforfirstcolumn'] .

Un objet dbx_result_object a 5 membres (éventuellement 4, suivants les valeurs de flags) : 'handle', 'cols', 'rows', 'info' (optionnel) et 'data'. Handle est un identifiant de résultat, qui peut être utilisé avec les fonctions spécifiques à son module. Par exemple :

```
<?php
$result = dbx_query($link, "SELECT id FROM tbl");
mysql_field_len($result->handle, 0);
?>
```

Les membres cols et rows contiennent les numéros de colonne et de champs.

```
<?php
$result = dbx_query($link, "SELECT id FROM tbl");
echo "Taille du résultat: " . $result->rows . " x " . $result->cols . "<br>\n";
?>
```

Le membre info n'est retourné que si DBX_RESULT_INFO et/ou DBX_RESULT_ASSOC sont spécifié dans le paramètre `flags` . C'est un deuxième tableau, qui possède deux lignes ("name" and "type"), pour connaître les informations sur les colonnes.

```
<?php
$result = dbx_query($link, "SELECT id FROM tbl");
echo "Nom de la colonne : " . $result->info["name"][0] . "<br>\n";
echo "Type de la colonne: " . $result->info["type"][0] . "<br>\n";
?>
```

Le membre data contient les données effectivement lues, éventuellement associées à des noms de colonnes. Si DBX_RESULT_ASSOC est utilisé, il est possible d'utiliser `$result->data[2]["fieldname"]` .

Exemple avec `dbx_query`

```
<?php
$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");
if ($result==0) echo "La requête a échoué\n<br>";
elseif ($result==1) {
    echo "La requête a réussie\n<br>";
} else {
    $rows=$result->rows;
    $cols=$result->cols;
    echo "<p>table dimension: {$result->rows} x {$result->cols}<br><table border=1>\n";
    echo "<tr>";
    for ($col=0; $col<$cols; ++$col) {
        echo "<td>-{$result->info["name"][$col]}-<br>-{$result->info["type"][$col]}-</td>";
    }
    echo "</tr>\n";
    for ($row=0; $row<$rows; ++$row){
        echo "<tr>";
        for ($col=0; $col<$cols; ++$col) {
            echo "<td>-{$result->data[$row][$col]}-</td>";
        }
        echo "</tr>\n";
    }
    echo "</table><p>\n";
    echo "table dimension: {$result->rows} x id, parentid, description<br><table border=1>\n";
    for ($row=0; $row<$rows; ++$row) {
        echo "<tr>";
        echo "<td>-{$result->data[$row]["id"]}-</td>";
        echo "<td>-{$result->data[$row]["parentid"]}-</td>";
        echo "<td>-{$result->data[$row]["description"]}-</td>";
        echo "</tr>\n";
    }
    echo "</table><p>\n";
}
dbx_close($link);
?>
```

Note

Reportez-vous aussi à la documentation de la base de données que vous utilisez.

Voir aussi `dbx_connect` .

6.21.13 dbx_sort() : Tri un résultat avec une fonction utilisateur

`boolean` **dbx_sort** (`dbx_result_object result` , `string user_compare_function`)
[lien librairie]

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 utiliser ce module à vos risques et périls.

`dbx_sort` retourne TRUE en cas de succès, et FALSE sinon.

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_cmp_asc` et `dbx_cmp_desc` .

6.22 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 utiliser ce module à vos risques et périls.

6.22.1 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 OpenSources, et DB++ dispose d'interface Perl et Tcl depuis longtemps. Il utilise aussi Tcl pour ses procédures stockées internes.

6.22.2 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 version 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).

6.22.3 Installation

Pour compiler cette extension par vous même, vous devez avoir les librairies clients et les fichiers d'entêtes 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'entêtes dans les dossiers suivants : `/usr/dbplus` , `/usr/local/dbplus` et `/opt/dblus` . Si vous avez installé Db++ dans un autre dossier, indiquez le chemin de ce dossier dans l'option de `configure` comme ceci : `−−with−dbplus=/your/installation/path` .

6.22.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.22.5 Types de ressources

6.22.5.1 dbplus_relation

La plupart des fonctions DB++ travaille avec une ressource de type `dbplus_relation` : c'est un pointeur sur une relation, ou une relation, générée par une requête.

6.22.6 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.

6.22.6.1 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 de 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'analyser

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	Imopssible 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 survenu 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_rcrtexact

- dbplus_rcrtlike
- dbplus_resolve
- dbplus_restorepos
- dbplus_rkeys
- dbplus_ropen
- dbplus_rquery
- dbplus_rrename
- dbplus_rsecindex
- dbplus_runlink
- dbplus_rzap
- dbplus_savepos
- dbplus_setindex
- dbplus_setindexbynumber
- dbplus_sql
- dbplus_tcl
- dbplus_tremove
- dbplus_undo
- dbplus_undoprepare
- dbplus_unlockrel
- dbplus_unselect
- dbplus_update
- dbplus_xlockrel
- dbplus_xunlockrel

6.22.8 dbplus_aql() : Exécute une requête AQL

`resource` **dbplus_aql** (`string query` , *`string server`* , *`string dbpath`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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'informatino sur le langague de requête AQL A... Query Language sont disponibles dans le manuel DB++.

6.22.9 dbplus_chdir() : Lit/modifie le chemin courant de base

`string` **dbplus_chdir** (*`string newdir`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.10 dbplus_close() : Ferme une relation

`int` **dbplus_close** (`resource relation`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.11 dbplus_curr() : Lit la ligne courante dans une relation

`int` **dbplus_curr** (`resource relation` , `array tuple`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.12 dbplus_errcode() : Get error string for given errorcode or last error

`string` **dbplus_errcode** (`int errno`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.13 dbplus_errno() : Lit le code d'erreur généré par la dernière opération

int **dbplus_errno** (void)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_errno retourne le code d'erreur généré par la dernière opération DB++.

Voir aussi dbplus_errcode .

6.22.14 dbplus_find() : Ajoute une contrainte sur une relation

int **dbplus_find** (resource relation , array constraints , mixed tuple)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 .

6.22.15 dbplus_first() : Lit la première ligne d'une relation

int **dbplus_first** (resource relation , array tuple)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.16 dbplus_flush() : Ecrit toutes les modifications apportées à une relation

`int` **dbplus_flush** (`resource relation`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.17 dbplus_freealllocks() : Libère tous les verrous posés par le client

`int` **dbplus_freealllocks** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_freeaalllocks` libère tous les verrous posés par le client.

Voir aussi `dbplus_getlock` , `dbplus_freelock` et `dbplus_freerlocks` .

6.22.18 dbplus_freelock() : Lève un verrou en écriture sur une ligne

`int` **dbplus_freelock** (`resource relation` , `string tname`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.19 dbplus_freerlocks() : Lève tous les verrous sur les lignes d'une relation

`int` **dbplus_freerlocks** (`resource relation`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.20 dbplus_getlock() : Pose un verrou sur une ligne

`int` **dbplus_getlock** (`resource relation` , `string tname`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.21 dbplus_getunique() : Lit un identifiant unique pour une relation

`int` **dbplus_getunique** (`resource relation` , `int uniqueid`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 garantit 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.

6.22.22 dbplus_info() : Encore non implémentée.

`int` **dbplus_info** (`resource relation` , `string key` , `array`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.23 dbplus_last() : Lit la dernière ligne d'une relation

`int` **dbplus_last** (`resource relation` , `array tuple`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.24 dbplus_lockrel() : Pose un verrou en écriture sur une relation

`int` **dbplus_lockrel** (`resource relation`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.25 dbplus_next() : Lit la ligne suivante dans la relation

`int` **dbplus_next** (`resource relation` , `array tuple`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.26 dbplus_open() : Ouvre un fichier de relation

`resource` **dbplus_open** (`string name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 relation (ressource de fichier de relation, dit aussi curseur) sera retourné. Il sera alors utilisé 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` .

6.22.27 dbplus_prev() : Lit la ligne précédente dans la relation

`int` **dbplus_prev** (`resource relation` , `array tuple`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.28 dbplus_rchperm() : Modifie les droits sur la relation

`int` **dbplus_rchperm** (`resource relation` , `int mask` , `string user` , `string group`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.29 dbplus_rcreate() : Crée une nouvelle relation DB++

`resource` **dbplus_rcreate** (`string name` , `mixed domlist` , *`boolean overwrite`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 spécificateur de type et leur signification.

6.22.30 dbplus_rcrtexact() : Crée une copie exacte mais vide d'une relation

`resource` **dbplus_rcrtexact** (`string name` , `resource relation` , `boolean overwrite`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` . Une relation qui porterait déjà le nom de `name` ne sera écrasée que si `overwrite` vaut TRUE et qu'aucun autre processus n'utilise la relation.

6.22.31 dbplus_rcrtlike() : Crée une copie exacte d'une relation, sans les indexes

`resource` **dbplus_rcrtlike** (`string name` , `resource relation` , `int flag`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 n'utilise la relation.

6.22.32 dbplus_resolve() : Résoud les informations d'hôte pour une relation

`int` **dbplus_resolve** (`string relation_name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.33 dbplus_restorepos() : Pas encore implémenté.

`int` **dbplus_restorepos** (`resource relation` , `array tuple`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas encore implémenté.

6.22.34 dbplus_rkeys() : Spécifie la nouvelle clé primaire d'une relation

`resource` **dbplus_rkeys** (`resource relation` , `mixed domlist`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 domaine spécifié par `domlist` .

`domlist` peut être passé comme un nom de domnaine seul, ou comme un tableau de noms de domaines.

6.22.35 dbplus_ropen() : Ouvre un fichier de relation local

`resource` **dbplus_ropen** (`string name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.36 dbplus_rquery() : Performe une requête AQL locale

`int` **dbplus_rquery** (`string query` , `string dbpath`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.37 dbplus_rrename() : Renomme une relation

`int` **dbplus_rrename** (`resource relation` , `string name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_rrename` va changer le nom de la relation `relation` en `name` .

6.22.38 dbplus_rsecindex() : Crée un nouvel index secondaire pour une relation

`resource` **dbplus_rsecindex** (`resource relation` , `mixed domlist` , `int type`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 domnaine seul, ou comme un tableau de noms de domaines.

6.22.39 dbplus_runlink() : Détruit une relation

`int` **dbplus_runlink** (`resource relation`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.40 dbplus_rzap() : Supprime toutes les lignes d'une relation

`int` **dbplus_rzap** (`resource relation`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.22.41 dbplus_savepos() : Pas encore implémentée

`int` **dbplus_savepos** (`resource relation`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas encore implémentée.

6.22.42 dbplus_setindex() : Pas encore implémentée

`int` **dbplus_setindex** (`resource relation` , `string idx_name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas encore implémentée.

6.22.43 dbplus_setindexbynumber() : Pas encore implémentée

`int` **dbplus_setindexbynumber** (`resource relation` , `int idx_number`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas encore implémentée.

6.22.44 dbplus_sql() : Exécute une requête SQL

`resource` **dbplus_sql** (`string query` , `string server` , `string dbpath`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas encore implémentée.

6.22.45 dbplus_tcl() : Exécute du code TCL coté serveur

`int` **dbplus_tcl** (`int sid` , `string script`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 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` .

6.22.46 dbplus_tremove() : Supprime la ligne courante et retourne la nouvelle ligne courante

`int` **dbplus_tremove** (`resource relation` , `array tuple` , *`array current`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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.

6.22.47 dbplus_undo() : Pas encore implémentée

`int` **dbplus_undo** `( resource relation )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas encore implémentée.

6.22.48 dbplus_undoprepare() : Pas encore implémentée

`int` **dbplus_undoprepare** `( resource relation )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas encore implémentée.

6.22.49 dbplus_unlockrel() : Libère un verrou en écriture posé sur une relation

`int` **dbplus_unlockrel** `( resource relation )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 la fonction `dbplus_lockrel` sur la relation `relation` .

6.22.50 dbplus_unselect() : Retire une contrainte d'une relation

`int` **dbplus_unselect** `( resource relation )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 .

6.22.51 dbplus_update() : Modifie une ligne dans une relation

int **dbplus_update** (resource relation , array old , array new)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 .

6.22.52 dbplus_xlockrel() : Pose un verrou exclusif sur une relation

int **dbplus_xlockrel** (resource relation)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 .

6.22.53 dbplus_xunlockrel() : Lève un verrou exclusif posé sur une relation

int **dbplus_xunlockrel** (resource relation)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 .

6.23 Direct IO

6.23.1 Fonctions d'entrée/sortie directes

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...).

6.23.2 Installation

Pour faire fonctionner ces fonctions, vous devez configurer PHP avec l'option `--enable-dio` .

- Fonctions d'entrée/sortie directes
- Installation
- dio_close
- dio_fcntl
- dio_open
- dio_read
- dio_seek
- dio_stat
- dio_tcsetattr
- dio_truncate
- dio_write

6.23.4 dio_fcntl() : Exécute une fonction fichier de langage C

`mixed` **dio_fcntl** (`resource fd` , `int cmd` , *`mixed arg`*)
[lien librairie]

`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` attent la levée du verrou.
- F_GETLK − `dio_fcntl` retourne un tableau associatif (comme décrit ci−dessus) si un autre utilisateur empêche la poste 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.

6.23.5 dio_open() : Ovre un nouveau fichier

`resource` **dio_open** (`string filename` , `int flags` , *`int mode`*)
[lien librairie]

`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 utilisé, `dio_open` échouera si le fichier éxiste déjà.
- O_TRUNC − si le fichier existe, etqu,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

6.23.6 dio_read() : Lit des octets dans un fichier

`string` **dio_read** (`resource fd` , *`int n`*)
[lien librairie]

`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.

6.23.7 dio_seek() : Déplace le pointeur interne de fichier

`int` **dio_seek** (`resource fd` , `int pos` , `int whence`)
[lien librairie]

`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.

6.23.8 dio_stat() : Lit des informations sur le fichier

`array` **dio_stat** (`resource fd`)
[lien librairie]

`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` .

6.23.9 dio_tcsetattr() : Modifie les attributs du terminal et le taux de baud du port série

dio_tcsetattr (`resource fd` , `array options`)
[lien librairie]

`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.

6.23.10 dio_truncate() : Tronque le fichier

`bool` **dio_truncate** (`resource fd` , `int offset`)
[lien librairie]

`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 aggrandi. Dans ce dernier cas, l'extension se fait avec des octets nuls. `dio_truncate` retourne 0 en cas de succès, et sinon −1.

6.23.11 dio_write() : Ecrit des données dans le fichier

`int` **dio_write** (`resource fd` , `string data` , *`int len`*)
[lien librairie]

`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'octsts écrits dans le fichier `fd` .

6.24 Accès aux dossiers

Pour les fonctions associées telles que `dirname` , `is_dir` , `mkdir` , et `rmdir` , reportez−vous à la section sur le système de fichiers .

- chdir
- chroot
- closedir
- dir
- getcwd
- opendir
- readdir
- rewinddir

6.24.2 chroot() : Change la racine

`int` **chroot** `( string directory )`
[lien librairie]

`chroot` change la racine du script en cours, et la remplace par `directory` . `chroot` retourne FALSE s'il n'a pas pu modifier la racine, et TRUE sinon.

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.

6.24.3 closedir() : Ferme le pointeur sur le dossier.

`void` **closedir** `( resource dir_handle )`
[lien librairie]

`closedir` ferme le pointeur de dossier `dir_handle` . Le dossier devait avoir été ouvert avec `opendir` .

6.24.4 dir() : Classe dossier

`new` **dir** `( string directory )`
[lien librairie]

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();
?>
```

6.24.5 getcwd() : Retourne le dossier de travail

`string` **getcwd** (`void`)
[lien librairie]

`getcwd` retourne le nom du dossier courant.

6.24.6 opendir() : Ouvre un dossier, et récupère un pointeur dessus.

`int` **opendir** (`string path`)
[lien librairie]

`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 `opendir`

```
>>>>>>> 1.9
<?php
if ($dir = @opendir("/tmp")) {
  while($file = readdir($dir)) {
    echo "$file\n";
  }
  closedir($dir);
}
?>
```

6.24.7 readdir() : Lit une entrée du dossier.

`string` **readdir** (`resource dir_handle`)
[lien librairie]

`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);
?>
```

6.24.8 rewinddir() : Retourne à la première entrée du dossier.

`void` **rewinddir** (`resource dir_handle`)
[lien librairie]

`rewinddir` retourne à la première entrée du dossier identifié par `dir_handle` : le prochain fichier lu sera le premier.

6.25 DOM XML

Note importante : cette documentation est en cours de rédaction, et n'est pas encore finie. Elle souffre naturellement d'un manque de détails et de relecture. Soyez en prévenu. (Damien Seguy).

Ces fonctions ne sont disponibles que si PHP a été configuré avec l'option `--with-dom=[DIR]` , et utilise la librairie GNOME xml library. Vous aurez aussi besoin de la librairie libxml-2.2.7 (la version beta ne fonctionne pas). Ces fonctions ont été ajoutées en PHP 4.

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 siginifie 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` .

Ce module définit les constantes suivantes :

Constante	Valeur	Description
XML_ELEMENT_NODE	1	Le noeud est un élément
XML_ATTRIBUTE_NODE	2	Le noeud est un attribut
XML_TEXT_NODE	3	Le noeud est un texte
XML_CDATA_SECTION_NODE	4	
XML_ENTITY_REF_NODE	5	
XML_ENTITY_NODE	6	Le noeud est une entité telle que
XML_PI_NODE	7	Le noeud est une instruction
XML_COMMENT_NODE	8	Le noeud est un commentaire
XML_DOCUMENT_NODE	9	Le noeud est un document
XML_DOCUMENT_TYPE_NODE	10	
XML_DOCUMENT_FRAG_NODE	11	
XML_NOTATION_NODE	12	
XML_GLOBAL_NAMESPACE	1	
XML_LOCAL_NAMESPACE	2	

Chaque fonction de cette extension peut être utilisée de deux manières différentes. Dans un contexte procédural, il faut passer l'objet en premier argument; dans un contexte objet, vous pouvez appeler la fonction comme une méthode de cet objet. Cette documentation présente les fonctions dans leur contexte procédural. Vous pouvez connaître la méthode objet en supprimant le préfixe " `domxml_` ". Les tables suivantes listes toutes les classes, leurs attributs et leurs méthodes.

Ce module définit un ensemble de classes, qui sont listées ci-dessous (y compris leur attributs et leur méthodes).

Nom de la méthode	Nom de la fonction	Description
root	`domxml_root`	
children	`domxml_childrен`	
add_root	`domxml_add_root`	
dtd	`domxml_intdtd`	

dumpmem	domxml_dumpmem
xpath_init	xpath_init
xpath_new_context	xpath_new_context
xptr_new_context	xptr_new_context

Nom	Type	Description
doc	class DomDocument	L'objet lui−même
name	string	
url	string	
version	string	Version de XML
encoding	string	
standalone	long	1 si le fichier est complet
type	long	Une des constantes de la table ...
compression	long	1 si le fichier est compressé
charset	long	

Nom	Nom en PHP	Description
lastchild	domxml_last_child	
children	domxml_children	
parent	domxml_parent	
new_child	domxml_new_child	
get_attribute	domxml_get_attribute	
set_attribute	domxml_set_attribute	
attributes	domxml_attributes	
node	domxml_node	
set_content	domxml_set_content	

Nom	Type	Description
node	class DomNode	L'objet lui−même
type	long	
name	string	
content	string	

- DomDocument−>add_root [deprecated]
- domxml_attributes
- domxml_children
- domxml_dumpmem
- domxml_get_attribute
- domxml_new_child
- domxml_new_xmldoc
- domxml_root
- domxml_set_attribute
- xmldoc
- xmldocfile
- xmltree
- xpath_eval
- xpath_new_context

6.25.2 domxml_attributes() : Retourne les attributs d'un noeud

`array` **domxml_attributes** (`resource node`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_attributes` retourne tous les attributs du noeud `node` sous forme d'un tableau d'objets "dom attribute".

6.25.3 domxml_children() : Retourne les fils d'un noeud

`array` **domxml_children** (`object doc|node`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_children` retourne tous les fils du noeud `doc|node` , sous forme d'un tableau de noeuds.

Dans l'exemple ci−dessous, la variable `children` contiendra un tableau avec les noeuds de type XML_ELEMENT. Ce noeud est l'élément TITLE.

Lire les fils d'un noeud

```
<?php
  $doc = new_xmldoc("1.0");
  $root = $doc->add_root("HTML");
  $head = $root->new_child("HEAD", "");
  $head->new_child("TITLE", "Hier der Titel");
  $head->set_attribute("Language", "ge");
  $children = $head->children();
?>
```

6.25.4 domxml_dumpmem() : Ecrit le document XML interne dans une chaîne

`string` **domxml_dumpmem** (`resource doc`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_dumpmem crée un document XML à partir de la représentation interne. domxml_dumpmem est généralement appelée avec avoir construit un nouveau document XML, comme dans l'exemple domxml_add_root .

Voir aussi domxml_add_root .

6.25.5 domxml_get_attribute() : Retourne un attribut d'un noeud

object **domxml_get_attribute** (resource node , string name)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_get_attribute retourn l'attribut name du noeud node .

Voir aussi domxml_set_attribute .

6.25.6 domxml_new_child() : Ajoute un nouveau fils

resource **domxml_new_child** (string name , string content)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_child ajoute un nouveau fils. (NDtraducteur : cette documentation n'est pas encore finie...)

6.25.7 domxml_new_xmldoc() : Crée un document XML vide

object **domxml_new_xmldoc** (string version)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_xmldoc crée un nouveau document XML vide, et le retourne.

Voir aussi domxml_add_root .

6.25.8 domxml_root() : Retourne l'élément racine

`object` **domxml_root** (`object doc`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_root` prend en argument `doc` , un objet de la classe "Dom document", et retourn l'élément racine de ce document. Les autres noeuds qui peuvent être considérés comme racine (tels que les commentaires) sont ignorés.

L'exemple suivant retourne simplement l'élément CHAPTER et l'affiche. Les autres racines (des commentaires) ne sont pas retournés.

Lecture de l'élément principal

```
<?php
  $xmlstr = "
```

6.25.9 domxml_set_attribute() : Modifie un attribut

`object` **domxml_set_attribute** (`resource node` , `string name` , `string value`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_set_attribute` modifie l'attribut `name` du noeud `node` en lui attribuant la valeur `value` .

En partant de l'exemple proposé à la fonction `domxml_add_root` , il est simple d'ajouter un attribut à l'élément HEAD en appelant simplement `set_attribute` .

Ajouter un attribut à un élément

```
<?php
  $doc = new_xmldoc("1.0");
  $root = $doc->add_root("HTML");
  $head = $root->new_child("HEAD", "");
  $head->new_child("TITLE", "Ici, le titre");
  $head->set_attribute("Language", "fr");
  $head->new_child("TITLE", "Hier der Titel");
  $head->set_attribute("Language", "ge");
  echo $doc->dumpmem();
?>
```

6.25.10 xmldoc() : Crée un objet DOM pour un document XML.

`object` **xmldoc** (`string str`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

`xmldoc` analyse le document XML `str` et retourne un objet de classe "Dom document", avec les propriétés de "doc" (ressources), "version" (string) et "type" (long).

6.25.11 xmldocfile() : Crée un objet DOM à partir d'un fichier XML

`object` **xmldocfile** (`string filename`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

`xmldocfile` analyse le fichier XML `filename` et retourne un objet "Dom document", avec les propriétées de "doc" (ressources) et "version" (string).

6.25.12 xmltree() : Crée un arbre d'objet PHP, à partir d'un document XML.

`object` **xmltree** (`string str`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

`xmltree` analyse le document XML `str` et retourne un arbre d'objets PHP qui représente le document analysé. `xmltree` est différentes des autres fonctions, car vous ne pouvez accéder à cet arbre avec aucune des autres fonctions. Modifier cet arbre n'a pas de sens, car il n'y a pas moyen de sauver ces modifications. Cette fonction a tout de même des applications en lecture seule.

6.25.13 xpath_eval() : Evalue une expression xpath

`array` **xpath_eval** (`object xpath_context`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas de documentation encore (22/2/2201).

6.25.14 xpath_new_context() : Crée un nouveau contexte xpath

`object` **xpath_new_context** (`object dom_document`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Pas de documentation encore (22/2/2201).

6.26 .NET

- dotnet_load

6.27 Gestion des erreurs

6.27.1 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.

6.27.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.27.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.27.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
error_reporting	E_ALL & ~E_NOTICE	PHP_INI_ALL
display_errors	"1"	PHP_INI_ALL
display_startup_errors	"0"	PHP_INI_ALL
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
report_memleaks	"1"	PHP_INI_SYSTEM
track_errors	"0"	PHP_INI_ALL
html_errors	"1"	PHP_INI_ALL
docref_root	"http://www.php.net/"	PHP_INI_ALL
docref_ext	".html"	PHP_INI_ALL
error_prepend_cha”ne de caract res	NULL	PHP_INI_ALL
error_append_cha”ne de caract res	NULL	PHP_INI_ALL
error_log	NULL	PHP_INI_ALL
warn_plus_overloading	NULL	PHP_INI??

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

`error_reporting` *Array*
: 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'éxécution 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 `E_NOTICE` 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 `$arr[item]` doit toujours être écrit `$arr['item']` car PHP va considéder "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 toutes fois 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` *Array*
: Cette directive détermine si les erreurs doivent être affichées à l'écran au format HTML ou non.

`display_startup_errors` *Array*
: 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` *Array*
: 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` *Array*
: 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` *Array*
: 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` *Array*
: Ignore la source du message lors des messages répétés. Lorsuqe 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 *Array*
: 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 *Array*
: Si cette option est activée, le dernier message d'erreur sera placé dans la variable globale `$php_errormsg` .

html_errors *Array*
: Inactive 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 *Array*
: 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://landonize.it/?how=url&theme=classic&filter=Landon&u`

  La plupart du temps, vous utilisez l'option docref_root avec un slash a la fin ('/'). Mais ce n'est pas obligatoire, comme le montre le second exemple ci−dessus.

docref_ext *Array*
: Voir aussi docref_root .

Note
La valeur de docref_ext doit commencer avec un point '.'.

error_prepend_ cha”ne de caract res *Array*
: La chaîne à placer avant les messages d'erreur.

error_append_ cha”ne de caract res *Array*
: La chaîne à placer après les messages d'erreur.

error_log *Array*
: 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 *Array*
: 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 (`.`).

6.27.5 Constantes Prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

Note
Vous pouvez utiliser ces constantes dans le fichier `php.ini` mais pas hors de PHP, comme dans le fichier `httpd.conf` , où vous devez utiliser les valeurs de champs de bits.

Valeur	Constante	Description	Note
1	E_ERROR (integer)	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	E_WARNING (integer)	Les alertes sont affichées par défaut, mais n'interrompent pas l'éxécution 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.	
4	E_PARSE (integer)	Les erreurs d'analyse de doivent être générées que par l'analyseur. Elles ne sont citées ici que dans le but d'être exhaustif.	
8	E_NOTICE (integer)	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.	
16	E_CORE_ERROR (integer)	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.	PHP 4 seulement
32	E_CORE_WARNING (integer)	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.	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 trigger_error . 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 trigger_error . 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 trigger_error . 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 compbiner cesv aleurs 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 .

6.27.6 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 interessent
    // 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";
    $err .= "\t<errormsg>".$errmsg."</errormsg>\n";
    $err .= "\t<scriptname>".$filename."</scriptname>\n";
    $err .= "\t<scriptlinenum>".$linenum."</scriptlinenum>\n";

    if (in_array($errno, $user_errors))
        $err .= "\t<vartrace>".wddx_serialize_value($vars,"Variables")."</vartrace>\n";
    $err .= "</errorentry>\n\n";

    // 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);
}

$old_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";

?>
```

6.27.7 Voir aussi

Voir aussi `syslog` .

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Constantes Prédefinies
- Exemples
- Voir aussi
- error_log
- error_reporting
- restore_error_handler

- set_error_handler
- trigger_error
- user_error

6.27.9 error_reporting() : Fixe le niveau de rapport d'erreurs PHP

`int` **error_reporting** (*`int level`*)
[lien librairie]

`error_reporting` fixe le niveau de rapport d'erreur PHP et retourne l'ancienne valeur. Le niveau d'erreur peut être un champs 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

```
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

Exemples avec `error_reporting`

```
error_reporting(0);
/* Empêche tout affichage d'erreur */
error_reporting(7); // Ancienne syntaxe PHP 2/3
error_reporting(E_ERROR | E_WARNING | E_PARSE); // Nouvelle syntaxe PHP 3/4
/* Utilisation appropriée pour les erreurs courantes d'exécution */
error_reporting(15); // Ancienne syntaxe, PHP 2/3
error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE); // Nouvelle syntaxe PHP 3/4
/*  Utilisation appropriée pour les erreurs courantes de développement
 (variables non initialisées..)*/
error_reporting(63); // Ancienne syntaxe, PHP 2/3
error_reporting(E_ALL); // Nouvelle syntaxe PHP 3/4
/* rapporte toutes les erreurs PHP*/
```

6.27.10 restore_error_handler() : Réactive l'ancienne fonction de gestion des erreurs

`void` **restore_error_handler** (`void`)
[lien librairie]

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`

6.27.11 set_error_handler() : Choisi une fonction utilisateur comme gestionnaire d'erreurs

`string` **set_error_handler** (`string error_handler`)
[lien librairie]

`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 :

Traitement des erreurs avec `set_error_handler` et `trigger_error`

```
<?php
// redéfinit les constantes utilisateurs - PHP 4 seulement
define (FATAL,E_USER_ERROR);
define (ERROR,E_USER_WARNING);
define (WARNING,E_USER_NOTICE);
// Fixe le niveau de rapport d'erreur pour ce script
error_reporting  (FATAL + ERROR + WARNING);
// Fonction de traitement des erreurs
function myErrorHandler ($errno, $errstr) {
    switch ($errno) {
    case FATAL:
    echo "<B>FATAL</B> [$errno] $errstr<br>\n";
    echo "  Erreur fatale à la ligne ".__LINE__." du fichier ".__FILE__;
    echo ", PHP ".PHP_VERSION." (".PHP_OS.")<br>\n";
    echo "Aborting...<br>\n";
    exit -1;
    break;
    case ERROR:
    echo "<B>ERREUR</B> [$errno] $errstr<br>\n";
    break;
    case WARNING:
    echo "<B>ALERTE</B> [$errno] $errstr<br>\n";
    break;
    default:
    echo "Erreur inconnue de type : [$errno] $errstr<br>\n";
    break;
```

```
    }
}
// fonction qui teste la gestion d'erreur
function scale_by_log ($vect, $scale) {
    if ( !is_numeric($scale) || $scale <= 0 )
    trigger_error("log(x) pour x <= 0 est indéfini, vous avez passé: scale = $scale",
      FATAL);
    if (!is_array($vect)) {
    trigger_error("Vecteur d'entrée incorrect : un tableau de valeurs est attendu : ", ERROR);
    return null;
    }
    for ($i=0; $i<count($vect); $i++) {
    if (!is_numeric($vect[$i]))
    trigger_error("La valeur à la position $i n'est pas un nombre. On utilise 0 (zéro) à la place",
      WARNING);
    $temp[$i] = log($scale) * $vect[$i];
    }
    return $temp;
}
// Ancienne fonction de traitement des erreurs
$old_error_handler = set_error_handler("myErrorHandler");
// Génération de quelques erreurs : définition d'un tableau avec des éléments non numériques
echo "vector a\n";
$a = array(2,3,"foo",5.5,43.3,21.11);
print_r($a);
// définition d'un deuxième table à problème
echo "----\nvector b - a alerte (b = log(PI) * a)\n";
$b = scale_by_log($a, M_PI);
print_r($b);
// Ceci est un problème, on passe une chaîne à la place d'un tableau
echo "----\nvector c - une erreur\n";
$c = scale_by_log("not array",2.3);
var_dump($c);
// Ceci est critique : le tableau contient des valeurs négatives
echo "----\nvector d - fatal error\n";
$d = scale_by_log($a, -2.5);
?>
```

L'éxécution du script devrait donner ceci :

```
vector a
Array
(
    [0] => 2
    [1] => 3
    [2] => foo
    [3] => 5.5
    [4] => 43.3
    [5] => 21.11
)
----
vector b - une alerte (b = log(PI) * a)
<B>WARNING</B> [1024] La valeur à la position 2 n'est pas un nombre. On utilise 0 (zéro) à la place<
Array
(
    [0] => 2.2894597716988
    [1] => 3.4341896575482
    [2] => 0
    [3] => 6.2960143721717
    [4] => 49.566804057279
    [5] => 24.165247890281
)
----
vector c - an error
```

```
<B>ERROR</B> [512] Vecteur d'entrée incorrect : un tableau de valeur est attendu<br>
NULL
----
vector d - fatal error
<B>FATAL</B> [256] log(x) de x <= 0 est indéfini : scale = -2.5<br>
Erreur fatale à la ligne 16 du fichier trigger_error.php, PHP 4.0.1pl2 (Linux)<br>
Annulation du script....<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.

Voir aussi `error_reporting` , `restore_error_handler` , `trigger_error` et `user_error`

6.27.12 trigger_error() : Déclenche une erreur utilisateur

`void` **trigger_error** (`string error_msg` , *`int error_type`*)
[lien librairie]

`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 :

```
<?php
if (assert ($divisor == 0))
   trigger_error ("Impossible de diviser par zéro", E_USER_ERROR);
?>
```

Note
Voir aussi `set_error_handler` pour illustration.

Voir aussi `error_reporting` , `set_error_handler` , `restore_error_handler` et `user_error`

6.27.13 user_error() : Génère un message d'erreur utilisateur

`void` **user_error** (`string error_msg` , *`int error_type`*)
[lien librairie]

`user_error` est un alias de la fonction `trigger_error` .

Voir aussi `error_reporting` , `set_error_handler` , `restore_error_handler` et `trigger_error` .

6.28 FrontBase

6.28.1 Introduction

Ces fonctions vous permettent d'accéder aux serveurs SQL FrontBase. Plus de détails sur le serveur FrontBase est disponible sur le site de http://www.frontbase.com/ .

La documentation complète de FrontBase est disponible à http://www.frontbase.com/cgi−bin/WebObjects/FrontBase.woa/wa/productsPage?currentPage=Documentation .

Frontbase est supporté depuis PHP 4.0.6.

6.28.2 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/ .

6.28.3 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.

6.28.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
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

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.28.5 Types de ressources

6.28.6 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` (nombre ˆ virgule flottante)
`FBSQL_NUM` (nombre ˆ virgule flottante)
`FBSQL_BOTH` (nombre ˆ virgule flottante)
`FBSQL_LOCK_DEFERRED` (nombre ˆ virgule flottante)
`FBSQL_LOCK_OPTIMISTIC` (nombre ˆ virgule flottante)
`FBSQL_LOCK_PESSIMISTIC` (nombre ˆ virgule flottante)
`FBSQL_ISO_READ_UNCOMMITTED` (nombre ˆ virgule flottante)
`FBSQL_ISO_READ_COMMITTED` (nombre ˆ virgule flottante)
`FBSQL_ISO_REPEATABLE_READ` (nombre ˆ virgule flottante)
`FBSQL_ISO_SERIALIZABLE` (nombre ˆ virgule flottante)
`FBSQL_ISO_VERSIONED` (nombre ˆ virgule flottante)
`FBSQL_UNKNOWN` (nombre ˆ virgule flottante)
`FBSQL_STOPPED` (nombre ˆ virgule flottante)
`FBSQL_STARTING` (nombre ˆ virgule flottante)
`FBSQL_RUNNING` (nombre ˆ virgule flottante)
`FBSQL_STOPPING` (nombre ˆ virgule flottante)
`FBSQL_NOEXEC` (nombre ˆ virgule flottante)
`FBSQL_LOB_DIRECT` (nombre ˆ virgule flottante)
`FBSQL_LOB_HANDLE` (nombre ˆ virgule flottante)

- 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_db_query
- fbsql_db_status
- fbsql_drop_db
- fbsql_errno
- 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_insert_id
- fbsql_list_dbs
- fbsql_list_fields
- fbsql_list_tables
- fbsql_next_result
- fbsql_num_fields
- fbsql_num_rows
- fbsql_pconnect
- fbsql_query
- fbsql_read_blob
- fbsql_read_clob
- fbsql_result
- fbsql_rollback
- fbsql_select_db
- fbsql_set_lob_mode
- fbsql_start_db
- fbsql_stop_db
- fbsql_tablename
- fbsql_warnings

6.28.7 fbsql_affected_rows() : Lit le nombre de ligne affectées par la dernière requête

`int` **fbsql_affected_rows** (*`resource link_identifier`*)
[lien librairie]

`fbsql_affected_rows` retourne le nombre de lignes affectées par la dernière requête INSERT, UPDATE ou DELETE, effectuée avec la connexion représentée par `link_identifier` . Si ce dernier n'est pas spécifié, c'est la dernière connexion ouverte par `fbsql_connect` qui sera utilisée.

Note
Si vous utilisez les transactions, vous devez appeler `fbsql_affected_rows` après votre requête INSERT, UPDATE ou DELETE, mais pas après la validation.

Si la dernière requête DELETE ne contenait pas de clause WHERE, toutes les lignes seront effacées, mais `fbsql_affected_rows` retournera 0.

Note

Lors d'une requête UPDATE, FrontBase ne modifie pas les lignes dont les anciennes valeurs sont égales aux nouvelles. Cela fait que `fbsql_affected_rows` ne retournera pas le nombre de ligne traitées, mais le nombre de lignes affectées (modifiées) par la requête.

Si la dernière requête échoue, `fbsql_affected_rows` retourne −1.

Voir aussi `fbsql_num_rows` .

6.28.8 fbsql_autocommit() : Active ou désactive la validation automatique.

`boolean` **fbsql_autocommit** (`resource link_identifier` , *`boolean OnOff`*)
[lien librairie]

`fbsql_autocommit` retourne l'état courant de la validation automatique, pour la connexion `link_identifier` . 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 trouveé. 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`

6.28.9 fbsql_change_user() : Change le nom d'utilisateur de la session active

`resource` **fbsql_change_user** (`string user` , `string password` , *`string database`* , *`resource link_identifier`*)
[lien librairie]

`fbsql_change_user` change le nom de l'utilisateur courant sur la session active courante, ou sur `link_identifier` . 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.

6.28.10 fbsql_close() : Ferme la connexion FrontBase

`boolean` **fbsql_close** (*`resource link_identifier`*)
[lien librairie]

`fbsql_close` retourne `TRUE` en cas de succès et `FALSE` en cas d'erreur.

`fbsql_close` ferme la connexion au serveur FrontBase associé à la ressource `link_identifier` . Si `link_identifier` 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 fermé à la fin du script.

Exemple avec `fbsql_close`

```
<?php
    $link = fbsql_connect("localhost", "_SYSTEM", "secret")
        or die("Could not connect");
    print("Connecté!");
    fbsql_close($link);
?>
```

Voir aussi `fbsql_connect` et `fbsql_pconnect` .

6.28.11 fbsql_commit() : Valide une transaction

`bool` **fbsql_commit** (*`resource link_identifier`*)
[lien librairie]

`fbsql_commit` retourne TRUE en cas de succès, et FALSE sinon.

`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 étés posés sur les tables. Cette commande n'est nécessaire que si FBSQL est en mode de validation manuelle.

Voir aussi `fbsql_autocommit` et `fbsql_rollback`

6.28.12 fbsql_connect() : Ouvre une connexion à un serveur FrontBase

`resource` **fbsql_connect** (*`string hostname`* , *`string username`* , *`string password`*)
[lien librairie]

`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 valeur 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 reutilisée, et retournée.

La connexion au serveur sera fermée dès la fin du scrpit, à moins qu'elle ne soit explicitement terminée plus tôt, avec la fonction `fbsql_close` .

Exemple avec `fbsql_connect`

```
<?php
    $link = fbsql_connect("localhost", "_SYSTEM", "secret")
        or die("Could not connect");
    print("Connected successfully");
    fbsql_close($link);
?>
```

Voir aussi `fbsql_pconnect` et `fbsql_close` .

6.28.13 fbsql_create_blob() : Crée un BLOB

`string` **fbsql_create_blob** (`string blob_data` , *`resource link_identifier`*)
[lien librairie]

`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 ("Could not connect");
    $filename = "blobfile.bin";
    $fp = fopen($filename, "rb");
    $blobdata = fread($fp, filesize($filename));
    fclose($fp);
    $blobHandle = fbsql_create_blob($blobdata, $link);
    $sql = "INSERT INTO BLOB_TABLE (BLOB_COLUMN) VALUES ($blobHandle);";
    $rs = fbsql_query($sql, $link);
?>
```

Voir aussi `fbsql_create_clob` , `fbsql_read_blob` , `fbsql_read_clob` et `fbsql_set_lob_mode` .

6.28.14 fbsql_create_clob() : Crée un CLOB

`string` **fbsql_create_clob** (`string clob_data` , *`resource link_identifier`*)
[lien librairie]

`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 BLOB dans la base de données. creates a clob from

Exemple avec `fbsql_create_clob`

```
<?php
    $link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
        or die ("Could not connect");
    $filename = "clob_file.txt";
    $fp = fopen($filename, "rb");
    $clobdata = fread($fp, filesize($filename));
    fclose($fp);
    $clobHandle = fbsql_create_clob($clobdata, $link);
    $sql = "INSERT INTO CLOB_TABLE (CLOB_COLUMN) VALUES ($clobHandle);";
    $rs = fbsql_query($sql, $link);
?>
```

Voir aussi fbsql_create_blob , fbsql_read_blob , fbsql_read_clob et fbsql_set_lob_mode .

6.28.15 fbsql_create_db() : Crée une base de données

boolean **fbsql_create_db** (string database_name , *resource link_identifier*)
[lien librairie]

fbsql_create_db crée une nouvelle base de données, nommée database_name , sur le serveur repéré par la ressource link_identifier .

Exemle 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 .

6.28.16 fbsql_data_seek() : Déplace le pointeur interne de résultat

int **fbsql_data_seek** (int result_identifier , int row_number)
[lien librairie]

fbsql_data_seek retourne TRUE en cas de succès et FALSE en cas d'erreur.

fbsql_data_seek déplace le pointeur interne de ligne dans le résultat de requête result_identifier jusqu'à la ligne row_number . Le prochain appel à fbsql_fetch_row retournera cette ligne.

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 ("Query failed");
// 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);
?>
```

6.28.17 fbsql_database_password() : Modifie/lit le mot de passe dans une base FBSQL

`string` **fbsql_database_password** (`resource link_identifier` , *`string database_password`*)
[lien librairie]

`fbsql_database_password` retourne le mot de passe du serveur représenté par `link_identifier` .

`fbsql_database_password` modifie et lit le mot de passe de la base de données courante. Si le second paramètre `database_password` est fourni, `fbsql_database_password` remplacera le mot de passe courant par celui−ci. Si aucun serveur n'est précisé car le paramètre `link_identifier` n'est pas fourni, la dernière connexion ouverte sera utilisée. Si aucune connexion n'a été ouvertem PHP essaiera d'en ouvrir une en appelant la fonction `fbsql_connect` , et en utilisant la connexion qui en résultera (si elle réussit).

Voir aussi `fbsql_connect` et `fbsql_pconnect` .

6.28.18 fbsql_db_query() : Envoie une requête à la base FrontBase

`resource` **fbsql_db_query** (`string database` , `string query` , *`resource link_identifier`*)
[lien librairie]

`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_identifier` 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` .

6.28.19 fbsql_db_status() : Lit le statut courant d'une base de données

`int` **fbsql_db_status** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`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_identifier` 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 connecter 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 par 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` .

6.28.20 fbsql_drop_db() : Supprime une base de données FrontBase

`boolean` **fbsql_drop_db** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`fbsql_drop_db` retourne TRUE en cas de succès et FALSE en cas d'erreur.

`fbsql_drop_db` essaie de supprimer la base de données `database_name` , sur la connexion représentée par `link_identifier` .

6.28.21 fbsql_errno() : Retourne le code d'erreur FrontBase

`int` **fbsql_errno** (*`resource link_identifier`*)
[lien librairie]

`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.

```
<?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`

6.28.22 fbsql_error() : Retourne le message d'erreur FrontBase

`string` **fbsql_error** (*`resource  link_identifier`*)
[lien librairie]

`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.

```
<?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`

6.28.23 fbsql_fetch_array() : Lit toute une ligne de résultat dans un tableau.

`array` **fbsql_fetch_array** (`resource result` , *`int  result_type`*)
[lien librairie]

`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 à indice numérique, elle les stocke aussi sous forme de tableau associatif, dont les indices sont les noms des colonnes.

SI deux colonnes (ou plus) on le même nom, la dernière colonne sera utilisée. Pour accéder aux autres colonnes de même nom, vous devez absolument utiliser les indices numériques.

```
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 lent 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: ".$row[0]."<br>\n";
    echo "fullname: ".$row["fullname"]."<br>\n";
    echo "fullname: ".$row[1]."<br>\n";
  }
  fbsql_free_result($result);
?>
```

6.28.24 fbsql_fetch_assoc() : Lit toute une ligne de résultat dans un tableau associatif

array **fbsql_fetch_assoc** (resource result)
[lien librairie]

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) on 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 lent 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);
?>
```

6.28.25 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`*)
[lien librairie]

`fbsql_fetch_field` retourne une objet contenant les informations sur un champs, 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 champs 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 ("Query failed");
// 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` .

6.28.26 fbsql_fetch_lengths() : Lit la taille de chaque colonne d'un résultat

`array` **fbsql_fetch_lengths** (*`resource result`*)
[lien librairie]

`fbsql_fetch_lengths` retourne un tableau contenant les tailles maximales de chaque champs, 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 retourné 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` .

6.28.27 fbsql_fetch_object() : Lit une ligne de résultat sous forme d'objet

`object` **fbsql_fetch_object** (`resource result` , *`int result_type`*)
[lien librairie]

`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 pouvez alors accéder aux données qu'avec les noms des colonnes, et sous la forme de membre d'objets, et non plus avec leurs offset (les nombres ne peuvent représenter un membre d'objet).

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` .

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` .

6.28.28 fbsql_fetch_row() : Lit une ligne de résultat sous forme de tableau numérique

`array` **fbsql_fetch_row** (`resource result`)
[lien librairie]

`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 numérique. Chaque colonne est stockés dans un élément 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` .

6.28.29 fbsql_field_flags() : Lit les options associé à une colonne de résultat

`string` **fbsql_field_flags** (`resource result` , `int field_offset`)
[lien librairie]

`fbsql_field_flags` retourne les options du champs `field_offset` , dans le résultat `field_offset` . 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` .

6.28.30 fbsql_field_len() : Retourne la taille d'un champs

`int` **fbsql_field_len** (`resource result` , `int field_offset`)
[lien librairie]

`fbsql_field_len` retourne la taille du champs `field_offset` dans le résultat `result` .

6.28.31 fbsql_field_name() : Lit le nom d'un champs

`string` **fbsql_field_name** (`int result` , `int field_index`)
[lien librairie]

`fbsql_field_name` retourne le nom du champs numéro `field_index` dans le résultat `result` . `field_index` est le résultat de `fbsql_query` et `field_index` est l'offset numérique du champs.

Note
`field_index` commence à 0. e.g. L'index du troisème champs est 2, et l'index du quatrième champs est 3...

Exemple avec `fbsql_field_name`

```
<?php
// La tablea 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
```

6.28.32 fbsql_field_seek() : Déplace le pointeur de résultat

`boolean` **fbsql_field_seek** (`int result` , `int field_offset`)
[lien librairie]

`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 colonne courant.

Voir aussi `fbsql_fetch_field` .

6.28.33 fbsql_field_table() : Lit le nom de la table d'origine d'un champs

`string` **fbsql_field_table** (`resource result` , `int field_offset`)
[lien librairie]

`fbsql_field_table` retourn le nom de la table d'où est issue le champs d'offset `field_offset` . Les numéros de colonne commencent à 0.

6.28.34 fbsql_field_type() : Lit le type d'une colonne

`string` **fbsql_field_type** (`resource result` , `int field_offset`)
[lien librairie]

`fbsql_field_type` est similaire à la fonction `fbsql_field_name` . Les arguments sont identiques, mais le type du champs 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();
?>
```

6.28.35 fbsql_free_result() : Libère le résultat de la mémoire

`bool` **fbsql_free_result** (`resource result`)
[lien librairie]

`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 va consommer trop de mémoire, lorsqu'une requête retourne de très grand résultats. Toutes les ressources mémoire utilisées par le script sont de toutes manières libérées à la fin du script.

6.28.36 fbsql_insert_id() : Lit le dernier identifiant généré par une requête INSERT

`int` **fbsql_insert_id** (*`resource link_identifier`*)
[lien librairie]

`fbsql_insert_id` retourne l'identifiant généré par la colonne de type DEFAULT UNIQUE, lors de la dernière requête INSERT, avec la connexion `link_identifier` . Si `link_identifier` est omis, la dernière connexion ouverte est utilisée.

`fbsql_insert_id` retournera 0 si la dernière requête n'a pas généré de valeur dans la colonne DEFAULT UNIQUE Si vous devez sauver cette valeur pour plus tard, n'oubliez pas d'appeler `fbsql_insert_id` tout de suite après la requête qui a généré cette valeur.

Note
La valeur de la fonction FrontBase SQL " `LAST_INSERT_ID()` " retourne toujours la dernière valeur générée par DEFAULT UNIQUE et n'est jamais annulée entre les requêtes.

6.28.37 fbsql_list_dbs() : Liste les bases de données

`resource` **fbsql_list_dbs** (*`resource link_identifier`*)
[lien librairie]

`fbsql_list_dbs` retourne un résultat contenant la liste des bases de données disponibles sur le serveur *link_identifier* . 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.

6.28.38 fbsql_list_fields() : Liste les champs d'un résultat FrontBase

`resource` **fbsql_list_fields** (`string database_name` , `string table_name` , *`resource link_identifier`*)
[lien librairie]

`fbsql_list_fields` lit les informations à propos de la table `table_name` , dans la base de données `table_name` , sur la connexion `link_identifier` . 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 `$phperrmsg` . Un message d'erreur sera aussi affiché, à moins que la fonction n'ai été appelée avec l'opérateur de suppression des erreurs @.

Exemple avec `fbsql_list_fields`

```
<?php
  $link = fbsql_connect('localhost', 'myname', 'secret');
  $fields = fbsql_list_fields("database1", "table1", $link);
  $columns = fbsql_num_fields($fields);
  for ($i = 0; $i < $columns; $i++) {
    echo fbsql_field_name($fields, $i) . "\n";;
  }
?>
```

L'exemple ci−dessus va afficher :

```
field1
field2
field3
..
```

6.28.39 fbsql_list_tables() : Liste les tables dans une base de données FrontBase

`resource` **fbsql_list_tables** (`string database` , *`resource link_identifier`*)
[lien librairie]

`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.

6.28.40 fbsql_next_result() : Déplace le pointeur interne vers le résultat suivant

`bool` **fbsql_next_result** `( resource result_id )`
[lien librairie]

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 jeu 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 de résultat précédent, et préparer la lecture dans les nouvelles lignes.

`fbsql_next_result` retourne TRUE si un autre résultat est disponbile, 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);
?>
```

6.28.41 fbsql_num_fields() : Lit le nombre de champs dans un résultat

`int` **fbsql_num_fields** `( resource result )`
[lien librairie]

`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` .

6.28.42 fbsql_num_rows() : Lit le nombre de lignes dans un résultat

`int` **fbsql_num_rows** `( resource result )`
[lien librairie]

`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", "username", "password");
  fbsql_select_db("database", $link);
  $result = fbsql_query("SELECT * FROM table1;", $link);
  $num_rows = fbsql_num_rows($result);
  echo "$num_rows Rows\n";
?>
```

Voir aussi `fbsql_affected_rows` , `fbsql_connect` , `fbsql_select_db` et `fbsql_query` .

6.28.43 fbsql_pconnect() : Ouvre une connexion persistante à un serveur FrontBase

`resource` **fbsql_pconnect** (*`string hostname`* , *`string username`* , *`string password`*)
[lien librairie]

`fbsql_pconnect` retourne une ressource représentant la connection 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 permanante déjà ouverte sur cet hote, 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 (`fbsql_close` ne terminera pas une connexion persistante établie par `fbsql_pconnect`).

C'est pourquoi ce type de connexion est dite 'persistante'.

6.28.44 fbsql_query() : Exécute une requête sur un serveur FrontBase

`resource` **fbsql_query** (`string query` , *`resource link_identifier`*)
[lien librairie]

`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 ("Invalid query");
?>
```

`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 utilisez `fbsql_num_rows` pour savoir combien de lignes ont été retournée 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` .

6.28.45 fbsql_read_blob() : Lit un BLOB dans une base de données

`string` **fbsql_read_blob** (`string blob_handle` , *`resource link_identifier`*)
[lien librairie]

`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 un BLOB et/ou une colonne 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_blob` pour obtenir la valeur du BLOB.

Exemple avec `fbsql_read_blob`

```
<?php
   $link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
       or die ("Could not connect");
   $sql = "SELECT BLOB_COLUMN FROM BLOB_TABLE;";
```

```
   $rs = fbsql_query($sql, $link);
   $row_data = fbsql_fetch_row($rs);
   // $row_data[0] will now contain the blob data for teh first row
   fbsql_free_result($rs);
   $rs = fbsql_query($sql, $link);
   fbsql_set_lob_mode($rs, FBSQL_LOB_HANDLE);
   $row_data = fbsql_fetch_row($rs);
   // $row_data[0] will now contain a handle to the BLOB data in the first row
   $blob_data = fbsql_read_blob($row_data[0]);
   fbsql_free_result($rs);
?>
```

Voir aussi `fbsql_create_blob` , `fbsql_read_blob` , `fbsql_read_clob` et `fbsql_set_lob_mode` .

6.28.46 fbsql_read_clob() : Lit un CLOB dans une base de données

`string` **fbsql_read_clob** (`string clob_handle` , *`resource link_identifier`*)
[lien librairie]

`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 un CLOB et/ou une colonne 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_clob` pour obtenir la valeur du CLOB.

Exemple avec `fbsql_read_blob`

```
<?php
   $link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
       or die ("Could not connect");
   $sql = "SELECT CLOB_COLUMN FROM CLOB_TABLE;";
   $rs = fbsql_query($sql, $link);
   $row_data = fbsql_fetch_row($rs);
   // $row_data[0] will now contain the clob data for teh first row
   fbsql_free_result($rs);
   $rs = fbsql_query($sql, $link);
   fbsql_set_lob_mode($rs, FBSQL_LOB_HANDLE);
   $row_data = fbsql_fetch_row($rs);
   // $row_data[0] will now contain a handle to the CLOB data in the first row
   $clob_data = fbsql_read_clob($row_data[0]);
   fbsql_free_result($rs);
?>
```

Voir aussi `fbsql_create_blob` , `fbsql_read_blob` , `fbsql_read_clob` et `fbsql_set_lob_mode` .

6.28.47 fbsql_result() : Lit des données dans un résultat

`mixed` **fbsql_result** (`resource result` , `int row` , *`mixed field`*)
[lien librairie]

`fbsql_result` lit le contenu du champs `field` , dans la ligne `row` , du résultat `result` . L'argument `field` peut être l'offset du champs, 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` .

6.28.48 fbsql_rollback() : Annule une transaction

`bool` **fbsql_rollback** (*`resource link_identifier`*)
[lien librairie]

`fbsql_rollback` retourne `TRUE` en cas de succès et `FALSE` sinon.

`fbsql_rollback` termine la transaction courante en annulant toutes les requêtes émises depuis la dernière validation. Cette commande n'est utile que si FrontBase est en mode de validation manuelle.

Voir aussi `fbsql_autocommit` et `fbsql_commit`

6.28.49 fbsql_select_db() : Sélectionne une base de données FrontBase

`resource` **fbsql_select_db** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`fbsql_select_db` retourne `TRUE` en cas de succès et `FALSE` en cas d'erreur.

`fbsql_select_db` remplace la base de données active courante par `database_name` , sur la connexion ouverte et représentée par `link_identifier` . Si `link_identifier` est omis, la dernière connexion ouverte sera utilisée. Si aucune connexion n'a été ouverte, `fbsql_select_db` essaiera de se connecter en appelant `fbsql_connect` sans argument.

Le client contacte FBExec pour connaître le numéro de port à utiliser pour la connexion à la base de données. Si le nom de la base est un numéro, le système l'utilisera comme numéro de port, et ne le demandera pas à FBExec. Le serveur Frontbase peut être démarré avec la commande : `FRontBase −FBExec=No −port=<port number> <database name>` .

Tous les prochains appel à `fbsql_query` se feront dans la base `database_name` .

Voir aussi `fbsql_connect` , `fbsql_pconnect` et `fbsql_query` .

6.28.50 fbsql_set_lob_mode() : Modifie le mode de lecture des LOB

`bool` **fbsql_set_lob_mode** (`resource result` , `string database_name`)
[lien librairie]

`fbsql_set_lob_mode` retourne `TRUE` en cas de succès et `FALSE` sinon.

`fbsql_set_lob_mode` modifie le mode de lecture des LOB. Lorsque des données BLOB ou CLOB sont stockées dans une base de données FrontBase, elles peuvent l'être de manière directe ou indirecte. Si les données sont stockées directement, elles seront automatiquement lues, quelque soit le mode choisi. Si un LOB fait moins de 512 octets, il sera aussi stocké directement.

- FBSQL_LOB_DIRECT − les données LOB sont lues directements. Lorsque les données sont lues dans la base avec `fbsql_fetch_row` ou d'autres fonctions de lectures, toutes les colonnes de type CLOB et BLOB seront retournées comme des colonnes habituelles. C'est le comportement par défaut de FrontBase.
- FBSQL_LOB_HANDLE − Les données LOB sont lues sous forme d'identifiant. Lorsque les données sont lues dans la base avec `fbsql_fetch_row` ou d'autres fonctions de lectures, toutes les colonnes de type CLOB et BLOB seront retournées sous la forme d'un identifiant. Ce identifiant sera une chaîne de 27 caractères de long, formatée comme ceci : `"@'00000000000000000000000'"` .

Voir aussi `fbsql_create_blob` , `fbsql_create_clob` , `fbsql_read_blob` et `fbsql_read_clob` .

6.28.51 fbsql_start_db() : Démarre une base de données

`bool` **fbsql_start_db** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`fbsql_start_db` retourne `TRUE` en cas de succès et `FALSE` sinon.

`fbsql_start_db` démarre la base de données `database_name` sur la connexion `link_identifier` .

Voir aussi `fbsql_db_status` et `fbsql_stop_db` .

6.28.52 fbsql_stop_db() : Stoppe une base de données

`bool` **fbsql_stop_db** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`fbsql_stop_db` retourne `TRUE` en cas de succès et `FALSE` sinon.

`fbsql_stop_db` stoppe la base de données `database_name` sur la connexion `link_identifier` .

Voir aussi `fbsql_db_status` et `fbsql_start_db` .

6.28.53 fbsql_tablename() : Lit le nom de la table d'un champs

`string` **fbsql_tablename** (`resource result` , `int i`)
[lien librairie]

`fbsql_tablename` retourne le nom de la table d'origine du champs `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++;
  }
?>
```

6.28.54 fbsql_warnings() : Active ou désactive les alertes FrontBase

`boolean` **fbsql_warnings** (*`boolean OnOff`*)
[lien librairie]

`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` .

6.29 FilePro

6.29.1 filepro_fieldcount() : Retourne le nombre de champs dans une base filePro.

`int` **filepro_fieldcount** (`void`)
[lien librairie]

`filepro_fieldcount` retourne le nombre de champs (ou colonnes) d'une base filePro.

Voir aussi `filepro` .

- filepro_fieldcount
- filepro_fieldname
- filepro_fieldtype
- filepro_fieldwidth
- filepro_retrieve
- filepro_rowcount
- filepro

6.29.3 filepro_fieldtype() : Retourne le type d'un champs.

`string` **filepro_fieldtype** (`int field_number`)
[lien librairie]

`filepro_fieldtype` retourne le type du champs d'index `field_number` .

6.29.4 filepro_fieldwidth() : Retourne la taille d'un champs.

`int` **filepro_fieldwidth** (`int field_number`)
[lien librairie]

`filepro_fieldwidth` retourne la taille du champs d'index `field_number` .

6.29.5 filepro_retrieve() : Retourne la valeur d'un champs.

`string` **filepro_retrieve** (`int row_number` , `int field_number`)
[lien librairie]

`filepro_retrieve` retourne la valeur du champs 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..

6.29.6 filepro_rowcount() : Retourne le nombre de champs dans une base filePro.

`int` **filepro_rowcount** (`void`)
[lien librairie]

`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..

6.29.7 filepro() : Lit et vérifie un fichier.

`boolean` **filepro** (`string directory`)
[lien librairie]

`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..

6.30 Système de fichiers

6.30.1 Introduction

6.30.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.30.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.30.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
allow_url_fopen	"1"	PHP_INI_ALL
user_agent	NULL	PHP_INI_ALL
default_socket_timeout	"60"	PHP_INI_ALL
from	NULL	??
auto_detect_line_endings	"Off"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les 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 protocole 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 compliation, avec l'option `−−disable−url−fopen−wrapper` .
Attention
Sous Windows, dans les versions antérieure à la version 4.3, les fonctions suiavntes ne supportent pas l'accès aux fichiers distants : `include` , `include_once` , `require` , `require_once` et les fonctions imagecreatefromXXX de l'extension .

`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 un légère pénalité en temps de traitement, mais aussi parce que ceux qui utilisent les retours chariots comme séparateurs auront des soucis de compatibilité.

Note
Cette option de configuration a été introduite en PHP 4.3.

6.30.5 Types de ressources

6.30.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

6.30.7 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 .

- 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
- fileatime
- filectime
- filegroup
- fileinode
- filemtime
- fileowner
- fileperms
- filesize
- filetype
- flock
- fnmatch
- fopen
- fpassthru
- 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
- stream_get_meta_data

- stream_register_wrapper
- stream_set_blocking
- stream_set_timeout
- stream_set_write_buffer
- symlink
- tempnam
- tmpfile
- touch
- umask
- unlink

6.30.9 chgrp() : Change le groupe possesseur du fichier.

`int` **chgrp** (`string filename` , `mixed group`)
[lien librairie]

`chgrp` essaie de changer le groupe propriétaire du fichier. 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.

`chgrp` renvoie TRUE en cas de succès, sinon renvoie FALSE .

Voir aussi `chown` et `chmod` .

Note
`chgrp` ne fonctionne pas sous Windows.

6.30.10 chmod() : Change le mode du fichier.

`int` **chmod** (`string filename` , `int mode`)
[lien librairie]

`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`):

```
<?php
  chmod( "/somedir/somefile", 755 );
// notation décimale; probablement
<TT>FALSE</TT>

  chmod( "/somedir/somefile", 0755 );
// notation octale; valeur du mode correcte
?>
```

`chmod` renvoie TRUE en cas de succès, FALSE sinon.

Voir aussi `chown` et `chgrp` .

Note
chmod ne fonctionne pas sous Windows.

6.30.11 chown() : Change le groupe propriétaire du fichier.

int **chown** (string filename , mixed user)
[lien librairie]

chown change le groupe propriétaire du fichier. Seul le super−utilisateur (root) peut changer le propriétaire arbitrairement d'un fichier.

chown renvoie TRUE en cas de succès, " FALSE " sinon.

Note
Sous Windows, chown ne fait rien et retourne TRUE .

Voir aussi chown et chmod .

Note
chown est inopérante sous Windows.

6.30.12 clearstatcache() : Efface le cache de stat

void **clearstatcache** (void)
[lien librairie]

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 .

6.30.13 copy() : Copie un fichier.

int **copy** (string source , string dest)
[lien librairie]

copy fait une copie du fichier. Elle renvoie TRUE en cas de succès, FALSE sinon.

Exemple avec copy

```
if ( !copy($file, $file.'.bak') ) {
    print("La copie du fichier $file n'a pas réussi...<br>\n");
}
```

Voir aussi rename .

6.30.14 delete() : Effacer

void **delete** (string file)
[lien librairie]

Ceci est une fausse entrée du manuel pour ceux qui recherchent en fait la fonction unlink ou unset .

Voir aussi unlink pour effacer des fichiers et unset pour effacer des variables.

6.30.15 dirname() : Renvoie le nom du dossier.

string **dirname** (string path)
[lien librairie]

Si path contient le chemin d'un fichier ou dossier, dirname retournera le nom du dossier qui le contient.

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 contient "/etc"
?>
```

Voir aussi basename .

6.30.16 disk_free_space() : Renvoie l'espace disque disponible dans le répertoire.

float **disk_free_space** (string directory)
[lien librairie]

disk_free_space retournera le nombre d'octets disponibles sur le disque correspondant contenant le dossier directory .

Exemple avec disk_free_space

```
<?php
  $df = disk_free_space("/");
  // $df contient le nombre d'octets libres sur "/"
?>
```

6.30.17 disk_total_space() : Retourne la taille d'un dossier

`float` **disk_total_space** (`string directory`)
[lien librairie]

`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

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..

6.30.18 diskfreespace() : Alias de `disk_free_space`

`float` **diskfreespace** (`string directory`)
[lien librairie]

`diskfreespace` est l'ancien nom de la fonction `disk_free_space` . Elle est obsolète, et il est vivement recommandé d'utiliser `disk_free_space` .

6.30.19 fclose() : Ferme un fichier.

`bool` **fclose** (`resource fp`)
[lien librairie]

`fclose` ferme le fichier `fp` .

`fclose` retourne TRUE en cas de succès, et FALSE en cas d'échec.

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par `fopen` ou `fsockopen` .

6.30.20 feof() : Teste la fin du fichier.

`int` **feof** (`resource fp`)
[lien librairie]

`feof` retourne TRUE si le pointeur `fp` 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` .

6.30.21 fflush() : Envoi tout le contenu généré dans un fichier

`int` **fflush** (`resource fp`)
[lien librairie]

`fflush` force l'écriture de toutes les données bufferisées dans le fichier désigné par `fp` . `fflush` retourne TRUE en cas de succès, et FALSE sinon.

`fp` est un pointeur de fichier ouvert avec `fopen` , `popen` , ou `fsockopen` .

6.30.22 fgetc() : Renvoie le caractère que pointe le pointeur du fichier.

`string` **fgetc** (`resource fp`)
[lien librairie]

`fgetc` retourne une chaîne contenant un seul caractère, lu depuis le fichier pointé par `fp` . `fgetc` retourne FALSE à la fin du fichier (tout comme `feof`).

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par `fopen` , `popen` , ou `fsockopen` .

Voir aussi `fread` , `fopen` , `popen` , `fsockopen` et `fgets` .

6.30.23 fgetcsv() : Renvoie la ligne courante et cherche les champs CSV

`array` **fgetcsv** (`resource fp` , `int length` , *`string delimiter`*)
[lien librairie]

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.

`fp` 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 dans le tableau comme une chaîne vide, et ne sera pas traitée comme une erreur.

Exemple 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>";
    $row++;
    for ($c=0; $c<$num; $c++) {
```

```
        print $data[$c] . "<br>";
    }
  }
  fclose ($fp);
?>
```

6.30.24 fgets() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier.

`string` **fgets** (`int fp` , `int length`)
[lien librairie]

`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.

`fp` 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);
?>
```

Voir aussi `fread` , `fopen` , `popen` , `fgetc` , `fsockopen` et `socket_set_timeout` .

6.30.25 fgetss() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier et élimine les balises HTML

`string` **fgetss** (`int fp` , `int length` , *`string allowable_tags`*)
[lien librairie]

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 dans PHP 3.0.13, PHP 4B3.

Voir aussi fgets , fopen , fsockopen , popen et strip_tags .

6.30.26 file_exists() : Vérifie si un fichier existe.

int **file_exists** (string filename)
[lien librairie]

file_exists retourne TRUE si le fichier filename existe, et FALSE sinon.

file_exists ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Le résultat de file_exists est mis en cache. Reportez−vous à clearstatcache pour plus de détails.

6.30.27 file_get_contents() : Lit tout un fichier dans une chaîne

string **file_get_contents** (string filename , *int use_include_path*)
[lien librairie]

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

Vous pouvez utiliser une URL comme nom de fichier avec cette fonction, si l'option des "fopen wrappers" a été activée. Voyez fopen pour plus de détails.

Voir aussi fgets , file , fread , include et readfile .

6.30.28 file() : Lit le fichier et renvoie le résultat dans un tableau.

array **file** (string filename , *int use_include_path*)
[lien librairie]

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.

Vous pouvez utiliser l'option use_include_path : en la mettant à "1", vous rechercherez aussi dans le dossier include_path .

```
<?php
```

```
// Lire une page web dans un tableau, et l'afficher
  $fcontents = file( 'http://www.php.net' );
  while ( list( $numero_ligne, $ligne ) = each( $fcontents ) ) {
   echo "<B>Ligne $numero_ligne:</B> ".htmlspecialchars( $ligne ) . "<br>\n";
  }
// lire une page web dans une chaîne
  $fcontents = join( '', file( 'http://www.php.net' ) );
?>
```

Voir aussi `readfile` , `fopen` , `fsockopen` et `popen` .

6.30.29 fileatime() : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois.

`int` **fileatime** (`string filename`)
[lien librairie]

`fileatime` renvoie la date à laquelle le fichier a été accédé pour la dernière fois, ou FALSE en cas d'erreur.

`fileatime` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Le résultat de `fileatime` est mis en cache. Reportez−vous à `clearstatcache` pour plus de détails.

6.30.30 filectime() : Renvoie l'heure à laquelle l'inode a été accédé pour la dernière fois.

`int` **filectime** (`string filename`)
[lien librairie]

`filectime` renvoie l'heure à laquelle l'inode `filename` a été accédé pour la dernière fois, ou FALSE en cas d'erreur.

`filectime` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Note: Sur la plupart des serveurs UNIX, un fichier est considéré comme modifié si les données de son inode sont modifiées. C'est−à−dire lorsque les permissions (utilisateur, groupe ou autre) ont été modifiées. Voyez aussi `filemtime` (que vous pourrez utiliser lorsque vous créerez des indications telles que "Dernière modification : " sur les pages web) et `fileatime` .

Notez aussi que sur certains systèmes UNIX, le ctime d'un fichier texte est considéré comme sa date de création. Cela est faux! Il n'y a pas de date de création de fichier sous la plupart des systèmes UNIX.

Le résultat de `filectime` est mis en cache. Reportez−vous à `clearstatcache` pour plus de détails.

6.30.31 filegroup() : Lire le nom du groupe

`int` **filegroup** (`string filename`)
[lien librairie]

`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.

`filegroup` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Le résultat de `filegroup` est mis en cache. Reportez−vous à `clearstatcache` pour plus de détails.

Note
`filegroup` est inopérante sous Windows.

6.30.32 fileinode() : Renvoie le numéro d'inode du fichier.

`int` **fileinode** (`string filename`)
[lien librairie]

`fileinode` renvoie le numéro d'inode du fichier `filename` , ou FALSE en cas d'erreur.

Le résultat de `fileinode` est mis en cache. Reportez−vous à `clearstatcache` pour plus de détails.

`fileinode` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Note
`fileinode` est inopérante sous Windows.

6.30.33 filemtime() : Renvoie la date de dernière modification du fichier.

`int` **filemtime** (`string filename`)
[lien librairie]

`filemtime` renvoie la date de dernière modification du fichier `filename` , ou FALSE en cas d'erreur.

Le résultat de `filemtime` est mis en cache. Reportez−vous à `clearstatcache` pour plus de détails.

`filemtime` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

`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.

6.30.34 fileowner() : Renvoie le nom du propriétaire du fichier.

`int` **fileowner** (`string filename`)
[lien librairie]

`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.

`fileowner` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Le résultat de `fileowner` est mis en cache. Reportez−vous à `clearstatcache` pour plus de détails.

Note
`fileowner` est inopérante sous Windows.

6.30.35 fileperms() : Renvoie les permissions affectées au fichier.

`int` **fileperms** (`string filename`)
[lien librairie]

`fileperms` renvoie les permissions affectées au fichier `filename` , ou FALSE en cas d'erreur.

`fileperms` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Le résultat de `fileperms` est mis en cache. Reportez−vous à `clearstatcache` pour plus de détails.

6.30.36 filesize() : Renvoie la taille du fichier.

`int` **filesize** (`string filename`)
[lien librairie]

`filesize` renvoie la taille du fichier `filename` , ou FALSE en cas d'erreur.

`filesize` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Le résultat de `filesize` est mis en cache. Reportez−vous à `clearstatcache` pour plus de détails.

6.30.37 filetype() : Retourne le type de fichier

`string` **filetype** (`string filename`)
[lien librairie]

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.

Le résultat de filetype est mis en cache. Reportez−vous à clearstatcache pour plus de détails.

6.30.38 flock() : Verrouille le fichier.

boolean **flock** (int fp , int operation)
[lien librairie]

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 fp qui doit avoir été ouvert au préalable. operation est une des valeurs suivantes :

- Acquisition d'un verrou : operation = 1.
- Acquisition d'un verrou exclusif (écriture), operation = 2.
- Libération d'un verrou (partagé ou exclusif), operation = 3.
- Si vous voulez que flock ne se bloque pas durant le verrouillage, ajoutez 4 à operation .

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).

flock retourne TRUE en cas de succès, et FALSE sinon. (le verrou n'a pas pu être obtenu).

Attention
Sur la plupart des OS, flock est implémenté au niveau processus. Lors de l'utilisation des API d'un serveur multi−thread, comme ISAPI, vous ne pouvez pas vous fier à flock pour protéger vos fichiers contre des accès concurrents du même serveur.

6.30.39 fnmatch() : Repère un fichier à partir d'un masque

array **fnmatch** (string pattern , string string , *int flags*)
[lien librairie]

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` , `preg_match` et le manuel Unix de `fnmatch(3)` pour les options (tant qu'elles ne seront pas documentées ici).

6.30.40 fopen() : Ouverture d'un fichier ou d'une URL.

`int` **fopen** (`string filename` , `string mode` , *`int use_include_path`*)
[lien librairie]

Si `filename` commence par "http://" (insensible à la casse), une connexion HTTP 1.x est ouverte avec le serveur spécifié, et un pointeur sur la réponse fournie est retourné. Une en−tête ' `Host:` ' est envoyé avec la requête, afin de gérer les virtual hosts basés sur les noms.

Notez que le pointeur de fichier retourné représente le corps de la réponse, et qu'il n'est pas possible d'accéder aux en−têtes HTTP avec cette fonction.

Les versions antérieures à PHP 4.0.6, ne gère pas les redirections automatiques, ce qui oblige à ajouter les slash finaux "/" pour indiquer un dossier.

Si `filename` commence par "ftp://" (insensible à la casse), une connexion FTP est ouverte avec le serveur spécifié, et un pointeur sur la réponse fournie est retourné. Si le serveur ne supporte par le mode FTP passif, `fopen` échouera. Vous pouvez ouvrir des fichiers en lecture seulement, ou en écriture seulement (le full duplex n'est pas supporté).

Si `filename` commence par "php://stdin", "php://stdout", ou "php://stderr", le flot correspondant sera ouvert. (Cela a été introduit en PHP 3.0.13; dans les anciennes versions, les fichiers "/dev/stdin" ou "/dev/fd/0" devaient être utilisés pour accéder à ces flots).

Si `filename` commence par n'importe quoi d'autre, PHP tentera de lire ce fichier dans le système local, et un pointeur sur le fichier ouvert sera retourné.

Si l'ouverture échoue, `fopen` retourne FALSE .

`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.

Exemple avec `fopen`

```
<?php
  $fp = fopen("/home/rasmus/file.txt", "r");
  $fp = fopen("http://www.php.net/", "r");
  $fp = fopen("ftp://user:password&#64;example.com/", "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.

```
<?php
  $fp = fopen("c:\\data\\info.txt", "r");
?>
```

Voir aussi `fclose` , `fsockopen` , `socket_set_timeout` et `popen` .

6.30.41 fpassthru() : Affiche la partie du fichier située après le pointeur du fichier.

`int` **fpassthru** (`int fp`)
[lien librairie]

`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 .

Le pointeur de fichier doit être valide, et doit avoir été correctement ouvert par `fopen` , `popen` , ou `fsockopen` . Après lecture, `fpassthru` va fermer le fichier (le pointeur `fp` sera alors invalide).

Si vous voulez simplement afficher le contenu d'un fichier, il suffit d'utiliser `readfile` , ce qui épargnera l'appel à `fopen` .

Voir aussi `readfile` , `fopen` , `popen` et `fsockopen` .

6.30.42 fputs() : Ecrit dans un fichier.

`int` **fputs** (`int fp` , `string str` , *`int length`*)
[lien librairie]

`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.

6.30.43 fread() : Lecture du fichier en mode binaire.

`string` **fread** (`int fp` , `int length`)
[lien librairie]

`fread` lit jusqu'à `length` octets dans le fichier referencé par `fp` . 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).

```php
<?php
// Lit un fichier, et le place dans une chaîne
  $filename = "/usr/local/quelquechose.txt";
  $fd = fopen($filename, "r");
  $contents = fread($fd, filesize ($filename));
  fclose($fd);
?>
```

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

```php
<?php
  $filename = "c:\\fichiers\\uneimage.gif";
  $fd = fopen($filename, "rb");
  $contents = fread($fd, filesize ($filename));
  fclose($fd);
?>
```

Voir aussi `fwrite` , `fopen` , `fsockopen` , `popen` , `fgets` , `fgetss` , `file` et `fpassthru` .

6.30.44 fscanf() : Analyse un fichier en fonction d'un format

`mixed` **fscanf** (`int handle` , `string format` , *`string var1`*)
[lien librairie]

`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.

Exemple `fscanf`

```php
<?php
  $fp = fopen ("users.txt","r");
```

```
  while ($userinfo = fscanf ($fp, "%s\t%s\t%s\n")) {
    list ($name, $profession, $countrycode) = $userinfo;
    //... traitement des données
  }
  fclose($fp);
?>
```

users.txt

```
janus  argonaute        gr
rodin  sculpteur        fr
sam    oncle            us
leonard   inventeur     it
```

Voir aussi `fread` , `fgets` , `fgetss` , `sscanf` , `printf` et `sprintf` .

6.30.45 fseek() : Modifie le pointeur de fichier.

`int` **fseek** (`int  fp` , `int  offset`)
[lien librairie]

`fseek` modifie le curseur de position dans le fichier `fp` . 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 TRUE 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.

Voir aussi `ftell` et `rewind` .

6.30.46 fstat() : Lit les informations sur un fichier à partir d'un pointeur de fichier

`array` **fstat** (`int  fp`)
[lien librairie]

`fstat` rassemble les informations sur le fichier dont on connaît le pointeur `fp` . `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 :

- 1 : volume
- 2 : inode
- 3 : mode de protection du inode
- 4 : nombre de liens
- 5 : id de l'utilisateur propriétaire
- 6 : id du groupe propriétaire
- 7 : type du volume de l'inode *
- 8 : taille en octets
- 9 : date du dernier accès
- 10 : date de la dernière modification
- 11 : date du dernier changement
- 12 : taille de bloc du système pour les entrées/sorties(*)
- 13 : Nombre de blocs alloués

* − uniquement sur les systèmes qui supportent le type st_blksize. Les autres systèmes (i.e. Windows) retournent −1.

Les résultats de `fstat` sont mis en cache. Reportez−vous à la fonction `clearstatcache` pour plus de détails.

6.30.47 ftell() : Renvoie la position du pointeur du fichier.

`int` **ftell** (`int fp`)
[lien librairie]

`ftell` retourne la position courante du pointeur dans le fichier repéré par le pointeur `fp` , 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` .

6.30.48 ftruncate() : Tronque un fichier.

`int` **ftruncate** (`int fp` , `int size`)
[lien librairie]

`ftruncate` prend le pointeur de fichier `fp` et le tronque à la taille de `size` . `ftruncate` retourne `TRUE` en cas de succès, et `FALSE` sinon.

6.30.49 fwrite() : Ecriture du fichier en mode binaire.

`int` **fwrite** (`int fp` , `string string` , *`int length`*)
[lien librairie]

`fwrite` écrit le contenu de la chaîne `string` dans le fichier pointé par `fp` . Si la longueur `length` est fournie, l'écriture s'arrêtera après `length` octets, ou à la fin de la chaîne (le premier des deux).

Notez que si `length` est fourni, alors l'option de configuration magic_quotes_runtime sera ignorée, et les slash seront conservés.

Note
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.

Voir aussi `fread` , `fopen` , `fsockopen` , `popen` et `fputs` .

6.30.50 glob() : Recheche des chemins qui vérifie un masque

`array` **glob** (`string pattern` , *`int flags`*)
[lien librairie]

`glob` recherche tous les chemins qui vérifient le masque `pattern` , en suivant les règles du shell. Aucun remlacement 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";
  }
?>
```

Voir aussi `opendir` , `readdir` , `closedir` , `fnmatch` et le manuel Unix de `fnmatch(3)` pour les options (tant qu'elles ne seront pas documentées ici).

6.30.51 is_dir() : Indique si le nom de fichier est un dossier.

`boolean` **is_dir** (`string filename`)
[lien librairie]

`is_dir` retourne `TRUE` si `filename` existe et est un dossier.

Le résultat de `is_dir` est mis en cache. Voir la fonction `clearstatcache` pour plus de détails.

`is_dir` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Voir aussi `is_file` et `is_link` .

6.30.52 is_executable() : Indique si le fichier est exécutable.

`boolean` **is_executable** (`string filename`)

[lien librairie]

`is_executable` retourne `TRUE` si `filename` existe et est exécutable.

Le résultat de `is_executable` est mis en cache. Voir la fonction `clearstatcache` pour plus de détails.

`is_executable` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Voir aussi `is_file` et `is_link` .

6.30.53 is_file() : Indique si le fichier est un véritable fichier.

`boolean` **is_file** (`string filename`)
[lien librairie]

`is_file` retourne `TRUE` si `filename` existe et est un fichier (et pas un dossier).

Le résultat de `is_file` est mis en cache. Voir la fonction `clearstatcache` pour plus de détails.

`is_file` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Voir aussi `is_dir` et `is_link` .

6.30.54 is_link() : Indique si le fichier est un lien symbolique.

`boolean` **is_link** (`string filename`)
[lien librairie]

`is_link` retourne `TRUE` si `filename` existe et est un lien symbolique.

Le résultat de `is_link` est mis en cache. Voir la fonction `clearstatcache` pour plus de détails.

`is_link` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Voir aussi `is_dir` et `is_file` .

Note
`is_link` est inopérante sous Windows.

6.30.55 is_readable() : Indique un fichier est autorisé en lecture

`boolean` **is_readable** (`string filename`)
[lien librairie]

`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).

Le résultat de `is_readable` est mis en cache. Voir la fonction `clearstatcache` pour plus de détails.

`is_readable` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Voir aussi `is_writable` .

6.30.56 is_uploaded_file() : Indique si le fichier a été téléchargé par HTTP POST

`boolean` **is_uploaded_file** (`string filename`)
[lien librairie]

`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 révélent leur contenu à l'utilisateur, ou même aux utilisateurs du même système.

Voir aussi `move_uploaded_file` , et la section Chargement de fichier pour un exemple simple.

6.30.57 is_writable() : Indique si un fichier est autorisé en écriture.

`boolean` **is_writable** (`string filename`)
[lien librairie]

`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).

`is_writable` ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

Le résultat de `is_writable` est mis en cache. Voir la fonction `clearstatcache` pour plus de détails.

Voir aussi `is_readable` .

6.30.58 is_writeable() : Indique si un fichier est autorisé en écriture.

`boolean` **is_writeable** (`string filename`)
[lien librairie]

is_writeable est un alias de is_writable .

6.30.59 link() : Crée un lien.

int **link** (string target , string link)
[lien librairie]

link crée un lien.

Voir aussi symlink pour créer des liens symboliques et readlink avec linkinfo .

Note
link est inopérante sous Windows.

6.30.60 linkinfo() : Renvoie les informations à propos d'un lien.

int **linkinfo** (string path)
[lien librairie]

linkinfo renvoie les informations à propos d'un lien, c'est−à−dire le champs 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 .

Note
linkinfo est inopérante sous Windows.

6.30.61 lstat() : Renvoie les informations à propos d'un fichier ou d'un lien symbolique.

array **lstat** (string filename)
[lien librairie]

lstat est identique à stat mais elle accepte aussi un lien symbolique comme argument.

lstat retourne un tableau avec les éléments suivants :

- 0 : volume
- 1 : inode
- 2 : droits d'accès au fichier (mode de protection du inode). A convertir en octal. Voir aussi fileperms .
- 3 : nombre de liens
- 4 : id de l'utilisateur propriétaire
- 5 : id du groupe propriétaire
- 6 : type du volume de l'inode *
- 7 : taille en octets
- 8 : date du dernier accès
- 9 : date de la dernière modification

- 10 : date du dernier changement
- 11 : taille de bloc du système pour les entrées/sorties *
- 12 : nombre de blocs alloués

* − uniquement sur les systèmes qui supportent le type st_blksize. Les autres systèmes (i.e. Windows) retournent −1.

Les résultats de `lstat` sont mis en cache. Reportez−vous à la fonction `clearstatcache` pour plus de détails.

6.30.62 mkdir() : Crée un dossier.

`int` **mkdir** (`string pathname` , `int mode`)
[lien librairie]

`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` .

```
<?php
  mkdir ("/chemin/de/mon/dossier", 0700);
?>
```

`mkdir` retourne `TRUE` en cas de succès, et `FALSE` en cas d'échec.

Voir aussi `rmdir` .

6.30.63 move_uploaded_file() : Déplace un fichier téléchargé.

`boolean` **move_uploaded_file** (`string filename` , `string destination`)
[lien librairie]

`move_uploaded_file` est disponible à partir des versions PHP 3.0.16 et 4.0.2.

`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.

Voir aussi `move_uploaded_file` et la section Chargement de fichier pour un exemple simple.

6.30.64 parse_ini_file() : Traite un fichier de configuration

`array` **parse_ini_file** (`string filename` , *`bool process_sections`*)
[lien librairie]

`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 a voir avec le fichier `php.ini` . Ce dernier a déjà était traité lorsque vous commencez a 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` .

Contenu de exemple.ini

```
; Ceco est un fichier de configuration
; Les commentaires commencent par ';', comme dans php.ini
[premiere_section]
un = 1
cinq = 5
[seconde_section]
chemin = /usr/local/bin
```

Exemple avec `parse_ini_file`

```
<?php
// Traitement sans les sections
$ini_array = parse_ini_file("exemple.ini");
print_r($ini_array);
// Traitement avec les sections
$ini_array = parse_ini_file("sample.ini", TRUE);
print_r($ini_array);
?>
```

Cet exemple va produire :

```
Array
(
   [un] => 1
   [cinq] => 5
   [chemin] => /usr/local/bin
)
Array
(
   [premiere_section] => Array
       (
           [un] => 1
           [cinq] => 5
       )
   [seconde_section] => Array
       (
           [chemin] => /usr/local/bin
       )
```

```
)
```

6.30.65 pathinfo() : Retourne des informations sur un chemin système

`array` **pathinfo** (`string path`)
[lien librairie]

`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
```

Voir aussi `dirname` , `parse_url` , `basename` et `realpath` .

6.30.66 pclose() : Ferme un processus de pointeur de fichier.

`int` **pclose** (`int fp`)
[lien librairie]

`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` .

6.30.67 popen() : Crée un processus de pointeur de fichier.

`int` **popen** (`string commet` , `string mode`)
[lien librairie]

`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 .

```
<?php
  $fp = popen("/bin/ls", "r");
?>
```

Voir aussi pclose .

6.30.68 readfile() : Affiche un fichier.

int **readfile** (string filename , *int use_include_path*)
[lien librairie]

readfile lit le fichier filename et l'envoit à la sortie standard.

readfile retourne le nombre d'octets lus depuis le fichier. Si une erreur survient, retourne FALSE .

Si filename commence par "http://" (insensible à la casse), une connexion HTTP 1.0 sera ouverte avec le serveur spécifié, et le texte de la réponse sera affiché sur la sortie standard.

Les versions antérieures à PHP 4.0.6, ne gère pas les redirections automatiques, ce qui oblige à ajouter les slash finaux "/" pour indiquer un dossier.

Si filename commence par "ftp://" (insensible à la casse), une connexion FTP est ouverte avec l'hôte spécifié et la réponse du serveur est affichée. Si le serveur ne supporte les connexions passives, la requête échouera.

Si filename ne commence par aucun des cas précédents, le fichier sera ouvert sur l'hôte local, et envoyé à la sortie standard.

Vous pouvez utiliser le deuxième paramètre optionnel pour explorer le dossier include_path , en passant la valeur de 1.

Voir aussi fpassthru , file , fopen , include , require et virtual .

6.30.69 readlink() : Renvoie le nom du fichier vers lequel pointe un lien symbolique.

string **readlink** (string path)
[lien librairie]

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 symlink , readlink et linkinfo .

Note
readlink est inopérante sous Windows.

6.30.70 realpath() : Retourne le chemin canonique absolu.

`string` **realpath** (`string path`)
[lien librairie]

`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 `realpath`

```
<?php
  $real_path = realpath("../../index.php");
?>
```

6.30.71 rename() : Renomme un fichier

`boolean` **rename** (`string oldname` , `string newname`)
[lien librairie]

`rename` tente de renommer le fichier `oldname` en `newname` .

`rename` retourne `TRUE` en cas de succès et `FALSE` sinon.

Exemple avec `rename`

```
<?php
  rename("/tmp/tmp_file.txt", "/home/user/login/docs/mon_file.txt");
?>
```

6.30.72 rewind() : Replace le pointeur de fichier au début.

`int` **rewind** (`int fp`)
[lien librairie]

`rewind` replace le pointeur du fichier `fp` au début.

Si une erreur survient, retourne `FALSE` .

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par `fopen` .

Voir aussi `fseek` et `ftell` .

6.30.73 rmdir() : Efface un dossier.

`int` **rmdir** (`string dirname`)
[lien librairie]

`rmdir` tente d'effacer le dossier dont le chemin est `dirname` . Le dossier doit être vide, et le script doit avoir les autorisations adéquates.

Si une erreur survient, `rmdir` retourne `FALSE` .

Voir aussi `mkdir` .

6.30.74 set_file_buffer() : Fixe la bufferisation de fichier

`int` **set_file_buffer** (`int fp` , `int buffer`)
[lien librairie]

L'écriture de fichier avec `fwrite` utilise normalement un buffer de 8K. Cela signifie que si deux processus essaient d'écrire dans le même fichier, ils font une pause tous les 8ko pour laisser le temps à l'autre d'écrire à son tour. `set_file_buffer` permet de modifier la taille du buffer de sortie pour le pointeur de fichier `fp` à `buffer` octets. Si `buffer` vaut 0, l'écriture se fera sans buffer. Cela force un processus à écrire toutes ses données dans un fichier avant que les autres puissent y accéder.

`set_file_buffer` retourne 0 en cas de succès, ou EOF si la requête ne peut pas être honorée.

L'exemple suivant montre comment utiliser la fonction `set_file_buffer` pour créer un fichier sans buffer.

Exemple avec `set_file_buffer`

```
<?php
  $fp=fopen($file, "w");
  if($fp){
    set_file_buffer($fp, 0);
    fputs($fp, $output);
    fclose($fp);
  }
?>
```

Voir aussi `fopen` et `fwrite` .

6.30.75 stat() : Renvoie les informations à propos d'un fichier.

`array` **stat** (`string filename`)
[lien librairie]

`stat` renvoie les informations à propos du fichier filename.

`stat` retourne un tableau avec les éléments suivants :

- 0 : volume
- 1 : inode
- 2 : droits d'accès au fichier (mode de protection du inode). A convertir en octal. Voir aussi `fileperms` .
- 3 : nombre de liens
- 4 : id de l'utilisateur propriétaire

- 5 : id du groupe propriétaire
- 6 : type du volume de l'inode *
- 7 : taille en octets
- 8 : date du dernier accès
- 9 : date de la dernière modification
- 10 : date du dernier changement
- 11 : taille de bloc du système pour les entrées/sorties *
- 12 : nombre de blocs alloués

* − uniquement sur les systèmes qui supportent le type st_blksize. Les autres systèmes (i.e. Windows) retournent −1.

`stat` retourne FALSE en cas d'erreur.

`stat` ne gère pas les URL comme peut le faire `fopen` .

Les résultats de `stat` sont mis en cache. Reportez−vous à la fonction `clearstatcache` pour plus de détails.

6.30.76 stream_get_meta_data() : Retrieves header/meta data from streams/file pointers

`array` **file_get_meta_data** (`resource stream`)
[lien librairie]

Returns information about an existing `stream` . The stream can be any stream created by `fopen` , `fsockopen` and `pfsockopen` . The result array contains the following items:

- `timed_out` (bool) − TRUE if the stream timed out while waiting for data on the last call to `fread` or `fgets` .
- `blocked` (bool) − TRUE if the stream is in blocking IO mode. See `socket_set_blocking` .
- `eof` (bool) − TRUE if the stream has reached end−of−file. Note that for socket streams this member can be TRUE even when `unread_bytes` is non−zero. To determine if there is more data to be read, use `feof` instead of reading this item.
- `unread_bytes` (int) − the number of bytes currently contained in the read buffer.

The following items were added in PHP 4.3:

- `stream_type` (string) − a label describing the underlying implementation of the stream.
- `wrapper_type` (string) − a label describing the protocol wrapper implementation layered over the stream. See for more information about wrappers.
- `wrapper_data` (mixed) − wrapper specific data attached to this stream. See for more information about wrappers and their wrapper data.
- `filters` (array) − and array containing the names of any filters that have been stacked onto this stream. Filters are currently undocumented.

Note
This function was introduced in PHP 4.3, but prior to this version, `socket_get_status` could be used to retrieve the first four items, for **socket based streams only** . In PHP 4.3 and later, `socket_get_status` is an alias for this function.

Note
This function does NOT work on sockets created by the Socket extension .

6.30.77 stream_register_wrapper() : Enregistre une enveloppe URL, implémenté comme classe PHP

`boolean` **stream_register_wrapper** (`string protocol` , `string classname`)
[lien librairie]

`stream_register_wrapper` vous permet d'implémenter vos propres gestionnairs 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_register_wrapper` retourne FALSE si le protocole `protocol` a déjà un gestionnaire attitré.

`boolean` **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 `path` est relatif, recherche la ressources en utilisant la configuration de l'include_path.
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 méthod 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 données 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 flôt, 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 flôt. 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 flôt, en ajoutant ou retranchant le nombre d'octets lus.

`boolean` **stream_eof** (`void`)

Cette méthode est appelée lorsque la fonction `feof` est utilisée. Vous devez retourner TRUE si la position de lecture se situe a la fin du fichier et si il n'y a plus de données disponibles pour la lecture, ou bien FALSE sinon.

`int` **stream_tell** (`void`)

Cette méthode est appelée lorsque la fonction `ftell` est utilisée. Vous devez retourner la position actuelle du pointeur de lecture/écriture.

`boolean` **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.

`boolean` **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 flôt, 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 données à stocker), ou bien FALSE si les données n'ont pu être stockées.

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"].

A Stream for reading/writing global variables

```
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_register_wrapper("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);
```

6.30.78 stream_set_blocking() : Configure le mode bloquant d'un flôt

`bool` **stream_set_blocking** (`resource stream` , `int mode`)
[lien librairie]

`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 seul qui soit supporté dans le coeur de PHP, même si les autres extensions pourrait supporter cette fonction.

6.30.79 stream_set_timeout() : Configure la durée d'expiration d'un flôt

`bool` **stream_set_timeout** (`resource stream` , `int seconds` , `int microseconds`)
[lien librairie]

`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 seul qui soit supporté dans le coeur de PHP, même si les autres extensions pourrait 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` .

6.30.80 stream_set_write_buffer() : Configure la bufferisation de fichier pour un flôt

`int` **stream_set_write_buffer** (`resource stream` , `int buffer`)
[lien librairie]

`fwrite` est habituellement configuré avec un buffer de 8ko. Cela signifie que si deux processus veulent écrirent 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 .

6.30.81 symlink() : Crée un lien symbolique.

int **symlink** (string target , string link)
[lien librairie]

symlink crée un lien symbolique pour l'objet target avec le nom de link .

Voir aussi link pour créer des liens durs et readlink ainsi que linkinfo .

Note
symlink est inopérante sous Windows.

6.30.82 tempnam() : Crée un fichier avec un nom unique.

string **tempnam** (string dir , string prefix)
[lien librairie]

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

```
<?php
  $tmpfname = tempnam("/tmp", "FOO");
?>
```

6.30.83 tmpfile() : Crée un fichier temporaire

int **tmpfile** (void)
[lien librairie]

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 .

Voir aussi tempnam .

6.30.84 touch() : Affecte une nouvelle date de modification à un fichier.

`int` **touch** (`string filename` , `int time`)
[lien librairie]

`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, il est créé.

`touch` retourne TRUE en cas de succès, et FALSE sinon.

Exemple avec `touch`

```
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";
}
```

6.30.85 umask() : Change le "umask" courant.

`int` **umask** (`int mask`)
[lien librairie]

`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 argument retourne simplement le umask courant.

6.30.86 unlink() : Efface un fichier

`int` **unlink** (`string filename`)
[lien librairie]

`unlink` efface `filename` . Identique à la fonction Unix C unlink().

`unlink` retourne FALSE en cas d'échec.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `rmdir` pour supprimer des dossiers.

6.31 Forms Data Format

6.31.1 Introduction

Forms Data Format (FDF) est un format de fomulaire pour les documents PDF. Vous pouvez lire la documentation (en anglais) à http://partners.adobe.com/asn/developer/acrosdk/forms.html 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 a 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..

6.31.2 Pré−requis

Vous aurez besoin du FDF toolkit SDK, disponible sur le site http://partners.adobe.com/asn/developer/acrosdk/forms.html . Depuis PHP 4.3, vous aurez besoin du SDK version 5.0. La librairie FDF toolkit est disponible sous forme de librarie compilée, éditée par Adobe, sur les systèmes d'exploitation Win32, Linux, Solaris et AIX.

6.31.3 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'entête `fdftk.h` et la librairie `libfdftk.so` sont à leur place. Le fichier de configuration supporte la hierarchie 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'entêtes 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` .

6.31.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.31.5 Types de ressources

6.31.5.1 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` .

6.31.6 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` (nombre ˆ virgule flottante)
`FDFStatus` (nombre ˆ virgule flottante)
`FDFFile` (nombre ˆ virgule flottante)
`FDFID` (nombre ˆ virgule flottante)
`FDFFf` (nombre ˆ virgule flottante)
`FDFSetFf` (nombre ˆ virgule flottante)
`FDFClearFf` (nombre ˆ virgule flottante)
`FDFFlags` (nombre ˆ virgule flottante)
`FDFSetF` (nombre ˆ virgule flottante)
`FDFClrF` (nombre ˆ virgule flottante)
`FDFAP` (nombre ˆ virgule flottante)
`FDFAS` (nombre ˆ virgule flottante)
`FDFAction` (nombre ˆ virgule flottante)
`FDFAA` (nombre ˆ virgule flottante)
`FDFAPRef` (nombre ˆ virgule flottante)
`FDFIF` (nombre ˆ virgule flottante)
`FDFEnter` (nombre ˆ virgule flottante)
`FDFExit` (nombre ˆ virgule flottante)
`FDFDown` (nombre ˆ virgule flottante)
`FDFUp` (nombre ˆ virgule flottante)
`FDFFormat` (nombre ˆ virgule flottante)
`FDFValidate` (nombre ˆ virgule flottante)
`FDFKeystroke` (nombre ˆ virgule flottante)
`FDFCalculate` (nombre ˆ virgule flottante)
`FDFNormalAP` (nombre ˆ virgule flottante)
`FDFRolloverAP` (nombre ˆ virgule flottante)
`FDFDownAP` (nombre ˆ virgule flottante)

6.31.7 Exemples

Les exemples suivants montre comme 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 "Le champ date contient la valeur : '<B>$date</B>'<BR>";

$comment = fdf_get_value($fdf, "comment");
echo "Le champ commentaire contient la valeur :  '<B>$comment</B>'<BR>";

if(fdf_get_value($fdf, "show_publisher") == "On") {
  $publisher = fdf_get_value($fdf, "publisher");
  echo "Le champ publisher contient la valeur :  '<B>$publisher</B>'<BR>";
} else
  echo "Le publisher ne doit pas être divulgué.<BR>";

if(fdf_get_value($fdf, "show_preparer") == "On") {
  $preparer = fdf_get_value($fdf, "preparer");
  echo "Le champ Preparer contient la valeur :  '<B>$preparer</B>'<BR>";
} else
  echo "Le Preparer ne doit pas être divulgué.<BR>";
fdf_close($fdf);
?>
```


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- Exemples
- fdf_close
- fdf_create
- fdf_get_file
- fdf_get_status
- fdf_get_value
- fdf_next_field_name
- fdf_open
- 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_value

6.31.9 fdf_create() : Crée un nouveau document FDF.

`int` **fdf_create** (`void`)
[lien librairie]

`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` .

6.31.10 fdf_get_file() : Lit la valeur de la clé /F.

`string` **fdf_get_file** (`int fdf_document`)
[lien librairie]

`fdf_set_file` lit la valeur de la clé /F.

Voir aussi `fdf_set_file` .

6.31.11 fdf_get_status() : Lit la valeur de la clé /STATUS.

`string` **fdf_get_status** (`int fdf_document`)
[lien librairie]

`fdf_get_status` retourne la valeur de la clé /STATUS.

Voir aussi `fdf_set_status` .

6.31.12 fdf_get_value() : Mot la valeur d'un champs.

`string` **fdf_get_value** (`int fdf_document` , `string fieldname`)
[lien librairie]

`fdf_get_value` retourne la valeur d'un champs.

Voir aussi `fdf_set_value` .

6.31.13 fdf_next_field_name() : Lit le nom du champs suivant.

`string` **fdf_next_field_name** (`int fdf_document` , `string fieldname`)
[lien librairie]

`fdf_next_field_name` retourne le nom du champs après le champs `fieldname` ou le nom du premier champs, si le second paramètre est NULL.

Voir aussi `fdf_set_value` et `fdf_get_value` .

6.31.14 fdf_open() : Ouvre un document FDF.

`resource` **fdf_open** (`string filename`)
[lien librairie]

`fdf_open` ouvre un fichier avec formulaire. Le fichier doit contenir les données retournées par le formulaire PDF. Actuellement, le fichier doit être créée '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 champs entrant, n'existe pas.

Accéder aux données du formulaire

```
<?php
// Sauver le fichier FDF dans un fichier temporaire.
$fdffp = fopen("test.fdf", "w");
fwrite($fdffp, $HTTP_FDF_DATA, strlen($HTTP_FDF_DATA));
fclose($fdffp);
// Ouvrir le fichier temporaire, et utiliser les données.
$fdf = fdf_open("test.fdf");
...
fdf_close($fdf);
?>
```

Voir aussi `fdf_close` .

6.31.15 fdf_save() : Sauver un document FDF.

`int` **fdf_save** (`string filename`)
[lien librairie]

`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_close` et pour avoir un exemple `fdf_create` .

6.31.16 fdf_set_ap() : Fixe l'apparence d'un champs.

`boolean` **fdf_set_ap** (`int fdf_document` , `string field_name` , `int face` , `string filename` , `int page_number`)
[lien librairie]

`fdf_set_ap` fixe l'apparence d'un champs (i.e. la valeur de la clé /AP). Les valeurs possibles de `face` sont sont 1=FDFNormalAP, 2=FDFRolloverAP, 3=FDFDownAP.

6.31.17 fdf_set_encoding() : Modifie l'encodage des caractères

`bool` **fdf_set_encoding** (`int fdf_document` , `string encoding`)
[lien librairie]

`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.

6.31.18 fdf_set_file() : Fixe la valeur de la clé /F.

`boolean` **fdf_set_file** (`int fdf_document` , `string filename`)
[lien librairie]

`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).

Voir aussi `fdf_get_file` et pour un exemple, `fdf_create` .

6.31.19 fdf_set_flags() : Modifie une option d'un champs

`boolean` **fdf_set_flags** (`int fdf_document` , `string fieldname` , `int whichFlags` , `int newFlags`)
[lien librairie]

`fdf_set_flags` modifie certaines options du champs `fieldname` .

Voir aussi `fdf_set_opt` .

6.31.20 fdf_set_javascript_action() : Modifie l'action javascript d'un champs

`boolean` **fdf_set_javascript_action** (`int fdf_document` , `string fieldname` , `int trigger` , `string script`)
[lien librairie]

fdf_set_javascript_action affecte un javascript au champs fieldname , exécuté lors de la validation d'un formulaire.

Voir aussi fdf_set_submit_form_action .

6.31.21 fdf_set_opt() : Modifie une option d'un champs

boolean **fdf_set_opt** (int fdf_document , string fieldname , int element , string str1 , string str2)
[lien librairie]

fdf_set_opt modifie les options du champs fieldname .

Voir aussi fdf_set_flags .

6.31.22 fdf_set_status() : Fixe la valeur de la clé /STATUS.

boolean **fdf_set_status** (int fdf_document , string status)
[lien librairie]

fdf_set_status fixe la valeur de la clé /STATUS.

Voir aussi fdf_get_status .

6.31.23 fdf_set_submit_form_action() : Modifie l'action d'un formulaire

boolean **fdf_set_submit_form_action** (int fdf_document , string fieldname , int trigger , string script , int flags)
[lien librairie]

fdf_set_submit_form_action affecte un javascript au champs fieldname , exécuté lors de la validation d'un formulaire.

Voir aussi fdf_set_javascript_action .

6.31.24 fdf_set_value() : Fixe la valeur d'un champs.

boolean **fdf_set_value** (int fdf_document , string fieldname , string value , int isName)
[lien librairie]

fdf_set_value fixe la valeur d'un champs. 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 .

6.32 FriBiDi

6.32.1 Introduction

6.32.2 Pré−requis

6.32.3 Installation

6.32.4 Configuration à l'exécution

6.32.5 Types de ressources

6.32.6 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 (nombre ˆ virgule flottante)
FRIBIDI_CHARSET_8859_6 (nombre ˆ virgule flottante)
FRIBIDI_CHARSET_8859_8 (nombre ˆ virgule flottante)
FRIBIDI_CHARSET_CP1255 (nombre ˆ virgule flottante)
FRIBIDI_CHARSET_CP1256 (nombre ˆ virgule flottante)
FRIBIDI_CHARSET_ISIRI_3342 (nombre ˆ virgule flottante)

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- fribidi_log2vis

6.33 FTP

6.33.1 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 .

6.33.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.33.3 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.

6.33.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.33.5 Types de ressources

This extension uses one resource type, which is the link identifier of the FTP connection.

6.33.6 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` (nombre ˆ virgule flottante)
`FTP_TEXT` (nombre ˆ virgule flottante)
`FTP_BINARY` (nombre ˆ virgule flottante)
`FTP_IMAGE` (nombre ˆ virgule flottante)
`FTP_TIMEOUT_SEC` (nombre ˆ virgule flottante)
Voir `ftp_set_option` pour les détails.

Les constantes suivantes ont été introduites en PHP 4.3.0.

`FTP_AUTOSEEK` (nombre ˆ virgule flottante)
Voir `ftp_set_option` pour les détails.
`FTP_AUTORESUME` (nombre ˆ virgule flottante)
Déterimine 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` (nombre ˆ virgule flottante)
Le mode asynchrone a échoué
`FTP_FINISHED` (nombre ˆ virgule flottante)

Le mode asynchrone a terminé
FTP_MOREDATA (nombre ˆ virgule flottante)
Le mode asynchrone est encore actif

6.33.7 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_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_rawlist
- ftp_rename
- ftp_rmdir
- ftp_set_option
- ftp_site
- ftp_size
- ftp_systype

6.33.9 ftp_chdir() : Change le dossier courant.

`bool` **ftp_chdir** (`resource ftp_stream` , `string directory`)
[lien librairie]

`ftp_chdir` retourne TRUE en cas de succès, et FALSE sinon.

`ftp_chdir` change le dossier courant en `directory` .

6.33.10 ftp_close() : Ferme une connexion FTP

`void` **ftp_close** (`resource ftp_stream`)
[lien librairie]

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` .

6.33.11 ftp_connect() : Ouvre une connexion FTP

`resource` **ftp_connect** (`string host` , *`int port`*)
[lien librairie]

`ftp_connect` retourne un flot 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.

6.33.12 ftp_delete() : Efface un fichier sur un serveur FTP.

`bool` **ftp_delete** (`resource ftp_stream` , `string path`)
[lien librairie]

`ftp_delete` retourne TRUE en cas de succès, et FALSE sinon.

`ftp_delete` efface le fichier `path` sur un serveur FTP.

6.33.13 ftp_exec() : Exécute un programme sur le serveur FTP

`bool` **ftp_exec** (`resource ftp_stream` , `string command`)
[lien librairie]

`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.

6.33.14 ftp_fget() : Télécharge un fichier depuis un serveur FTP et le sauve dans un fichier déjà ouvert.

`bool` **ftp_fget** (`resource ftp_stream` , `int fp` , `string remote_file` , `int mode`)
[lien librairie]

`ftp_fget` retourne TRUE en cas de succès, et FALSE sinon.

`ftp_fget` télécharge le fichier `remote_file` depuis le serveur FTP, et l'écrit dans le fichier identifié par `fp` . Le mode de transfert `mode` spécifié doit être FTP_ASCII ou FTP_BINARY.

6.33.15 ftp_fput() : Charge un fichier ouvert sur un serveur FTP.

`bool` **ftp_fput** (`resource ftp_stream` , `string remote_file` , `int fp` , `int mode`)
[lien librairie]

`ftp_fput` retourne TRUE en cas de succès, et FALSE sinon.

`ftp_fput` charge les données issues du fichier identifié par `fp` 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.

6.33.16 ftp_get_option() : Lit différenctes options pour la connexion FTP courante

`mixed` **ftp_get_option** (`resource ftp_stream` , `int option`)
[lien librairie]

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'éxpriation pour les opérations liées au réseau.

Exemple avec ftp_get_option

```
<?php
  // Lit le délai d'éxpiration pour la connexion en cours
  $timeout = ftp_get_option($conn_id, FTP_TIMEOUT_SEC);
?>
```

6.33.17 ftp_get() : Télécharge un fichier depuis un serveur FTP.

bool **ftp_get** (resource ftp_stream , string local_file , string remote_file , int mode)
[lien librairie]

ftp_get retourne TRUE en cas de succès, et FALSE sinon.

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.

6.33.18 ftp_login() : Authentification d'une connexion FTP

bool **ftp_login** (resource ftp_stream , string username , string password)
[lien librairie]

ftp_login retourne TRUE en cas de succès, et FALSE sinon.

ftp_login authentifie le flot FTP.

6.33.19 ftp_mdtm() : Retourne la date de dernière modification d'un fichier sur un serveur FTP.

int **ftp_mdtm** (resource ftp_stream , string remote_file)
[lien librairie]

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.

6.33.20 ftp_mkdir() : Crée un dossier.

`string` **ftp_mkdir** (`resource ftp_stream` , `string directory`)
[lien librairie]

`ftp_mkdir` retourne le nom du dossier ainsi créé en cas de succès, et `FALSE` sinon.

`ftp_mkdir` crée le dossier nommé `directory` .

6.33.21 ftp_nb_continue() : Reprend le téléchargement d'un fichier (non bloquant)

`bool` **ftp_nb_continue** (`resource ftp_stream`)
[lien librairie]

`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` .

6.33.22 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 fp` , `string remote_file` , `int mode` , *`int resumepos`*)
[lien librairie]

`ftp_nb_fget` lit le fichier `remote_file` présent sur le serveur FTP `ftp_stream` , puis écrit ce fichier dans le pointeur de fichier `fp` . 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.

`ftp_nb_fget` retourne `TRUE` en cas de succès, et `FALSE` sinon.

Voir aussi `ftp_nb_get` .

6.33.23 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 fp` , `int mode` , *`int startpos`*)
[lien librairie]

`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 `fp` . 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.

`ftp_nb_fput` retourne `TRUE` en cas de succès, et `FALSE` sinon.

Voir aussi ftp_nb_put , ftp_nb_continue , ftp_put et ftp_fput .

6.33.24 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*)
[lien librairie]

ftp_nb_get lit le fichier remote_file présent sur le serveur FTP ftp_stream , puis écrit ce fichier dans le pointeur de fichier fp . 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.

ftp_nb_get retourne TRUE en cas de succès, et FALSE sinon.

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` .

6.33.25 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`*)
[lien librairie]

`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 `fp` . 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.

`ftp_nb_put` retourne `TRUE` en cas de succès, et `FALSE` sinon.

`ftp_nb_put` example

```
// 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 .

6.33.26 ftp_nlist() : Retourne la liste des fichiers dans un dossier.

array **ftp_nlist** (resource ftp_stream , string directory)
[lien librairie]

ftp_nlist retourne un tableau de nom de fichiers en cas de succès, et FALSE sinon.

6.33.27 ftp_pasv() : Active ou désactive le mode passif.

bool **ftp_pasv** (resource ftp_stream , int pasv)
[lien librairie]

ftp_pasv retourne TRUE en cas de succès, et FALSE sinon.

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.

6.33.28 ftp_put() : Charge un fichier sur un serveur FTP.

bool **ftp_put** (resource ftp_stream , string remote_file , string local_file , int mode)
[lien librairie]

ftp_put retourne TRUE en cas de succès, et FALSE sinon.

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 .

6.33.29 ftp_pwd() : Retourne le nom du dossier courant.

`string` **ftp_pwd** (`resource ftp_stream`)
[lien librairie]

`ftp_pwd` retourne le nom du dossier courant, ou FALSE en cas d'erreur.

6.33.30 ftp_quit() : Ferme une connexion FTP.

`bool` **ftp_quit** (`resource ftp_stream`)
[lien librairie]

`ftp_quit` ferme la connexion `ftp_stream` .

6.33.31 ftp_rawlist() : Fait une liste détaillée de fichiers dans un dossier.

`array` **ftp_rawlist** (`resource ftp_stream` , `string directory`)
[lien librairie]

`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.

6.33.32 ftp_rename() : Renomme un fichier sur un serveur FTP.

`bool` **ftp_rename** (`resource ftp_stream` , `string from` , `string to`)
[lien librairie]

`ftp_rename` retourne TRUE en cas de succès, et FALSE sinon.

`ftp_rename` renomme le fichier ou dossier `from` en `to` , sur le serveur `ftp_stream` .

6.33.33 ftp_rmdir() : Efface un dossier.

`bool` **ftp_rmdir** (`resource ftp_stream` , `string directory`)
[lien librairie]

`ftp_rmdir` retourne TRUE en cas de succès, et FALSE sinon.

`ftp_rmdir` efface le dossier `directory` .

6.33.34 ftp_set_option() : Modifie les options de la connexion FTP

`bool` **ftp_set_option** (`resource ftp_stream` , `int option` , `mixed value`)
[lien librairie]

Note

ftp_set_option n'est disponible que dans le CVS.

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);
?>
```

6.33.35 ftp_site() : Envoie la commande SITE au serveur.

bool **ftp_site** (resource ftp_stream , string cmd)
[lien librairie]

ftp_site retourne TRUE en cas de succès, et FALSE sinon.

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.

6.33.36 ftp_size() : Retourne la taille d'un fichier.

int **ftp_size** (resource ftp_stream , string remote_file)
[lien librairie]

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.

6.33.37 ftp_systype() : Retourne un identifiant de type de serveur FTP.

string **ftp_systype** (resource ftp_stream)
[lien librairie]

`ftp_systype` retourne le type de serveur, ou FALSE en cas d'erreur.

6.34 Fonctions

Ces fonctions effectuent les manipulations liées à la gestion des fonctions.

- 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

6.34.2 call_user_func() : Appelle une fonction utilisateur

`mixed` **call_user_func** (`string function_name` , *`mixed parameter`* , *`mixed ...`*)
[lien librairie]

`call_user_func` appelle la fonction utilisateur `function_name` , et lui passe les paramètres `parameter` . Par exemple :

```
<?php
  function barbier ($type) {
    print "Vous vouliez une coupe $type, pas de problème";
  }
  call_user_func('barbier', "iroquois");
  call_user_func('barbier', "militaire");
  call_user_func('barbier', "au bol");
?>
```

Voir aussi `call_user_func_array` , `call_user_method` et `call_user_method_array` .

6.34.3 create_function() : Crée une fonction anonyme (style lambda)

`string` **create_function** (`string args` , `string code`)
[lien librairie]

`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 proteger 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'éxécution.

Création d'une fonction anonyme avec `create_function`

```php
<?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
<?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 "paramétres: 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,$a','if (strncmp($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 brilling and the slithy toves", "Twas the night", $garr);
?>
```

Et lorsque vous utilisez le code ci−dessus, l'affichage va être

```
Utilisation de la première liste de fonctions anonymes
paramétres: 2.3445, M_PI
Un peu de trigo: −1.6291725057799
Une hypoténuse: 3.9199852871011
b*a^2 = 4.8103313314525
min(b^2+a, a^2,b) = 8.6382729035898
ln(a/b) = 0.27122299212594
Utilisation de la seconde liste de fonctions anonymes
** "Twas the night" et "Twas brilling 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
<?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
<A HREF="function.var-dump.php"><TT>var_dump</TT></a>

// 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
// )
?>
```

6.34.4 func_get_arg() : Retourne un élément de la liste des arguments

`mixed` **`func_get_arg`** `( int arg_num )`
[lien librairie]

`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 .

```
<?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.

6.34.5 func_get_args() : Retourne les arguments d'une fonction sous forme de tableau

`array` **func_get_args** `( void )`
[lien librairie]

`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.

```
<?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.

6.34.6 func_num_args() : Retourne le nombre d'arguments passé à la fonction

`int` **func_num_args** `( void )`
[lien librairie]

`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.

```
<?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.

6.34.7 function_exists() : Indique si une fonction est définie.

`boolean` **function_exists** (`string function_name`)
[lien librairie]

`function_exists` vérifie la liste des fonctions définies par l'utilisateur, et retourne TRUE si `function_name` y est trouvé, FALSE sinon.

```
<?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` .

6.34.8 get_defined_functions() : Liste toutes les fonctions définies

`array` **get_defined_functions** (`void`)
[lien librairie]

`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"]` .

```
<?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` .

6.34.9 register_shutdown_function() : Enregistre une fonction pour exécution à l'extinction

`int` **register_shutdown_function** (`string func`)
[lien librairie]

`register_shutdown_function` enregistre la fonction `func` pour exécution à l'extinction du script.

Erreur courante :

Etant donné qu'aucun affichage n'est possible depuis la fonction `func` , vous ne pouvez pas y mettre d'informations de débuggage par `print` ou `echo` !

6.34.10 register_tick_function() : Enregistre une fonction exécutée à chaque tick

`void` **register_tick_function** (`string func` , *`mixed arg`*)
[lien librairie]

`register_tick_function` enregistre la fonction `func` pour être exécutée à chaque fois qu'un tick est appelé.

6.34.11 unregister_tick_function() : Annule la fonction exécutée à chaque tick

`void` **unregister_tick_function** (`string func` , *`mixed arg`*)
[lien librairie]

`unregister_tick_function` annule l'exécution automatique de `func` à chaque tick .

6.35 Gettext (GNU)

Les fonctions gettext implémentent l'API NLS (Native Language Support) qui peut servir à internationaliser vos scripts PHP. Lisez la documentation GNU pour plus d'explications sur ces fonctions.

- bindtextdomain
- dcgettext
- dgettext
- gettext
- textdomain

6.35.2 dcgettext() : Remplace le domaine lors d'une recherche.

`string` **dcgettext** `( string domain , string message , int category )`
[lien librairie]

`dcgettext` permet de remplacer le domaine courant lors de la recherche d'un message. Elle permet aussi de spécifier une catégorie.

6.35.3 dgettext() : Remplace le domaine courant.

`string` **dgettext** `( string domain , string message )`
[lien librairie]

`dgettext` remplace le domaine courant.

6.35.4 gettext() : Recherche un message dans le domaine courant.

`string` **gettext** `( string message )`
[lien librairie]

`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
<?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"));
?>
```

6.35.5 textdomain() : Fixe le domaine par défaut.

`string` **textdomain** (`string text_domain`)
[lien librairie]

`textdomain` fixe le domaine à utiliser lors de recherche avec `gettext` . Ce domaine dépend généralement de l'application. Le domaine par défaut précédent est retourné. Appelez cette fonction avec with `NULL` comme paramètre pour avoir la valeur courante, sans la modifier.

6.36 GMP

6.36.1 Introduction

Ces fonctions vous permettent de travailler avec des nombres de taille arbitraire, en utilisant la librairie GNU MP .

Ces fonctions ont été ajoutées en PHP 4.0.4.

Note
La majorité des fonctions GMP acceptent des nombres GMP comme arguments, définis ci−dessous comme `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.

6.36.2 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 .

6.36.3 Installation

Pour pouvoir y accéder, vous devez compiler PHP avec le support GMP en utilisant l'option `−−with−gmp` .

6.36.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.36.5 Types de ressources

Cette extension ne définit aucune ressource.

6.36.6 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 (nombre ˆ virgule flottante)
GMP_ROUND_PLUSINF (nombre ˆ virgule flottante)
GMP_ROUND_MINUSINF (nombre ˆ virgule flottante)

6.36.7 Exemples

Factorielle avec GMP

```
<?php
function fact($x) {
  if($x <= 1)
        return 1;
  else
        return gmp_mul($x,fact($x-1));
}
print gmp_strval(fact(1000))."\n";
?>
```

Cet exemple va calculer factorielle de 1000 (un plutôt grand nombre) très vite.

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- Exemples
- 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

6.36.9 gmp_add() : Addition de 2 nombres GMP

`resource` **gmp_add** (`resource a` , `resource b`)
[lien librairie]

`gmp_add` additionne les nombres GMP `a` et `b` . Le résultat est un nombre GMP.

6.36.10 gmp_and() : ET logique

`resource` **gmp_and** (`resource a` , `resource b`)
[lien librairie]

`gmp_and` calcule le ET logique de `a` et `b` .

6.36.11 gmp_clrbit() : Annule un bit

`resource` **gmp_clrbit** (`resource &a` , `int index`)
[lien librairie]

`gmp_clrbit` met le bit `index` à 0 dans le nombre GMP `a` .

6.36.12 gmp_cmp() : Compare des nombres GMP

`int` **gmp_cmp** (`resource a` , `resource b`)
[lien librairie]

`gmp_cmp` retourne une valeur positive si `a > b` , zéro si `a = b` et négative si `a < b` .

6.36.13 gmp_com() : Calcule le complémentaire d'un nombre

`resource` **gmp_com** (`resource a`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.36.14 gmp_div_q() : Divisions de 2 nombres GMP

`resource` **gmp_div_q** (`resource a` , `resource b` , *`int round`*)
[lien librairie]

`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` .

6.36.15 gmp_div_qr() : Divise deux nombres GMP

`array` **gmp_div_qr** (`resource n` , `resource d` , *`int round`*)
[lien librairie]

`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` .

6.36.16 gmp_div_r() : Reste de la division de deux nombres GMP

`resource` **gmp_div_r** (`resource n` , `resource d` , *`int round`*)
[lien librairie]

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 .

6.36.17 gmp_div() : Divise deux nombres GMP

resource **gmp_div** (resource a , resource b)
[lien librairie]

gmp_div est un alias de gmp_div_q .

6.36.18 gmp_divexact() : Division exacte de nombres GMP

resource **gmp_divexact** (resource n , resource d)
[lien librairie]

gmp_divexact divis 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 .

6.36.19 gmp_fact() : Factorielle GMP

resource **gmp_fact** (int a)
[lien librairie]

gmp_fact calcule la factorielle de a : a! .

6.36.20 gmp_gcd() : PGCD

resource **gmp_gcd** (resource a , resource b)
[lien librairie]

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.

6.36.21 gmp_gcdext() : PGCD étendu

array **gmp_gcdext** (resource a , resource b)
[lien librairie]

gmp_gcdext calcule les entiers g, s, et t, tels que a*s + b*t = g = gcd(a,b) , où gcd est le pgcd de a et b . La fonction retourne un tableau avec les éléments g, s et t.

6.36.22 gmp_hamdist() : Distance de Hamming

`int` **gmp_hamdist** (`resource a` , `resource b`)
[lien librairie]

`gmp_hamdist` retourne la distance de Hamming entre a et b . Les deux paramètres doivent être strictement positifs.

6.36.23 gmp_init() : Crée un nombre GMP

`resource` **gmp_init** (`mixed number`)
[lien librairie]

`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("0xFFFFDEBACDFEDF7200");
?>
```

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` .

6.36.24 gmp_intval() : Convertit un nombre GMP en entier.

`int` **gmp_intval** (`resource gmpnumber`)
[lien librairie]

`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 vous voulez simplement afficher un nombre GMP, utilisez plutôt `gmp_strval` .

6.36.25 gmp_invert() : Inverse modulo

`resource` **gmp_invert** (`resource a` , `resource b`)
[lien librairie]

`gmp_invert` calcule l'inverse de a modulo `b` . `gmp_invert` retourne FALSE si un tel inverse n'existe pas.

6.36.26 gmp_jacobi() : Symbole de Jacobi

`int` **gmp_jacobi** (`resource a` , `resource p`)
[lien librairie]

`gmp_jacobi` calcule le symbole de Jacobi de a et `p` . p doit être positif et impair.

6.36.27 gmp_legendre() : Symbole de Legendre

`int` **gmp_legendre** (`resource a` , `resource p`)
[lien librairie]

`gmp_legendre` calcule le symbole de Legendre de a et `p` . p doit être positif et impair.

6.36.28 gmp_mod() : Modulo GMP

`resource` **gmp_mod** (`resource n` , `resource d`)
[lien librairie]

`gmp_mod` calcule n modulo `d` . Le résultat est toujours positif ou nul, car le signe de `d` est ignoré.

6.36.29 gmp_mul() : Multiplication de 2 nombres GMP

`resource` **gmp_mul** (`resource a` , `resource b`)
[lien librairie]

`gmp_mul` multiplie les nombres GMP a et `b` . Le résultat est un nombre GMP.

6.36.30 gmp_neg() : Opposé de nombre GMP

`resource` **gmp_neg** (`resource a`)
[lien librairie]

`gmp_neg` retourne l'opposé de a (−1 * a).

6.36.31 gmp_or() : OU logique

`resource` **gmp_or** (`resource a` , `resource b`)
[lien librairie]

gmp_or calcule le OU logique de a et b .

6.36.32 gmp_perfect_square() : Carré parfait GMP

boolean **gmp_perfect_square** (resource a)
[lien librairie]

gmp_perfect_square retourn TRUE si a est un carré parfait, et FALSE sinon.

Voir aussi gmp_sqrt et gmp_sqrtrm .

6.36.33 gmp_popcount() : Compte de population

int **gmp_popcount** (resource a)
[lien librairie]

gmp_popcount dénombre la population de a .

6.36.34 gmp_pow() : Puissance

resource **gmp_pow** (resource base , int exp)
[lien librairie]

gmp_pow élève base à la puissance exp . Dans le cas de 0^0, gmp_pow retourne 1. exp ne doit pas être négatif.

6.36.35 gmp_powm() : Puissance et modulo

resource **gmp_powm** (resource base , resource exp , resource mod)
[lien librairie]

gmp_powm calcule (base puissnace exp) modulo mod . Si exp est négatif, le résultat est indéfini.

6.36.36 gmp_prob_prime() : Nombre GMP probablement premier

int **gmp_prob_prime** (resource a , *int reps*)
[lien librairie]

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.

6.36.37 gmp_random() : Nombre GMP aléatoire

`resource` **gmp_random** (`int limiter`)
[lien librairie]

`gmp_random` génère un nombre aléatoire. Ce nombre est limité par `limiter` . Si `limiter` est négatif, un nombre négatif est généré.

6.36.38 gmp_scan0() : Recherche 0

`int` **gmp_scan0** (`resource a` , `int start`)
[lien librairie]

`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.

6.36.39 gmp_scan1() : Recherche 1

`int` **gmp_scan1** (`resource a` , `int start`)
[lien librairie]

`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.

6.36.40 gmp_setbit() : Modifie un bit

`resource` **gmp_setbit** (`resource &a` , `int index` , *`boolean set_clear`*)
[lien librairie]

`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.

6.36.41 gmp_sign() : Signe du nombre GMP

`int` **gmp_sign** (`resource a`)
[lien librairie]

`gmp_sign` retourne le signe de a : 1 si a est positif et −1 s'il est négatif.

6.36.42 gmp_sqrt() : Racine carrée GMP

`resource` **gmp_sqrt** (`resource a`)
[lien librairie]

`gmp_sqrt` calcule la racine carrée de a .

6.36.43 gmp_sqrtrm() : Racine carrée avec reste GMP

`array` **gmp_sqrtrm** (`resource a`)
[lien librairie]

`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).

6.36.44 gmp_strval() : Convertit un nombre GMP en chaîne

`string` **gmp_strval** (`resource gmpnumber` , *`int base`*)
[lien librairie]

`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));
?>
```

6.36.45 gmp_sub() : Soustraction de 2 nombres GMP

`resource` **gmp_sub** (`resource a` , `resource b`)
[lien librairie]

`gmp_sub` soustrait le nombre GMP b de a . Le résultat est un nombre GMP.

6.36.46 gmp_xor() : OU exclusif logique

`resource` **gmp_xor** (`resource a` , `resource b`)
[lien librairie]

`gmp_or` calcule le OU exclusif logique de a et b .

6.37 HTTP

Ces fonctions permettent de travailler sur les informations transmises au navigateur, via le protocole HTTP.

- header
- headers_sent
- setcookie

6.37.2 headers_sent() : Indique si les entêtes HTTP ont déjà été envoyés

`boolean` **headers_sent** (`void`)
[lien librairie]

`headers_sent` retourne `TRUE` si les entêtes `HTTP` ont déjà été envoyés, et `FALSE` sinon.

Voir aussi `header` .

6.37.3 setcookie() : Envoie un cookie

`int` **setcookie** (`string name` , *`string value`* , *`int expire`* , *`string path`* , *`string domain`* , *`int secure`*)
[lien librairie]

`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>` .

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.

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 dan 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 :

```
<?php
  echo $TestCookie;
  echo $HTTP_COOKIE_VARS["TestCookie"];
?>
```

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 :

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

Pour d'autres informations sur les cookies, jetez un oeil sur http://www.netscape.com/newsref/std/cookie_spec.html .

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.

6.38 Hyperwave

6.38.1 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 serveur.

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 ce 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 fourni 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.

6.38.2 Pré−requis

Cette extension requiert un serveur Hyperwave, téléchargeable sur le site de http://www.hyperwave.com/ .

6.38.3 Installation

6.38.4 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 grandment la configuration. Le concept est très simple. Premièrement, vous avez besoin d'un script PHP qui évalue la variable **`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

Demarrer 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.

6.38.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
hyerwave.allow_persistent	"0"	PHP_INI_SYSTEM
hyperwave.default_port	"418"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.38.6 Types de ressources

Cette extension ne définit aucune ressource.

6.38.7 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.

HW_ATTR_LANG (nombre ˆ virgule flottante)
HW_ATTR_NR (nombre ˆ virgule flottante)
HW_ATTR_NONE (nombre ˆ virgule flottante)

6.38.8 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 object record en un object 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_Children
- hw_ChildrenObj
- hw_Close
- hw_Connect
- hw_Cp
- hw_Deleteobject
- hw_DocByAnchor
- hw_DocByAnchorObj
- hw_Document_Attributes
- hw_Document_BodyTag
- hw_Document_Content
- hw_Document_SetContent
- hw_Document_Size
- hw_EditText
- hw_Error
- hw_ErrorMsg
- hw_Free_Document

- hw_GetAnchors
- hw_GetAnchorsObj
- hw_GetAndLock
- hw_GetChildColl
- hw_GetChildCollObj
- hw_GetChildDocColl
- hw_GetChildDocCollObj
- hw_GetObject
- hw_GetObjectByQuery
- hw_GetObjectByQueryColl
- hw_GetObjectByQueryCollObj
- hw_GetObjectByQueryObj
- hw_GetParents
- hw_GetParentsObj
- hw_GetRemote
- hw_GetRemoteChildren
- hw_GetSrcByDestObj
- hw_GetText
- hw_Username
- hw_Identify
- hw_InCollections
- hw_Info
- hw_InsColl
- hw_InsDoc
- 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_Unlock
- hw_Who

6.38.10 hw_Children() : Liste des object ids des objets fils.

`array` **hw_children** (`resource connection` , `resource objectID`)
[lien librairie]

h`w_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.

6.38.11 hw_ChildrenObj() : Liste des object records des objets fils.

`array` **hw_childrenobj** (`resource connection` , `resource objectID`)
[lien librairie]

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

6.38.12 hw_Close() : Ferme la connexion Hyperwave.

int **hw_close** (resource connection)
[lien librairie]

hw_close retourne FALSE si la connexion n'est pas valide, et sinon, TRUE . Ferme la connexion connection à un serveur Hyperwave.

6.38.13 hw_Connect() : Ouvre une connexion Hyperwave.

resource **hw_connect** (string host , int port , string username , string password)
[lien librairie]

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 .

6.38.14 hw_Cp() : Copie des objets.

resource **hw_cp** (resource connection , array object_id_array , int destination_id)
[lien librairie]

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 .

6.38.15 hw_Deleteobject() : Efface des objets.

int **hw_deleteobject** (resource connection , int object_to_delete)
[lien librairie]

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 .

6.38.16 hw_DocByAnchor() : Identifiant d'objet de l'objet dans l'ancrage.

resource **hw_docbyanchor** (resource connection , int anchorID)
[lien librairie]

hw_docbyanchor retourne l'dentifiant d'objet de l'objet dans l'ancrage anchorID .

6.38.17 hw_DocByAnchorObj() : Attributs de l'objet dans l'ancrage.

string **hw_docbyanchorobj** (resource connection , int anchorID)
[lien librairie]

hw_docbyanchorobj retourne les attributs du document qui correspond à anchorID .

6.38.18 hw_Document_Attributes() : Object record de hw_document.

string **hw_document_attributes** (int hw_document)
[lien librairie]

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.

6.38.19 hw_Document_BodyTag() : Balise de corps d'un document.

string **hw_document_bodytag** (int hw_document)
[lien librairie]

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.

6.38.20 hw_Document_Content() : Contenu d'un document.

string **hw_document_content** (int hw_document)
[lien librairie]

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.

6.38.21 hw_Document_SetContent() : Modifie/remplace le contenu d'un document.

`string` **hw_document_setcontent** (`int hw_document` , `string content`)
[lien librairie]

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.

6.38.22 hw_Document_Size() : Taille d'un document.

`int` **hw_document_size** (`int hw_document`)
[lien librairie]

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.

6.38.23 hw_EditText() : Retourne un document texte.

`int` **hw_edittext** (`resource connection` , `int hw_document`)
[lien librairie]

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_documentbodytag , hw_documentsize , hw_outputdocument et hw_gettext .

6.38.24 hw_Error() : Retourne le code d'erreur.

`int` **hw_error** (`resource connection`)
[lien librairie]

h`w_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.

6.38.25 hw_ErrorMsg() : Retourne un message d'erreur.

`string` **hw_errormsg** (`resource connection`)
[lien librairie]

h`w_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.

6.38.26 hw_Free_Document() : Détruit un document.

`int` **hw_free_document** (`resource hw_document`)
[lien librairie]

h`w_free_document` détruit un document Hyperwave.

6.38.27 hw_GetAnchors() : Identifiants des ancrages d'un document.

`array` **hw_getanchors** (`resource connection` , `resource objectID`)
[lien librairie]

h`w_getanchors` retourne un tableau contenant les identifiants des ancrages du document `objectID` .

6.38.28 hw_GetAnchorsObj() : Attributs des ancrages d'un document.

`array` **hw_getanchorsobj** (`resource connection` , `resource objectID`)
[lien librairie]

h`w_getanchorsobj` retourne un tableau d'attributs des ancrages du document `objectID` .

6.38.29 hw_GetAndLock() : Retourne les attributs, et verrouille l'objet.

`string` **hw_getandlock** (`resource connection` , `resource objectID`)
[lien librairie]

h`w_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 h`w_unlock` et h`w_getobject` .

6.38.30 hw_GetChildColl() : Identifiant d'objet des groupes fils.

`array` **hw_getchildcoll** (`resource connection` , `resource objectID`)
[lien librairie]

`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_getchilddoccoll` .

6.38.31 hw_GetChildCollObj() : object records d'un groupe d'enfants.

`array` **hw_getchildcollobj** (`resource connection` , `resource objectID`)
[lien librairie]

`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` .

6.38.32 hw_GetChildDocColl() : ids des documents fils d'un groupe.

`array` **hw_getchilddoccoll** (`resource connection` , `resource objectID`)
[lien librairie]

`hw_getchilddoccoll` retourne un tableau avec les id des documents fils d'une collection.

Voir aussi `hw_getchildcoll` .

6.38.33 hw_GetChildDocCollObj() : Attributs des documents fils d'un groupe.

`array` **hw_getchilddoccollobj** (`resource connection` , `resource objectID`)
[lien librairie]

`hw_getchilddoccollobj` retourne un tableau contenant les attributs des documents fils du groupe `objectID` .

Voir aussi `hw_childrenobj` et `hw_getchildcollobj` .

6.38.34 hw_GetObject() : Attributs d'un objet.

`array` **hw_getobject** (`resource connection` , `int|array objectID` , `string query`)
[lien librairie]

`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 .

6.38.35 hw_GetObjectByQuery() : Recherche un objet.

array **hw_getobjectbyquery** (resource connection , string query , int max_hits)
[lien librairie]

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 .

6.38.36 hw_GetObjectByQueryColl() : Recherche un objet dans un groupe.

array **hw_getobjectbyquerycoll** (resource connection , resource objectID , string query , int max_hits)
[lien librairie]

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 .

6.38.37 hw_GetObjectByQueryCollObj() : Recherche un objet dans un groupe.

array **hw_getobjectbyquerycollobj** (resource connection , resource objectID , string query , int max_hits)
[lien librairie]

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 .

6.38.38 hw_GetObjectByQueryObj() : Recherche un objet.

array **hw_getobjectbyqueryobj** (resource connection , string query , int max_hits)
[lien librairie]

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 .

6.38.39 hw_GetParents() : Identifiant d'objet des parents.

array **hw_getparents** (resource connection , resource objectID)
[lien librairie]

hw_getparents retourne un tableau indexé avec les identifiants des objets parents de objectID .

6.38.40 hw_GetParentsObj() : Attributs des parents.

array **hw_getparentsobj** (resource connection , resource objectID)
[lien librairie]

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 .

6.38.41 hw_GetRemote() : Retourne un document distant.

`resource` **hw_getremote** (`resource connection` , `resource objectID`)
[lien librairie]

`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` .

6.38.42 hw_GetRemoteChildren() : Retourne les fils d'un document distant.

`resource` **hw_getremotechildren** (`resource connection` , `string object_record`)
[lien librairie]

`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` .

6.38.43 hw_GetSrcByDestObj() : Retourne les ancrages qui pointent sur un objet.

`array` **hw_getsrcbydestobj** (`resource connection` , `resource objectID`)
[lien librairie]

`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` .

6.38.44 hw_GetText() : Retourne un document texte.

`int` **hw_gettext** (`resource connection` , `resource objectID` , *`mixed rootID/prefix`*)
[lien librairie]

h`w_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 <A HREF="/internet_movie">. La position réelle de la source et de la cible dans la hiérarchie seront ignorés. Vous devrez modificer 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 prefixe. 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.

h`w_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 h`w_pipedocument` , `hw_free_document` , `hw_documentbodytag` , `hw_documentsize` et `hw_outputdocument` .

6.38.45 hw_Username() : Nom de l'utilisateur actuellement identifié.

`string` **hw_getusername** (`resource connection`)
[lien librairie]

h`w_getusername` retourne le nom d'utilisateur utilisé par la connexion.

6.38.46 hw_Identify() : Identifie un utilisateur.

`int` **hw_identify** (`string username` , `string password`)
[lien librairie]

h`w_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 h`w_connect` .

6.38.47 hw_InCollections() : Vérifie qu'un identifiant d'objet est dans un groupe.

`array` **hw_incollections** (`resource connection` , `array object_id_array` , `array collection_id_array` , `int return_collections`)
[lien librairie]

h`w_incollections` vérifie qu'un ensemble d'objets (documents ou groupes) donnés par `object_id_array` fait partie des groupe listés par `object_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.

6.38.48 hw_Info() : Informations à propos d'une connexion.

`string` **hw_info** (`resource connection`)
[lien librairie]

h`w_info` retourne les informations de la connexion courante. La chaîne retournée a le format suivant : <Serverstring>, <Host>, <Port>, <Username>, <Port of Client>, <Byte swapping>

6.38.49 hw_InsColl() : Insère un groupe.

`int` **hw_inscoll** (`resource connection` , `resource objectID` , `array object_array`)
[lien librairie]

h`w_inscoll` insère un nouveau groupe, avec les attributs `object_array` dans le groupe `objectID` .

6.38.50 hw_InsDoc() : Insère un document.

`int` **hw_insdoc** (`resource connection` , `int parentID` , `string object_record` , `string text`)
[lien librairie]

h`w_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 h`w_insertdocument` .

Voir aussi h`w_insertdocument` et h`w_inscoll` .

6.38.51 hw_InsertDocument() : Insère un document dans un groupe.

`int` **hw_insertdocument** (`resource connection` , `int parent_id` , `int hw_document`)
[lien librairie]

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 .

6.38.52 hw_InsertObject() : Insère un object record.

int **hw_insertobject** (resource connection , string object_rec , string parameter)
[lien librairie]

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 .

6.38.53 hw_mapid() : Représente un id globale en un id virtuel local.

int **hw_mapid** (resource connection , int server_id , int object_id)
[lien librairie]

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

6.38.54 hw_Modifyobject() : Modifie les attributs d'objet record.

int **hw_modifyobject** (resource connection , int object_to_change , array remove , array add , int mode)
[lien librairie]

hw_modifyobject permet d'effacer, d'ajouter ou de modifier les attributs d'un objet. L'objet est reperé 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 dont 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 modifié.

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 échouerai de toute manière, car cet attribut n'existe pas. L'ajout de cet attribut échouerai 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 deviendrai 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 certains 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 language, le préfixe 'xx' sera assigné.

Note

L'attribut 'Name' est un peu particuluer. Dans certains cas, il ne peut pas être complétement 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.

6.38.55 hw_Mv() : Déplace un objet.

`int` **hw_mv** (`resource connection` , `array object_id_array` , `int source_id` , `int destination_id`)
[lien librairie]

`hw_mv` déplaces 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` .

6.38.56 hw_New_Document() : Crée un nouveau document.

`int` **hw_new_document** (`string object_record` , `string document_data` , `int document_size`)
[lien librairie]

h`w_new_document` retourne un nouveau document Hyperwave avec comme données `document_data` et comme attributs `object_record` . La longueur de `document_data` doit être données dans `document_size` . Cette fonction n'insère pas l'objet dans le serveur Hyperwave.

Voir aussi h`w_free_document` , `hw_documentsize` , `hw_documentbodytag` , `hw_outputdocument` et h`w_insertdocument` .

6.38.57 hw_Objrec2Array() : Convertit les attributs d'un objet en tableau.

`array` **hw_objrec2array** (`string object_record`)
[lien librairie]

h`w_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. Actuellement, seuls les attributs 'Title', 'Description' et 'Keyword' sont traités correctement.

Voir aussi h`w_array2objrec` .

6.38.58 hw_Output_Document() : Affiche hw_document.

`int` **hw_output_document** (`int hw_document`)
[lien librairie]

h`w_output_document` affiche h`w_document` sans la balise BODY.

Pour la compatibilité ascendente, h`w_outputdocument` est aussi accepté, mais il est déconseillé.

6.38.59 hw_pConnect() : Crée une connexion persistante.

`int` **hw_pconnect** (`string host` , `int port` , `string username` , `string password`)
[lien librairie]

h`w_pconnect` retourne un index de connexion en cas de succès, et FALSE si la connexion n'a pas pu être créée. h`w_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 h`w_connect` .

6.38.60 hw_PipeDocument() : Retourne un document.

`int` **hw_pipedocument** (`resource connection` , `resource objectID`)
[lien librairie]

`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` (insertions), `hw_free_document` , `hw_documentsize` , `hw_documentbodytag` et `hw_outputdocument` .

6.38.61 hw_Root() : Object id de la racine.

`int` **hw_root** (`void`)
[lien librairie]

`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.

6.38.62 hw_Unlock() : Déverrouille un objet.

`int` **hw_unlock** (`resource connection` , `resource objectID`)
[lien librairie]

`hw_unlock` déverrouille un document, et laisse l'accès aux autres utilisateurs.

Voir aussi `hw_getandlock` .

6.38.63 hw_Who() : Liste des utilisateurs actuellement identifiés.

`int` **hw_who** (`resource connection`)
[lien librairie]

`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.

6.39 Hyperwave API

6.39.1 Introduction

Hyperwave a été développé à l' IICM à Graz. Le projet a commencé avec le nom de HYPER−G , puis il est devenu Hyperwave, lors de sa commercialisation en 1996.

Hyperwave n'est pas un logiciel libre. Sa version courante, la 5.5, est disponible à http://www.hyperwave.com/ . Une version limitée dans le temps est commandable gratuitement (30 jours).

Voir aussi le module Hyperwave .

Hyperwave est un système d'information comparable à une base de données (HIS , Hyperwave Information Server). Il se concentre sur le stockage et la gestion de documents. Un document peut être n'importe quelle pièce de données qui soit stockable dans un fichier. Chaque document est accompagné de son propre enregistrement. L'enregistrement contient les méta−données du document. Les 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.

6.39.2 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.

6.39.3 Installation

Après avoir installé le SDK Hyperwave, configurez PHP avec `−−with−hwapi=<dir$gt;` .

6.39.4 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.

6.39.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
hwapi.allow_persistent	"0"	PHP_INI_SYSTEM

6.39.6 Types de ressources

Cette extension ne définit aucune ressource.

6.39.7 Constantes Prédefinies

Cette extension ne définit aucune constante.

6.39.8 Classes

L'API fournie avec l'extension HW_API est complétement 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`

Certains 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", "0x873A8768 0x00000002".
- `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->setcommitedversion
- hw_api->srcanchors
- hw_api->srcsofdst
- hw_api->unlock
- hw_api->user
- hw_api->userlist

6.39.10 hw_api_attribute−>langdepvalue() : Returns value for a given language

`string` **langdepvalue** (`string language`)
[lien librairie]

Returns the value in the given language of the attribute.

See also `hwapi_attribute_value` .

6.39.11 hw_api_attribute−>value() : Returns value of the attribute

`string` **value** (`void`)
[lien librairie]

Returns the value of the attribute.

See also `hwapi_attribute_key` , `hwapi_attribute_values` .

6.39.12 hw_api_attribute−>values() : Returns all values of the attribute

`array` **values** (`void`)
[lien librairie]

Returns all values of the attribute as an array of strings.

See also `hwapi_attribute_value` .

6.39.13 hw_api_attribute() : Creates instance of class hw_api_attribute

`object` **attribute** (*`string name`* , *`string value`*)
[lien librairie]

Creates a new instance of hw_api_attribute with the given name and value.

6.39.14 hw_api−>checkin() : Checks in an object

`object` **checkin** (`array parameter`)
[lien librairie]

This function checks in an object or a whole hiearchie of objects. The parameters array contains the required element 'objectIdentifier' and the optional element 'version', 'comment', 'mode' and 'objectQuery'. 'version' sets the version of the object. It consists of the major and minor version separated by a period. If the version is not set, the minor version is incremented. 'mode' can be one of the following values:

HW_API_CHECKIN_NORMAL
: Checks in and commits the object. The object must be a document.

HW_API_CHECKIN_RECURSIVE

If the object to check in is a collection, all children will be checked in recursively if they are documents. Trying to check in a collection would result in an error.

HW_API_CHECKIN_FORCE_VERSION_CONTROL
Checks in an object even if it is not under version control.

HW_API_CHECKIN_REVERT_IF_NOT_CHANGED
Check if the new version is different from the last version. Unless this is the case the object will be checked in.

HW_API_CHECKIN_KEEP_TIME_MODIFIED
Keeps the time modified from the most recent object.

HW_API_CHECKIN_NO_AUTO_COMMIT
The object is not automatically commited on checkin.

See also `hwapi_checkout` .

6.39.15 hw_api−>checkout() : Checks out an object

`object` **checkout** `( array parameter )`
[lien librairie]

This function checks out an object or a whole hiearchie of objects. The parameters array contains the required element 'objectIdentifier' and the optional element 'version', 'mode' and 'objectQuery'. 'mode' can be one of the following values:

HW_API_CHECKIN_NORMAL
Checks out an object. The object must be a document.

HW_API_CHECKIN_RECURSIVE
If the object to check out is a collection, all children will be checked out recursively if they are documents. Trying to check out a collection would result in an error.

See also `hwapi_checkin` .

6.39.16 hw_api−>children() : Returns children of an object

`array` **children** `( array parameter )`
[lien librairie]

Retrieves the children of a collection or the attributes of a document. The children can be further filtered by specifying an object query. The parameter array contains the required elements 'objectIdentifier' and the optional elements 'attributeSelector' and 'objectQuery'.

The return value is an array of objects of type `HW_API_Object` or `HW_API_Error` .

See also `hwapi_parents` .

6.39.17 hw_api_content−>mimetype() : Returns mimetype

`string` **mimetype** `( void )`
[lien librairie]

Returns the mimetype of the content.

6.39.18 hw_api_content−>read() : Read content

`string` **read** (`string buffer` , `integer len`)
[lien librairie]

Reads `len` bytes from the content into the given buffer.

6.39.19 hw_api−>content() : Returns content of an object

`object` **content** (`array parameter`)
[lien librairie]

This function returns the content of a document as an object of type h`w_api_content` . The parameter array contains the required elements 'objectidentifier' and the optional element 'mode'. The mode can be one of the constants HW_API_CONTENT_ALLLINKS, HW_API_CONTENT_REACHABLELINKS or HW_API_CONTENT_PLAIN. HW_API_CONTENT_ALLLINKS means to insert all anchors even if the destination is not reachable. HW_API_CONTENT_REACHABLELINKS tells h`w_api_content` to insert only reachable links and HW_API_CONTENT_PLAIN will lead to document without any links.

6.39.20 hw_api−>copy() : Copies physically

`object` **copy** (`array parameter`)
[lien librairie]

This function will make a physical copy including the content if it exists and returns the new object or an error object. The parameter array contains the required elements 'objectIdentifier' and 'destinationParentIdentifier'. The optional parameter is 'attributeSelector'`

See also `hwapi_move` , `hwapi_link` .

6.39.21 hw_api−>dbstat() : Returns statistics about database server

`object` **dbstat** (`array parameter`)
[lien librairie]

See also `hwapi_dcstat` , `hwapi_hwstat` , `hwapi_ftstat` .

6.39.22 hw_api−>dcstat() : Returns statistics about document cache server

`object` **dcstat** (`array parameter`)
[lien librairie]

See also `hwapi_hwstat` , `hwapi_dbstat` , `hwapi_ftstat` .

6.39.23 hw_api−>dstanchors() : Returns a list of all destination anchors

`object` **dstanchors** `( array parameter )`
[lien librairie]

Retrieves all destination anchors of an object. The parameter array contains the required element 'objectIdentifier' and the optional elements 'attributeSelector' and 'objectQuery'.

See also `hwapi_srcanchors` .

6.39.24 hw_api−>dstofsrcanchors() : Returns destination of a source anchor

`object` **dstofsrcanchors** `( array parameter )`
[lien librairie]

Retrieves the destination object pointed by the specified source anchors. The destination object can either be a destination anchor or a whole document. The parameters array contains the required element 'objectIdentifier' and the optional element 'attributeSelector'.

See also `hwapi_srcanchors` , `hwapi_dstanchors` , `hwapi_objectbyanchor` .

6.39.25 hw_api_error−>count() : Returns number of reasons

`int` **count** `( void )`
[lien librairie]

Returns the number of error reasons.

See also `hwapi_error_reason` .

6.39.26 hw_api_error−>reason() : Returns reason of error

`object` **reason** `( void )`
[lien librairie]

Returns the first error reason.

See also `hwapi_error_count` .

6.39.27 hw_api−>find() : Search for objects

`array` **find** `( array parameter )`
[lien librairie]

This functions searches for objects either by executing a key or/and full text query. The found objects can further be filtered by an optional object query. They are sorted by their importance. The second search operation is relatively slow and its result can be limited to a certain number of hits. This allows to perform an incremental search, each returning just a subset of all found documents,

starting at a given index. The parameter array contains the 'keyquery' or/and 'fulltextquery' depending on who you would like to search. Optional parameters are 'objectquery', 'scope', 'lanugages' and 'attributeselector'. In case of an incremental search the optional parameters 'startIndex', numberOfObjectsToGet' and 'exactMatchUnit' can be passed.

6.39.28 hw_api−>ftstat() : Returns statistics about fulltext server

`object` **ftstat** (`array parameter`)
[lien librairie]

See also `hwapi_dcstat` , `hwapi_dbstat` , `hwapi_hwstat` .

6.39.29 hwapi_hgcsp() : Returns object of class hw_api

`object` **hwapi_hgcsp** (`string hostname` , *`int port`*)
[lien librairie]

Opens a connection to the Hyperwave server on host `hostname` . The protocol used is HGCSP. If you do not pass a port number, 418 is used.

See also `hwapi_hwtp` .

6.39.30 hw_api−>hwstat() : Returns statistics about Hyperwave server

`object` **hwstat** (`array parameter`)
[lien librairie]

See also `hwapi_dcstat` , `hwapi_dbstat` , `hwapi_ftstat` .

6.39.31 hw_api−>identify() : Log into Hyperwave Server

`object` **identify** (`array parameter`)
[lien librairie]

Logs into the Hyperwave Server. The parameter array must contain the elements 'username' und 'password'.

The return value will be an object of type `HW_API_Error` if identification failed or TRUE if it was successful.

6.39.32 hw_api−>info() : Returns information about server configuration

`object` **info** (`array parameter`)
[lien librairie]

See also `hwapi_dcstat` , `hwapi_dbstat` , `hwapi_ftstat` , `hwapi_hwstat` .

6.39.33 hw_api−>insert() : Inserts a new object

`object` **insert** (`array parameter`)
[lien librairie]

Insert a new object. The object type can be user, group, document or anchor. Depending on the type other object attributes has to be set. The parameter array contains the required elements 'object' and 'content' (if the object is a document) and the optional parameters 'parameters', 'mode' and 'attributeSelector'. The 'object' must contain all attributes of the object. 'parameters' is an object as well holding futher attributes like the destination (attribute key is 'Parent'). 'content' is the content of the document. 'mode' can be a combination of the following flags:

HW_API_INSERT_NORMAL
The object in inserted into the server.
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

See also `hwapi_replace` .

6.39.34 hw_api−>insertanchor() : Inserts a new object of type anchor

`object` **insertanchor** (`array parameter`)
[lien librairie]

This function is a shortcut for h`wapi_insert` . It inserts an object of type anchor and sets some of the attributes required for an anchor. The parameter array contains the required elements 'object' and 'documentIdentifier' and the optional elements 'destinationIdentifier', 'parameter', 'hint' and 'attributeSelector'. The 'documentIdentifier' specifies the document where the anchor shall be inserted. The target of the anchor is set in 'destinationIdentifier' if it already exists. If the target does not exists the element 'hint' has to be set to the name of object which is supposed to be inserted later. Once it is inserted the anchor target is resolved automatically.

See also `hwapi_insertdocument` , `hwapi_insertcollection` , `hwapi_insert` .

6.39.35 hw_api−>insertcollection() : Inserts a new object of type collection

`object` **insertcollection** (`array parameter`)
[lien librairie]

This function is a shortcut for h`wapi_insert` . It inserts an object of type collection and sets some of the attributes required for a collection. The parameter array contains the required elements 'object' and 'parentIdentifier' and the optional elements 'parameter' and 'attributeSelector'. See `hwapi_insert` for the meaning of each element.

See also `hwapi_insertdocument` , `hwapi_insertanchor` , `hwapi_insert` .

6.39.36 hw_api−>insertdocument() : Inserts a new object of type document

`object` **insertdocument** `( array parameter )`
[lien librairie]

This function is a shortcut for `hwapi_insert` . It inserts an object with content and sets some of the attributes required for a document. The parameter array contains the required elements 'object', 'parentIdentifier' and 'content' and the optional elements 'mode', 'parameter' and 'attributeSelector'. See `hwapi_insert` for the meaning of each element.

See also `hwapi_insert` `hwapi_insertanchor` , `hwapi_insertcollection` .

6.39.37 hw_api−>link() : Creates a link to an object

`object` **link** `( array parameter )`
[lien librairie]

Creates a link to an object. Accessing this link is like accessing the object to links points to. The parameter array contains the required elements 'objectIdentifier' and 'destinationParentIdentifier'. 'destinationParentIdentifier' is the target collection.

The function returns true on success or an error object.

See also `hwapi_copy` .

6.39.38 hw_api−>lock() : Locks an object

`object` **lock** `( array parameter )`
[lien librairie]

Locks an object for exclusive editing by the user calling this function. The object can be only unlocked by this user or the sytem user. The parameter array contains the required element 'objectIdentifier' and the optional parameters 'mode' and 'objectquery'. 'mode' determines how an object is locked. HW_API_LOCK_NORMAL means, an object is locked until it is unlocked. HW_API_LOCK_RECURSIVE is only valid for collection and locks all objects within th collection und possible subcollections. HW_API_LOCK_SESSION means, an object is locked only as long as the session is valid.

See also `hwapi_unlock` .

6.39.39 hw_api−>move() : Moves object between collections

`object` **move** `( array parameter )`
[lien librairie]

See also `hw_objrec2array` .

6.39.40 hw_api_content() : Create new instance of class hw_api_content

`string` **content** (`string content` , `string mimetype`)
[lien librairie]

Creates a new content object from the string `content` . The mimetype is set to `mimetype` .

6.39.41 hw_api_object−>assign() : Clones object

`object` **assign** (`array parameter`)
[lien librairie]

Clones the attributes of an object.

6.39.42 hw_api_object−>attreditable() : Checks whether an attribute is editable

`bool` **attreditable** (`array parameter`)
[lien librairie]

6.39.43 hw_api_object−>count() : Returns number of attributes

`int` **count** (`array parameter`)
[lien librairie]

6.39.44 hw_api_object−>insert() : Inserts new attribute

`bool` **insert** (`object attribute`)
[lien librairie]

Adds an attribute to the object. Returns true on success and otherwise false.

See also `hwapi_object_remove` .

6.39.45 hw_api_object() : Creates a new instance of class hw_api_object

`object` **hw_api_object** (`array parameter`)
[lien librairie]

See also `hwapi_lock` .

6.39.46 hw_api_object−>remove() : Removes attribute

`bool` **remove** (`string name`)
[lien librairie]

Removes the attribute with the given name. Returns true on success and otherwise false.

See also `hwapi_object_insert` .

6.39.47 hw_api_object−>title() : Returns the title attribute

`string` **title** `( array parameter )`
[lien librairie]

6.39.48 hw_api_object−>value() : Returns value of attribute

`string` **value** `( string name )`
[lien librairie]

Returns the value of the attribute with the given name or false if an error occured.

6.39.49 hw_api−>object() : Retrieve attribute information

`object` **hw_api−>object** `( array parameter )`
[lien librairie]

This function retrieves the attribute information of an object of any version. It will not return the document content. The parameter array contains the required elements 'objectIdentifier' and the optional elements 'attributeSelector' and 'version'.

The returned object is an instance of class `HW_API_Object` on success or `HW_API_Error` if an error occured.

This simple example retrieves an object and checks for errors.

Retrieve an object

```
<?php
function handle_error($error) {
  $reason = $error->reason(0);
  echo "Type: <B>";
  switch($reason->type()) {
    case 0:
      echo "Error";
      break;
    case 1:
      echo "Warning";
      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]);
$parms = array("objectIdentifier"=>"rootcollection, "attributeSelector"=>array("Title", "Name", "Doc
$root = $hwapi->object($parms);
if(get_class($root) == "HW_API_Error") {
  handle_error($root);
  exit;
}
list_attr($root);
?>
```

See also `hwapi_content` .

6.39.50 hw_api->objectbyanchor() : Returns the object an anchor belongs to

`object` **objectbyanchor** (`array parameter`)
[lien librairie]

This function retrieves an object the specified anchor belongs to. The parameter array contains the required element 'objectIdentifier' and the optional element 'attributeSelector'.

See also `hwapi_dstofsrcanchor` , `hwapi_srcanchors` , `hwapi_dstanchors` .

6.39.51 hw_api->parents() : Returns parents of an object

`array` **parents** (`array parameter`)
[lien librairie]

Retrieves the parents of an object. The parents can be further filtered by specifying an object query. The parameter array contains the required elements 'objectidentifier' and the optional elements 'attributeselector' and 'objectquery'.

The return value is an array of objects of type `HW_API_Object` or `HW_API_Error` .

See also `hwapi_children` .

6.39.52 hw_api_reason->description() : Returns description of reason

`string` **description** (`void`)
[lien librairie]

Returns the description of a reason

6.39.53 hw_api_reason−>type() : Returns type of reason

`object` **type** `( void )`
[lien librairie]

Returns the type of a reason.

6.39.54 hw_api−>remove() : Delete an object

`object` **remove** `( array parameter )`
[lien librairie]

This function removes an object from the specified parent. Collections will be removed recursively. You can pass an optional object query to remove only those objects which match the query. An object will be deleted physically if it is the last instance. The parameter array contains the required elements 'objectidentifier' and 'parentidentifier'. If you want to remove a user or group 'parentidentifier' can be skiped. The optional parameter 'mode' determines how the deletion is performed. In normal mode the object will not be removed physically until all instances are removed. In physical mode all instances of the object will be deleted imediately. In removelinks mode all references to and from the objects will be deleted as well. In nonrecursive the deletion is not performed recursive. Removing a collection which is not empty will cause an error.

See also `hwapi_move` .

6.39.55 hw_api−>replace() : Replaces an object

`object` **replace** `( array parameter )`
[lien librairie]

Replaces the attributes and the content of an object The parameter array contains the required elements 'objectIdentifier' and 'object' and the optional parameters 'content', 'parameters', 'mode' and 'attributeSelector'. 'objectIdentifier' contains the object to be replaced. 'object' contains the new object. 'content' contains the new content. 'parameters' contain extra information for HTML documents. HTML_Language is the letter abbreviation of the language of the title. HTML_Base sets the base attribute of the HTML document. 'mode' can be a combination of the following flags:

HW_API_REPLACE_NORMAL
: The object on the server is replace with the object passed.

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

See also `hwapi_insert` .

6.39.56 hw_api−>setcommitedversion() : Commits version other than last version

`object` **setcommitedversion** `( array parameter )`
[lien librairie]

Commits a version of a docuemnt. The commited version is the one which is visible to users with read access. By default the last version is the commited version.

See also `hwapi_checkin` , `hwapi_checkout` , `hwapi_revert` .

6.39.57 hw_api−>srcanchors() : Returns a list of all source anchors

`object` **srcanchors** `( array parameter )`
[lien librairie]

Retrieves all source anchors of an object. The parameter array contains the required element 'objectIdentifier' and the optional elements 'attributeSelector' and 'objectQuery'.

See also `hwapi_dstanchors` .

6.39.58 hw_api−>srcsofdst() : Returns source of a destination object

`object` **srcsofdst** `( array parameter )`
[lien librairie]

Retrieves all the source anchors pointing to the specified destination. The destination object can either be a destination anchor or a whole document. The parameters array contains the required element 'objectIdentifier' and the optional element 'attributeSelector' and 'objectQuery'. The function returns an array of objects or an error.

See also `hwapi_dstofsrcanchor` .

6.39.59 hw_api−>unlock() : Unlocks a locked object

`object` **unlock** `( array parameter )`
[lien librairie]

Unlocks a locked object. Only the user who has locked the object and the system user may unlock an object. The parameter array contains the required element 'objectIdentifier' and the optional parameters 'mode' and 'objectquery'. The meaning of 'mode' is the same as in function `hwapi_lock` .

Returns true on success or an object of class HW_API_Error.

See also `hwapi_lock` .

6.39.60 hw_api−>user() : Returns the own user object

`object` **user** `( array parameter )`
[lien librairie]

See also `hwapi_userlist` .

6.39.61 hw_api−>userlist() : Returns a list of all logged in users

`object` **userlist** `( array parameter )`
[lien librairie]

See also `hwapi_user` .

6.40 Iconv

6.40.1 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.

6.40.2 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/ .

6.40.3 Installation

Pour pouvoir l'utiliser, vous devez compiler PHP avec l'option `−−with−iconv` .

6.40.4 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` .

6.40.5 Types de ressources

Cette extension ne définit aucune ressource.

6.40.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

6.40.7 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

6.40.9 iconv_set_encoding() : Modifie le jeu courant de caractères courant

`array` **iconv_set_encoding** (`string type` , `string charset`)
[lien librairie]

`iconv_set_encoding` modifie le jeu de caractères courant, et remplace la valeur courante du paramétre `type` par `charset` et TRUE en cas de succès et FALSE , en cas d'échec.

Exemple `iconv_set_encoding`

```
<?php
  iconv_set_encoding("internal_encoding", "UTF-8");
  iconv_set_encoding("output_encoding", "ISO-8859-1");
?>
```

Voir aussi `iconv_get_encoding` et `ob_iconv_handler` .

6.40.10 iconv() : Convertit une chaîne dans un jeu de caractères

`string` **iconv** (`string in_charset` , `string out_charset` , `string str`)
[lien librairie]

`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.");
?>
```

6.40.11 ob_iconv_handler() : Gestionnaire de sortie pour maitriser le jeu de caractères de sortie

`array` **ob_iconv_handler** (`string contents` , `int status`)
[lien librairie]

`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.

Exemple avec `ob_iconv_handler`

```
<?php
  ob_start("ob_iconv_handler"); // start output buffering
?>
```

Voir aussi `iconv_get_encoding` et `iconv_set_encoding` .

6.41 Images

6.41.1 Introduction

PHP ne se limite pas à la génération de pages HTML. Il peut aussi servir à créer des 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 librarie GD. GD et PHP auront aussi besoin de 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 sections sur les pré−requis pour savoir comment améliorer les possibilités le lecture, modification et écriture d'images, ainsi que les possibilités de lecture d'images prises par des appareils numériques.

6.41.2 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 à 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.

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 méta données 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.

6.41.3 Installation

Pour accéder aux images en `JPEG` , vous devez installer la librairie jpeg−6b (disponible à ftp://ftp.uu.net/graphics/jpeg/), puis, recompiler GD pour qu'elle utilise jpeg−6b. Vous devrez aussi compiler PHP avec `−−with−jpeg−dir=/path/to/jpeg−6b` .

Pour ajouter le support des polices Type 1, vous devez installer t1lib (disponible à ftp://sunsite.unc.edu/pub/Linux/libs/graphics/), puis ajouter l'option `−−with−t1lib[=dir]` .

6.41.4 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
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

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les 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 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` .

6.41.5 Types de ressources

Cette extension ne définit aucune ressource.

6.41.6 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 (nombre ˆ virgule flottante)
IMG_JPG (nombre ˆ virgule flottante)
IMG_JPEG (nombre ˆ virgule flottante)
IMG_PNG (nombre ˆ virgule flottante)
IMG_WBMP (nombre ˆ virgule flottante)
IMG_XPM (nombre ˆ virgule flottante)
IMG_COLOR_TILED (nombre ˆ virgule flottante)
IMG_COLOR_STYLED (nombre ˆ virgule flottante)
IMG_COLOR_BRUSHED (nombre ˆ virgule flottante)
IMG_COLOR_STYLEDBRUSHED (nombre ˆ virgule flottante)
IMG_COLOR_TRANSPARENT (nombre ˆ virgule flottante)
IMG_ARC_ROUNDED (nombre ˆ virgule flottante)
IMG_ARC_PIE (nombre ˆ virgule flottante)
IMG_ARC_CHORD (nombre ˆ virgule flottante)
IMG_ARC_NOFILL (nombre ˆ virgule flottante)
IMG_ARC_EDGED (nombre ˆ virgule flottante)
IMAGETYPE_GIF (nombre ˆ virgule flottante)
IMAGETYPE_JPG (nombre ˆ virgule flottante)
IMAGETYPE_PNG (nombre ˆ virgule flottante)
IMAGETYPE_SWF (nombre ˆ virgule flottante)
IMAGETYPE_PSD (nombre ˆ virgule flottante)
IMAGETYPE_BMP (nombre ˆ virgule flottante)
IMAGETYPE_TIFF_II (nombre ˆ virgule flottante)
IMAGETYPE_TIFF_MM (nombre ˆ virgule flottante)
IMAGETYPE_JPC (nombre ˆ virgule flottante)
IMAGETYPE_JP2 (nombre ˆ virgule flottante)
IMAGETYPE_JPX (nombre ˆ virgule flottante)
IMAGETYPE_SWC (nombre ˆ virgule flottante)

6.41.7 Exemples

Création d'une image PNG avec PHP

```
<?php
    header("Content-type: image/png");
    $string = $_GET['text'];
    $im     = imagecreatefrompng("images/button1.png");
    $orange = imagecolorallocate($im, 220, 210, 60);
    $px     = (imagesx($im) - 7.5 * strlen($string)) / 2;
    imagestring($im, 3, $px, 9, $string, $orange);
    imagepng($im);
    imagedestroy($im);
?>
```

Cet exemple doit être appelé depuis une page HTML avec une balise image telle que : `<img src="button.php?text">` . Le script ci−dessus, button.php, prend la chaîne "text" et l'inscrit sur le fond d'image appelée "images/button1.png" et l'affiche. C'est une méthode très pratique pour éviter de redessiner un nouveau bouteau, dès que vous changez le texte d'un bouton. De cette façon, les boutons sont dynamiquement générés.

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- Exemples
- getimagesize
- image2wbmp
- ImageAlphaBlending
- ImageArc
- ImageChar
- ImageCharUp
- ImageColorAllocate
- ImageColorAt
- ImageColorClosest
- ImageColorClosestAlpha
- ImageColorDeAllocate
- ImageColorExact
- ImageColorExactAlpha
- ImageColorResolve
- ImageColorResolveAlpha
- ImageColorSet
- ImageColorsForIndex
- ImageColorsTotal
- ImageColorTransparent
- ImageCopy
- ImageCopyMerge
- ImageCopyMergeGray
- ImageCopyResampled
- ImageCopyResized
- ImageCreate
- imagecreatefromgif
- ImageCreateFromJPEG
- ImageCreateFromPNG
- ImageCreateFromString
- ImageCreateFromWBMP
- ImageCreateFromXBM
- ImageCreateFromXPM
- ImageCreateTrueColor
- ImageDashedLine
- ImageDestroy
- ImageEllipse
- ImageFill
- imagefilledarc
- ImageFilledEllipse

- ImageFilledPolygon
- ImageFilledRectangle
- ImageFillToBorder
- ImageFontHeight
- ImageFontWidth
- ImageGammaCorrect
- ImageGif
- ImageInterlace
- ImageJPEG
- ImageLine
- ImageLoadFont
- ImagePaletteCopy
- ImagePNG
- ImagePolygon
- ImagePSBBox
- imagepscopyfont
- ImagePSEncodeFont
- ImagePsExtendFont
- ImagePSFreeFont
- ImagePSLoadFont
- ImagePsSlantFont
- ImagePSText
- ImageRectangle
- imagesetbrush
- ImageSetPixel
- ImageSetThickness
- ImageSetTile
- ImageString
- ImageStringUp
- ImageSX
- ImageSY
- ImageTrueColorToPalette
- ImageTTFBBox
- ImageTTFText
- ImageTypes
- ImageWBMP
- iptcparse
- JPEG2WBMP
- PNG2WBMP
- read_exif_data

6.41.9 image2wbmp() : Crée une image WBMP

`int` **image2wbmp** (`int im` , *`string filename`* , *`int threshold`*)
[lien librairie]

`image2wbmp` crée une image `WBMP` à partir de l'image `im` . Si le paramètre `filename` est fourni, l'image sera créée dans ce fichier, et sinon, elle sera envoyée au navigateur. `im` 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 entête HTTP Content−Type : `IMAGE/VND.WAP.WBMP` avec la fonction `header` , vous pouvez creer 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` .

6.41.10 ImageAlphaBlending() : Modifie le mode de blending d'une image

`int` **imagealphablending** (`resource im` , `boolean blendmode`)
[lien librairie]

`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.

6.41.11 ImageArc() : Dessine une ellipse partielle.

`int` **imagearc** (`resource im` , `int cx` , `int cy` , `int w` , `int h` , `int s` , `int e` , `int col`)
[lien librairie]

`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 `im` . `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` .

6.41.12 ImageChar() : Dessine un caractère horizontalement.

`int` **imagechar** (`resource im` , `resource font` , `int x` , `int y` , `string c` , `int col`)
[lien librairie]

`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 `col` . 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` .

6.41.13 ImageCharUp() : Dessine un caractère verticalement.

`int` **imagecharup** (`resource im` , `resource font` , `int x` , `int y` , `string c` , `int col`)

[lien librairie]

`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 `col` . 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` .

6.41.14 ImageColorAllocate() : Alloue une couleur pour une image.

`int` **imagecolorallocate** (`resource im` , `int red` , `int green` , `int blue`)
[lien librairie]

`imagecolorallocate` retourne un identifiant de couleur, représentant la couleur composée avec les couleurs RGB (`red` , `green` , `blue`). L'argument `im` est le résultat de la fonction `imagecreate` . `imagecolorallocate` doit être appelée pour créer chaque couleur qui sera représentée par `im` .

```
<?php
  $white = imagecolorallocate($im, 255,255,255);
  $black = imagecolorallocate($im, 0,0,0);
?>
```

6.41.15 ImageColorAt() : Retourne l'index de la couleur d'un pixel donné.

`int` **imagecolorat** (`resource im` , `int x` , `int y`)
[lien librairie]

`imagecolorat` retourne l'index de la couleur du pixel situé aux coordonnées (`x` , `y`), dans l'image `im` .

Voir aussi `imagecolorset` et `imagecolorsforindex` .

6.41.16 ImageColorClosest() : Retourne l'index de la couleur la plus proche d'une couleur donnée.

`int` **imagecolorclosest** (`resource im` , `int red` , `int green` , `int blue`)
[lien librairie]

`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` .

6.41.17 ImageColorClosestAlpha() : Retourne la couleur la plus proche, en tenant compte du canal alpha

`int` **imagecolorclosestalpha** (`resource im` , `int red` , `int green` , `int blue` , `int alpha`)
[lien librairie]

`imagecolorclosestalpha` retourne l'index de la couleur, dans la palette de l'image `im` , 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.

6.41.18 ImageColorDeAllocate() : Désalloue une couleur pour une image

`int` **imagecolordeallocate** (`resource im` , `int index`)
[lien librairie]

`imagecolordeallocate` désalloue une couleur précédemment allouée avec la fonction `imagecolorallocate` .

```
<?php
  $white = imagecolorallocate($im, 255, 255, 255);
  imagecolordeallocate($im, $white);
?>
```

6.41.19 ImageColorExact() : Retourne l'index de la couleur donnée.

`int` **imagecolorexact** (`resource im` , `int red` , `int green` , `int blue`)
[lien librairie]

`imagecolorexact` retourne l'index de la couleur spécifiée dans la palette de l'image `im` .

Si la couleur n'existe pas dans cette palette, `imagecolorexact` retourne −1.

Voir aussi `imagecolorclosest` .

6.41.20 ImageColorExactAlpha() : Retourne l'index d'une couleur avec son canal alpha

`int` **imagecolorexactalpha** (`resource im` , `int red` , `int green` , `int blue` , `int alpha`)
[lien librairie]

imagecolorexactalpha retourne l'index de la couleur fournie au format RGB et son canal alpha alpha , dans l'image im .

Si la couleur n'existe pas dans la palette de l'image, imagecolorexactalpha retourne −1.

Voir aussi imagecolorclosestalpha .

Note
imagecolorexactalpha a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

6.41.21 ImageColorResolve() : Retourne l'index de la couleur donnée, ou la plus proche possible.

int **imagecolorresolve** (resource im , int red , int green , int blue)
[lien librairie]

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 .

6.41.22 ImageColorResolveAlpha() : Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha

int **imagecolorresolvealpha** (resource im , int red , int green , int blue , int alpha)
[lien librairie]

imagecolorresolvealpha retourne toujours un index de couleur, disponible dans la palette de l'image im : 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.

6.41.23 ImageColorSet() : Change la couleur dans une palette à l'index donné.

boolean **imagecolorset** (resource im , int index , int red , int green , int blue)
[lien librairie]

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 .

6.41.24 ImageColorsForIndex() : Retourne la couleur associée à un index.

`array` **imagecolorsforindex** (`resource im` , `int index`)
[lien librairie]

`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` .

6.41.25 ImageColorsTotal() : Calcule le nombre de couleurs d'une palette.

`int` **imagecolorstotal** (`resource im`)
[lien librairie]

`imagecolorstotal` retourne le nombre de couleurs de la palette.

Voir aussi `imagecolorat` et `imagecolorsforindex` .

6.41.26 ImageColorTransparent() : Définit la couleur transparente.

`int` **imagecolortransparent** (`resource im` , *`int col`*)
[lien librairie]

`imagecolortransparent` permet de choisir la couleur transparente d'une image, et de lui donner la valeur de col. `im` est un identifiant d'image, retourné par `imagecreate` et `col` est un identifiant de couleur retourné par `imagecolorallocate` .

L'identifiant de la nouvelle (ou courante) couleur transparente est retourné.

6.41.27 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`)
[lien librairie]

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` .

6.41.28 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`)
[lien librairie]

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.

6.41.29 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)
[lien librairie]

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.

6.41.30 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)
[lien librairie]

imagecopyresampled copie une zone rectangulaire de l'image src_im vers l'image src_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.

Voir aussi imagecopyresized .

Note
imagecopyresampled a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

6.41.31 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`)
[lien librairie]

`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.

Voir aussi `imagecopyresampled` .

6.41.32 ImageCreate() : Crée une nouvelle image à palette.

`resource` **imagecreate** (`int x_size , int y_size`)
[lien librairie]

`imagecreate` retourne un identifiant d'image représentant une image vide, de largeur `x_size` et longueur `y_size` .

6.41.33 imagecreatefromgif() : Crée une nouvelle image à partir d'un fichier ou d'une URL.

`resource` **imagecreatefromgif** (`string filename`)
[lien librairie]

`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 débuggage, utilisez le code suivant, qui retourne une erreur `GIF` :

Exemple de gestion des erreurs durant la création d'image (gracieusement offert par vic@zymsys.com)

```
<?php
function loadgif($imgname){
  $im = @imagecreatefromgif($imgname); /* Tentative d'ouverture */
  if ($im == "") { /* échec ? */
    $im = ImageCreate(150,30); /* Crée une image vide */
    $bgc = ImageColorAllocate($im,255,255,255);
    $tc  = ImageColorAllocate($im,0,0,0);
    imagefilledrectangle($im,0,0,150,30,$bgc);
    imagestring($im,1,5,5,"Erreur lors du chargement du fichier $imgname",$tc);
    /* Affiche un message d'erreur */
  }
  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.

6.41.34 ImageCreateFromJPEG() : Crée une nouvelle image JPEG à partir d'un fichier ou d'une URL

resource **imagecreatefromjpeg** (string filename)
[lien librairie]

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 débuggage, voici une erreur JPEG :

Exemple de gestion d'erreur lors de la création d'image (gracieusement offert par vic@zymsys.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;
}
?>
```

6.41.35 ImageCreateFromPNG() : Crée une nouvelle image PNG à partir d'un fichier ou d'une URL

resource **imagecreatefrompng** (string filename)
[lien librairie]

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 débuggage, voici une erreur PNG :

```
```

Exemple de gestion d'erreur lors de la création d'image (gracieusement offert par vic@zymsys.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);
        $tc  = imagecolorallocate($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;
}
?>
```

6.41.36 ImageCreateFromString() : Crée une image à partir d'une chaîne

`resource` **imagecreatefromstring** (`string string`)
[lien librairie]

`imagecreatefromstring` retourne un identifiant d'image représentant la chaîne `string` .

6.41.37 ImageCreateFromWBMP() : Crée une image depuis un fichier WBMP

`resource` **imagecreatefromwbmp** (`string filename`)
[lien librairie]

`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 débuggage, l'exemple suivant va produire une erreur `WBMP` :

Exemple de gestion des erreurs durant la création d'une image WBMP (gracieusement proposé par vic@zymsys.com)

```
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);
    $tc  = imagecolorallocate($im, 0, 0, 0);
    imagefilledrectangle($im, 0, 0, 10, 10, $bgc);
// Affiche le message d'erreur
    imagestring($im, 1, 5, 5, "Erreur de chargement de $imgname", $tc);
  }
  return $im;
}
```

Note

Le support WBMP n'est disponible qu'avec GD−1.8 ou plus récent.

6.41.38 ImageCreateFromXBM() : Crée une image à partir d'un fichier XBM

`int` **imagecreatefromxbm** (`string filename`)
[lien librairie]

`imagecreatefromxbm` retourne un identifiant d'image représentant l'image obtenue à partir du fichier `filename` .

6.41.39 ImageCreateFromXPM() : Crée une image à partir d'un fichier XPM

`int` **imagecreatefromxpm** (`string filename`)
[lien librairie]

`imagecreatefromxpm` retourne un identifiant d'image représentant l'image obtenue à partir du fichier `filename` .

6.41.40 ImageCreateTrueColor() : Crée une nouvelle image en vraies couleurs

`resource` **imagecreatetruecolor** (`int x_size` , `int y_size`)
[lien librairie]

`imagecreatetruecolor` retourne une ressource représentant une image noire de largeur `x_size` , et de hauteur `y_size` .

Note
`imagecreatetruecolor` a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

6.41.41 ImageDashedLine() : Dessine une ligne pointillée.

`int` **imagedashedline** (`resource im` , `int x1` , `int y1` , `int x2` , `int y2` , `int col`)
[lien librairie]

`imagedashedline` dessine une ligne pointillée entre les points (`x1` , `y1`) et (`x2` , `y2`) (le coin supérieur droit est l'origine (0,0)) dans l'image `im` , avec la couleur `col` .

Voir aussi `imageline` .

6.41.42 ImageDestroy() : détruit une image.

`int` **imagedestroy** (`resource im`)
[lien librairie]

imagedestroy libère toute la mémoire associée à l'image im . im est un identifiant d'image valide retourné par imagecreate .

6.41.43 ImageEllipse() : Dessine une ellipse

int **imageellipse** (resource im , int cx , int cy , int w , int h , int col)
[lien librairie]

imageellipse dessine une ellipse centrée sur le point (cx , cy). Le coin supérieur gauche est aux coordonnées (0,0). L'image de dessin est im . w et h spécifient respectivement la largeur et la hauteur de l'ellipse. La couleur de dessin de l'ellipse est color .

Note
imageellipse a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

6.41.44 ImageFill() : Remplit.

int **imagefill** (resource im , int x , int y , int col)
[lien librairie]

imagefill effectue un remplissage avec la couleur col , dans l'image im , à partir du point de coordonnées (x , y) (le coin supérieur gauche est l'origine (0,0)).

6.41.45 imagefilledarc() : Dessine une ellipse partielle et la remplit

int **imagefilledarc** (int im , int cx , int cy , int w , int h , int s , int e , int col , int style)
[lien librairie]

imagefilledarc dessine une ellipse partielle, centrée sur le point (cx , cy). Le coin supérieur gauche est (0, 0), dans l'image im . 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 mutellemement 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 camenbert.

Note
imagefilledarc a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

6.41.46 ImageFilledEllipse() : Dessine une ellipse pleine

`int` **imagefilledellipse** (`resource im` , `int cx` , `int cy` , `int w` , `int h` , `int col`)
[lien librairie]

`imagefilledellipse` dessine une ellipse centrée sur le point (`cx` , `cy`). Le coin supérieur gauche est aux coordonnées (0,0). L'image de dessin est `im` . `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.

6.41.47 ImageFilledPolygon() : Dessine un polygone rempli.

`int` **imagefilledpolygon** (`resource im` , `array points` , `int num_points` , `int col`)
[lien librairie]

`imagefilledpolygon` dessine un polygone rempli dans l'image `im` . `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.

6.41.48 ImageFilledRectangle() : Dessine un rectangle rempli.

`int` **imagefilledrectangle** (`resource im` , `int x1` , `int y1` , `int x2` , `int y2` , `int col`)
[lien librairie]

`imagefilledrectangle` dessine un rectangle de couleur `col` dans l'image `im` , en commençant par le sommet supérieur gauche (`x1` , `y1`) et finissant au sommet inférieur droit (`x2` , `y2`). Le coin supérieur gauche est l'origine (0, 0).

6.41.49 ImageFillToBorder() : Remplit avec une région avec une couleur spécifique.

`int` **imagefilltoborder** (`resource im` , `int x` , `int y` , `int border` , `int col`)
[lien librairie]

`imagefilltoborder` remplit avec la couleur `col` 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)).

6.41.50 ImageFontHeight() : Retourne la hauteur de la police.

`int` **imagefontheight** (`resource font`)
[lien librairie]

`imagefontheight` retourne la hauteur de la police `font` en pixels.

Voir aussi `imagefontwidth` et `imageloadfont` .

6.41.51 ImageFontWidth() : Retourne la largeur de la police.

`int` **imagefontwidth** (`resource font`)
[lien librairie]

`imagefontwidth` retourne la largeur de la police `font` en pixels.

Voir aussi `imagefontheight` et `imageloadfont` .

6.41.52 ImageGammaCorrect() : Applique une correction gamma à l'image

`int` **imagegammacorrect** (`resource im` , `float inputgamma` , `float outputgamma`)
[lien librairie]

`imagegammacorrect` applique une correction gamma à l'image GD `im` . Le facteur d'entrée est `inputgamma` , et le facteur de sortie est `outputgamma` .

6.41.53 ImageGif() : Envoie une image GIF vers un navigateur ou un fichier.

`int` **imagegif** (`resource im` , `string filename`)
[lien librairie]

`imagegif` crée un fichier image `GIF` avec le nom `filename` d'après l'image `im` . L'argument `im` 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 `Header("Content−type: image/gif"); ImageGif($im);` par un code plus souple :

```
<?php
  if (function_exists("imagegif")) {
    header("Content-type: image/gif");
    imagegif($im);
  }
  elseif (function_exists("imagejpeg")) {
    header("Content-type: image/jpeg");
    imagejpeg($im, "", 0.5);
  }
  elseif (function_exists("imagepng")) {
```

```
    header("Content-type: image/png");
    imagepng($im);
  } elseif (function_exists("imagewbmp")) {
    header("Content-type: image/vnd.wap.wbmp");
    imagewbmp($im);
  }  else {
    die("Pas de support graphique avec PHP sur ce serveur");
  }
?>
```

Note

En PHP 4, à partir de la version 4.0.2, vous pouvez utiliser la fonction `imagetypes` à la place de `function_exists` pour vérifier que certains formats d'images sont supportés :

```
<?php
  if (function_exists("imagegif")) {
    header("Content-type: image/gif");
    imagegif($im);
  }
  elseif (function_exists("imagejpeg")) {
    header("Content-type: image/jpeg");
    imagejpeg($im, "", 0.5);
  }
  elseif (function_exists("imagepng")) {
    header("Content-type: image/png");
    imagepng($im);
  } elseif (function_exists("imagewbmp")) {
    header("Content-type: image/vnd.wap.wbmp");
    imagewbmp($im);
  } else {
    die("Pas de support graphique avec PHP sur ce serveur");
  }
?>
```

Voir aussi `imagejpeg` , `imagewbmp` , `imagepng` , `imageinterlace` , `imagegif` et `imagetypes` .

6.41.54 ImageInterlace() : Active ou désactive l'entrelacement.

`int` **imageinterlace** (`resource im` , *`int interlace`*)
[lien librairie]

`imageinterlace` active ou désactive le bit d'entrelacement.

Si l'entrelacement est à 1, l'image `im` 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.

6.41.55 ImageJPEG() : Envoie une image JPEG vers un navigateur ou un fichier.

`int` **imagejpeg** (`resource im` , *`string filename`* , *`int quality`*)

[lien librairie]

imagejpeg envoie l'image GD (im) au format JPEG sur la sortie standard (typiquement, le navigateur web), ou si filename est fourni, l'envoie dans un fichier. im 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 .

6.41.56 ImageLine() : Dessine une ligne.

int **imageline** (resource im , int x1 , int y1 , int x2 , int y2 , int col)
[lien librairie]

imageline dessine une ligne depuis le point (x1 , y1) jusqu'au point (x2 , y2) (le coin supérieur gauche est l'origine (0,0)) dans l'image im et avec la couleur col .

Voir aussi imagecreate et imagecolorallocate .

6.41.57 ImageLoadFont() : Charge une nouvelle police.

resource **imageloadfont** (string file)
[lien librairie]

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` .

6.41.58 ImagePaletteCopy() : Copie la palette d'une image à l'autre

`int` **imagepalettecopy** (`resource destination` , `resource source`)
[lien librairie]

`imagepalettecopy` copie la palette de l'image `source` dans l'image `destination` .

6.41.59 ImagePNG() : Envoie une image PNG vers un navigateur ou un fichier.

`int` **imagepng** (`resource im` , *`string filename`*)
[lien librairie]

`imagepng` envoie l'image GD (`im`) au format PNG sur la sortie standard (typiquement, le navigateur web), ou si `filename` est fourni, l'envoie dans un fichier.

```
<?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 `imagejpeg` , `imagewbmp` , `imagegif` , `imageinterlace` , `imagegif` et `imagetypes` .

6.41.60 ImagePolygon() : Dessine un polygone.

`int` **imagepolygon** (`resource im` , `array points` , `int num_points` , `int col`)
[lien librairie]

`imagepolygon` dessine un polygone dans l'image `im` . `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` .

6.41.61 ImagePSBBox() : Retourne le rectangle entourant un texte et dessiné avec une police PostScript Type1.

`array` **imagepsbbox** (`string text` , `resource font` , `int size` , `int space` , `int width` , `float angle`)
[lien librairie]

`size` est exprimé en pixels.

`space` permet de changer la valeur par défaut du charactère 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
3	Ordonnée supérieure droite

Voir aussi `imagepstext` .

6.41.62 imagepscopyfont() : Fait une copie d'une police chargée pour modification ultérieure.

`int` **imagepscopyfont** (`int fontindex`)
[lien librairie]

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 `imageploadpsfont` .

6.41.63 ImagePSEncodeFont() : Change le codage vectoriel d'un caractère dans une police.

`int` **imagepsencodefont** (`resource font` , `string encodingfile`)
[lien librairie]

`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 set ps.default_encoding dans le fichier de configuration pour utiliser par défaut l'encodage correct.

6.41.64 ImagePsExtendFont() : Etend ou condense une police de caractères

`boolean` **imagepsextendfont** (`resource font` , `float extend`)
[lien librairie]

`imagepsextendfont` étend ou condense la police de caractères `font` . Si la valeur de `extend` est inférieure à 1, ce sera une condensation.

6.41.65 ImagePSFreeFont() : Libère la mémoire occupée par une police PostScript Type 1.

`void` **imagepsfreefont** (`resource font`)
[lien librairie]

Voir aussi `imagepsloadfont` .

6.41.66 ImagePSLoadFont() : Charge une police PostScript Type 1 depuis un fichier.

`resource` **imagepsloadfont** (`string filename`)
[lien librairie]

Au cas où tout a bien marché, un index de police va être retourné, et pourra être utilisé pour des opérations ultérieures. Sinon, la fonction retourne FALSE et affiche un message décrivant ce qui est erroné.

```
<?php
  header("Content-type: image/jpeg");
  $im =    imagecreate(350, 45);
  $noir =  imagecolorallocate($im, 0, 0, 0);
  $Blanc = imagecolorallocate($im, 255, 255, 255);
  $font =  imagepsloadfont("bchbi.pfb");
  imagepstext($im, "Test ... Ca marche!", $font, 32, $white, $black, 32, 32);
  imagepsfreefont($font);
  imagejpeg($im, "");
  imagedestroy ($im);
?<
```

Voir aussi `imagepsfreefont` .

6.41.67 ImagePsSlantFont() : Incline une police de caractères

`boolean` **imagepsslantfont** (`resource font` , `float slant`)
[lien librairie]

`imagepsslantfont` met en italique la police de caractères `font` avec le coefficient `slant` .

6.41.68 ImagePSText() : Dessine un texte sur une image avec une police PostScript Type1.

`array` **imagepstext** (`resource im` , `string text` , `resource font` , `int size` , `int foreground` , `int background` , `int x` , `int y` , *`int space`* , *`int tightness`* , *`float angle`* , *`int antialias_steps`*)
[lien librairie]

`size` est exprimé en pixels.

`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.

`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.

imagepstext 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 .

6.41.69 ImageRectangle() : Dessine un rectangle.

int **imagerectangle** (resource im , int x1 , int y1 , int x2 , int y2 , int col)
[lien librairie]

imagerectangle dessine un rectangle dans la couleur col , dans l'image im , et en commençant au point supérieur gauche (x1 , y1), et en finissant au point inférieur droit (x2 , y2). Le coin supérieur gauche est l'origine (0,0).

6.41.70 imagesetbrush() : Modifie la brosse pour le dessin des lignes

int **imagesetbrush** (resource im , resource brush)
[lien librairie]

imagesetbrush remplace la brosse courante pour le dessin des lignes par brush . Cette brosse sera alors utilisée avec les fonctions imageline et imagepolygon .

Note
Vous n'avez rien à faire lorsque vous en avez terminé avec une brosse, mais si vous détruisez l'image de brosse, vous ne DEVEZ plus utiliser les options IMG_COLOR_BRUSHED et IMG_COLOR_STYLEDBRUSHED des fonctions imageline et imagepolygon , avant d'avoir créé une nouvelle brosse.

Note
imagesetbrush a été ajoutée en PHP 4.0.6.

6.41.71 ImageSetPixel() : Dessine un pixel.

int **imagesetpixel** (resource im , int x , int y , int col)
[lien librairie]

imagesetpixel dessine un pixel au point (x , y) (le coin supérieur gauche est l'origine (0,0)) dans l'image im , et avec la couleur col .

Voir aussi imagecreate et imagecolorallocate .

6.41.72 ImageSetThickness() : Modifie l'épaisseur d'un trait

void **imagesetthickness** (resource im , int thickness)
[lien librairie]

imagesetthickness modifie l'épaisseur du trait des lignes de l'image im . 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.

6.41.73 ImageSetTile() : Modifie l'image utilisée pour le carrelage

int **imagesettile** (resource im , resource tile)
[lien librairie]

imagesettile remplace l'image de carrelage 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.

Note
imagesettile a été ajoutée en PHP 4.0.6.

6.41.74 ImageString() : Dessine une chaîne horizontale.

int **imagestring** (resource im , int font , int x , int y , string s , int col)
[lien librairie]

imagestring dessine une la chaîne sur une ligne horizontale, dans l'image im , aux coordonnées (x , y) (le coin supérieur gauche est l'origine (0,0)) dans la couleur col . Si l'argument de police vaut 1, 2, 3, 4 ou 5, une des polices par défaut sera utilisée).

Voir aussi imageloadfont .

6.41.75 ImageStringUp() : Dessine une chaîne verticale.

`int` **imagestringup** (`resource im` , `int font` , `int x` , `int y` , `string s` , `int col`)
[lien librairie]

`imagestringup` dessine une chaîne sur une ligne verticale dans l'image `im` aux coordonnées (`x` , `y`) (l'origine est le coin supérieur gauche (0,0)) dans la couleur `col` . Si la police utilisée est 1, 2, 3, 4 ou 5, une police par défaut sera utilisée.

Voir aussi `imageloadfont` .

6.41.76 ImageSX() : Retourne la largeur d'une image.

`int` **imagesx** (`resource im`)
[lien librairie]

`imagesx` retourne la largeur de l'image référencée par `im` .

Voir aussi `imagecreate` et `imagesy` .

6.41.77 ImageSY() : Retourne la hauteur de l'image.

`int` **imagesy** (`resource im`)
[lien librairie]

`imagesy` retourne la hauteur de l'image référencée par `im` .

Voir aussi `imagecreate` et `imagesx` .

6.41.78 ImageTrueColorToPalette() : Convertit une image en vraies couleurs en image à palette

`void` **imagetruecolortopalette** (`resource im` , `boolean dither` , `int ncolors`)
[lien librairie]

`imagetruecolortopalette` convertit l'image en vraies couleurs `im` 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
`imagetruecolortopalette` a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

6.41.79 ImageTTFBBox() : Retourne le rectangle entourant un texte et dessiné avec une police TrueType.

`array` **imagettfbbox** (`int size` , `int angle` , `string fontfile` , `string text`)
[lien librairie]

`imagettfbbox` 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.

`imagettfbbox` retourne une 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.

`imagettfbbox` requiert les bibliothèques GD et Freetype.

Voir aussi `imagettftext` .

6.41.80 ImageTTFText() : Dessine un texte avec une police TrueType.

`array` **imagettftext** (`resource im` , `int size` , `int angle` , `int x` , `int y` , `int col` , `string fontfile` , `string text`)
[lien librairie]

`imagettftext` dessine la chaîne `text` dans l'image `im` , en commençant aux coordonnées (`x` , `y`) (le coin supérieur gauche est l'origine (0,0)), avec un angle de `angle` , et dans la couleur `col` , en utilisant la police TrueType identifiée par `fontfile` .

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.

`col` 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 `GIF` 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/gif");
  $im = imagecreate(400,30);
  $black = imagecolorallocate($im, 0,0,0);
  $white = imagecolorallocate($im, 255,255,255);
  imagettftext($im, 20, 0, 10, 20, $white, "/path/arial.ttf",
  "Test en cours... Oméga: &#937;");
  imagegif($im);
  imagedestroy($im);
?>
```

`imagettftext` requiert les bibliothèques GD ainsi que FreeType .

Voir aussi `imagettfbbox` .

6.41.81 ImageTypes() : Retourne les types d'images supportés par la version courante de PHP

`int` **imagetypes** (`void`)
[lien librairie]

`imagetypes` retourne un champs 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";
  }
?>
```

6.41.82 ImageWBMP() : Affiche une image WBMP

`int` **imagewbmp** (`resource im` , *`string filename`* , *`int foreground`*)
[lien librairie]

`imagewbmp` crée l'image `WBMP` dans le fichier `filename` , à partir de l'image `im` . Le paramètre `im` 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 champs "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` .

6.41.83 iptcparse() : Analyse un bloc binaire IPTC http://www.iptc.org/ et recherche les balises simples.

`array` **iptcparse** (`string iptcblock`)
[lien librairie]

`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.

6.41.84 JPEG2WBMP() : Convertit une image JPEG en image WBMP

`int` **jpeg2wbmp** (`string jpegname` , `string wbmpname` , `int d_height` , `int d_width`)
[lien librairie]

`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` .

6.41.85 PNG2WBMP() : Convertit une image PNG en image WBMP

`int` **png2wbmp** (`string pngname` , `string wbmpname` , `int d_height` , `int d_width`)
[lien librairie]

`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` .

6.41.86 read_exif_data() : Lit les en−têtes EXIF d'une image JPEG

`array` **read_exif_data** (`string filename`)
[lien librairie]

`read_exif_data` lit les en−têtes EXIF de l'image `JPEG` nommée `filename` . Elle retourne un tableau associatif où les index sont les noms d'en−têtes EXIF, et les valeurs sont leur valeur associée. Les en−têtes EXIF sont souvent disponibles dans les images générées par les appareils photos numériques, mais chaque constructeur marque ses images d'une manière qui lui est propre : il est impossible de savoir quelles en−têtes seront présents.

Exemple avec `read_exif_data`

```
<?php
  $exif = read_exif_data('p0001807.jpg');
  while(list($k,$v)=each($exif)) {
    echo "$k: $v<br>\n";
  }
?>
```

Note
`read_exif_data` n'est disponible que sous PHP 4 , compilé avec −−enable−exif . `read_exif_data` ne requiert pas la librairie GD.

6.42 IMAP

6.42.1 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.

6.42.2 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`
- `imap4r1.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.

6.42.3 Installation

Pour avoir accès à ces fonctions, vous devez compiler PHP avec l'option `−−with−imap` .

Note

Suivant le la librairie C cliente qui a été configurée, vous pouvez aussi avoir besoion d'ajouter `--with-imap-ssl=/path/to/openssl/` et/ou `--with-kerberos` dans la ligne de configuration de PHP.

6.42.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.42.5 Types de ressources

6.42.6 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` (nombre ˆ virgule flottante)
`OP_DEBUG` (nombre ˆ virgule flottante)
`OP_READONLY` (nombre ˆ virgule flottante)
`OP_ANONYMOUS` (nombre ˆ virgule flottante)
`OP_SHORTCACHE` (nombre ˆ virgule flottante)
`OP_SILENT` (nombre ˆ virgule flottante)
`OP_PROTOTYPE` (nombre ˆ virgule flottante)
`OP_HALFOPEN` (nombre ˆ virgule flottante)
`OP_EXPUNGE` (nombre ˆ virgule flottante)
`OP_SECURE` (nombre ˆ virgule flottante)
`CL_EXPUNGE` (nombre ˆ virgule flottante)
`FT_UID` (nombre ˆ virgule flottante)
`FT_PEEK` (nombre ˆ virgule flottante)
`FT_NOT` (nombre ˆ virgule flottante)
`FT_INTERNAL` (nombre ˆ virgule flottante)
`FT_PREFETCHTEXT` (nombre ˆ virgule flottante)
`ST_UID` (nombre ˆ virgule flottante)
`ST_SILENT` (nombre ˆ virgule flottante)
`ST_SET` (nombre ˆ virgule flottante)
`CP_UID` (nombre ˆ virgule flottante)
`CP_MOVE` (nombre ˆ virgule flottante)
`SE_UID` (nombre ˆ virgule flottante)
`SE_FREE` (nombre ˆ virgule flottante)
`SE_NOPREFETCH` (nombre ˆ virgule flottante)
`SO_FREE` (nombre ˆ virgule flottante)
`SO_NOSERVER` (nombre ˆ virgule flottante)
`SA_MESSAGES` (nombre ˆ virgule flottante)
`SA_RECENT` (nombre ˆ virgule flottante)
`SA_UNSEEN` (nombre ˆ virgule flottante)
`SA_UIDNEXT` (nombre ˆ virgule flottante)
`SA_UIDVALIDITY` (nombre ˆ virgule flottante)
`SA_ALL` (nombre ˆ virgule flottante)
`LATT_NOINFERIORS` (nombre ˆ virgule flottante)
`LATT_NOSELECT` (nombre ˆ virgule flottante)
`LATT_MARKED` (nombre ˆ virgule flottante)

LATT_UNMARKED (nombre ˆ virgule flottante)
SORTDATE (nombre ˆ virgule flottante)
SORTARRIVAL (nombre ˆ virgule flottante)
SORTFROM (nombre ˆ virgule flottante)
SORTSUBJECT (nombre ˆ virgule flottante)
SORTTO (nombre ˆ virgule flottante)
SORTCC (nombre ˆ virgule flottante)
SORTSIZE (nombre ˆ virgule flottante)
TYPETEXT (nombre ˆ virgule flottante)
TYPEMULTIPART (nombre ˆ virgule flottante)
TYPEMESSAGE (nombre ˆ virgule flottante)
TYPEAPPLICATION (nombre ˆ virgule flottante)
TYPEAUDIO (nombre ˆ virgule flottante)
TYPEIMAGE (nombre ˆ virgule flottante)
TYPEVIDEO (nombre ˆ virgule flottante)
TYPEOTHER (nombre ˆ virgule flottante)
ENC7BIT (nombre ˆ virgule flottante)
ENC8BIT (nombre ˆ virgule flottante)
ENCBINARY (nombre ˆ virgule flottante)
ENCBASE64 (nombre ˆ virgule flottante)
ENCQUOTEDPRINTABLE (nombre ˆ virgule flottante)
ENCOTHER (nombre ˆ virgule flottante)

6.42.7 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 disponibles dans les livres suivants (en anglais): Programming Internet Email par David Wood et Managing IMAP par Dianna Mullet & Kevin Mullet.

Attention
Des crashs 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_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_getmailboxes
- imap_getsubscribed
- imap_header
- imap_headerinfo
- imap_headers
- imap_last_error
- imap_listmailbox
- imap_listsubscribed
- 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_setflag_full
- imap_sort
- imap_status
- imap_subscribe
- imap_uid
- imap_undelete
- imap_unsubscribe

- imap_utf7_decode
- imap_utf7_encode
- imap_utf8

6.42.9 imap_alerts() : Retourne toutes les alertes

`array` **imap_alerts** (`void`)
[lien librairie]

`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.

6.42.10 imap_append() : Ajoute une chaîne dans une boîte aux lettres.

`int` **imap_append** (`resource imap_stream` , `string mbox` , `string message` , `string flags`)
[lien librairie]

`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.

`imap_append` retourne `TRUE` en cas de succès, et `FALSE` en cas d'erreur.

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("{your.imap.host}INBOX.Drafts","username", "password");
$check = imap_check($stream);
print "Nombre de message avant ajout : ". $check->Nmsgs."\n";
imap_append($stream,"{your.imap.host}INBOX.Drafts"
                  ,"From: me@my.host\r\n"
                  ."To: you@your.host\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 message après ajout : ". $check->Nmsgs."\n";
imap_close($stream);
?>
```

6.42.11 imap_base64() : Décode un texte encodé en BASE64.

`string` **imap_base64** (`string text`)
[lien librairie]

`imap_base64` décode un texte encodé en BASE64. Le texte décodé est retourné sous la forme d'une chaîne.

6.42.12 imap_binary() : Convertit une chaîne à 8 bits en une chaîne à base64.

`string` **imap_binary** (`string string`)
[lien librairie]

`imap_binary` convertit la chaîne à 8 bits `string` en une chaîne à base64.

`imap_binary` retourne la chaîne codée.

Voir aussi `imap_base64` .

6.42.13 imap_body() : Lit le corps d'un message.

`string` **imap_body** (`resource imap_stream` , `int msg_number` , `int flags`)
[lien librairie]

`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.

6.42.14 imap_check() : Vérifie le courrier de la boîte aux lettres courante.

`object` **imap_check** (`resource imap_stream`)
[lien librairie]

`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

6.42.15 imap_clearflag_full() : Supprime un flag sur un message.

`string` **imap_clearflag_full** (`resource stream` , `string sequence` , `string flag` , `string options`)
[lien librairie]

`imap_clearflag_full` efface le flag `flag` dans les messages de la séquence `sequence` , du flot imap `stream` .

Les options sont un masque de bit, qui accepte les valeurs suivantes :

```
      ST_UID  : la séquence contient des UIDs au lieu de numéro
      de séquence
```

6.42.16 imap_close() : Termine un flot IMAP.

`int` **imap_close** (`resource imap_stream` , `int flags`)
[lien librairie]

`imap_close` termine un flot `IMAP` . `imap_close` prend un argument optionnel `flag` , CL_EXPUNGE, qui va retirer automatiquement de la liste la boîte aux lettres.

6.42.17 imap_createmailbox() : Crée une nouvelle boîte aux lettres.

`int` **imap_createmailbox** (`resource imap_stream` , `string mbox`)
[lien librairie]

`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 = "nouvellephpbox";
$name2 = imap_utf7_encode("nouvellephpboxéx");
$newname = $name1;
echo "Le nouveau nom sera '$name1'<br>\n";
# Nous allons créer maintenant une nouvelle boîte aux lettres "phptestbox"
# dans votre dossier inbox, vérifier son état et finalement, la supprimer
# pour remettre votre inbox dans son état initial.
if(@imap_createmailbox($mbox,imap_utf7_encode("{your.imap.host}INBOX.$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 "renommage 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
  }
} 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_er
}
} else {
 print  "Impossible de créer une nouvelle boîte aux lettres : ".implode("<br>\n",imap_errors())."<br
}
imap_close($mbox);
?>
```

Voir aussi `imap_renamemailbox` , `imap_deletemailbox` et `imap_open` pour connaître le format des noms de `mbox` .

6.42.18 imap_delete() : Marque le fichier pour l'effacement, dans la boîte aux lettres courante.

`int` **imap_delete** (`resource imap_stream` , `int msg_number` , *`int flags`*)
[lien librairie]

`imap_delete` retourne `TRUE` .

`imap_delete` marque le fichier `msg_number` pour l'effacement, dans la boîte aux lettres courante. Le paramètre optionnel `flags` ne prend qu'une seule valeur, `FT_UID` , qui indique à PHP qu'il faut traiter `msg_number` comme un `UID` . L'effacement réel n'interviendra que lors de l'appel de la fonction `imap_expunge` .

Exemple `imap_delete`

```
<?php
$mbox = imap_open ("{your.imap.host}INBOX", "utilisateur", "mot_de_passe")
   or die ("connexion impossible: " . imap_last_error());
$check = imap_mailboxmsginfo ($mbox);
print "Nombre de messages avant effacement  : " . $check->Nmsgs . "<br>\n" ;
imap_delete ($mbox, 1);
$check = imap_mailboxmsginfo ($mbox);
print "Nombre de messages après effacement: " . $check->Nmsgs . "<br>\n" ;
imap_expunge ($mbox);
$check = imap_mailboxmsginfo ($mbox);
print "Nombre de messages après imap_expunge: " . $check->Nmsgs . "<br>\n" ;
imap_close ($mbox);
?>
```

6.42.19 imap_deletemailbox() : Efface une boîte aux lettres.

`int` **imap_deletemailbox** (`resource imap_stream` , `string mbox`)
[lien librairie]

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 .

6.42.20 imap_errors() : Retourne toutes les erreurs

array **imap_errors** (void)
[lien librairie]

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.

6.42.21 imap_expunge() : Efface tous les messages marqués pour l'effacement.

int **imap_expunge** (resource imap_stream)
[lien librairie]

imap_expunge efface tous les messages marqués pour l'effacement par imap_delete .

imap_expunge retourne TRUE .

6.42.22 imap_fetch_overview() : Lit un sommaire des en−têtes de messages

array **imap_fetch_overview** (resource imap_stream , string sequence , *int flags*)
[lien librairie]

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 flags 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("{votre.hote.imap}","utilisateur","mot_de_passe")
    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);
?>
```

Voir aussi imap_fetchstructure .

6.42.23 imap_fetchbody() : Retourne une section extraite du corps d'un message.

string **imap_fetchbody** (resource imap_stream , int msg_number , string part_number , *flags flags*)
[lien librairie]

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 − msgono 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.

6.42.24 imap_fetchheader() : Retourne l'en−tête d'un message.

string **imap_fetchheader** (resource imap_stream , int msgno , int flags)
[lien librairie]

imap_fetchheader retourne l'en−tête brute et complète RFC 822 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).
```

6.42.25 imap_fetchstructure() : Lit la structure d'un message.

`object` **imap_fetchstructure** (`resource imap_stream` , `int msg_number` , *`int flags`*)
[lien librairie]

`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 au ne 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_fetchstructure` .

6.42.26 imap_get_quota() : Lit les quotas des boîtes aux lettres

`array` **imap_get_quota** (`resource imap_stream` , `string quota_root`)
[lien librairie]

`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 flot 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
$mbox = imap_open("{votre.hote.imap}","mailadmin","mot de passe",OP_HALFOPEN)
     or die("Connexion impossible : ".imap_last_error());
$quota_value = imap_get_quota($mbox, "user.toto");
if(is_array($quota_value)) {
   print "Utilisation actuelle : " . $quota_value['usage'];
   print "Quota : " . $quota_value['limit'];
}
imap_close($mbox);
?>
```

Voir aussi `imap_open` et `imap_set_quota` .

6.42.27 imap_getmailboxes() : Liste les boîtes aux lettres, et retourne le détail pour chacune.

`array` **imap_getmailboxes** (`resource imap_stream` , `string ref` , `string pat`)
[lien librairie]

`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 charcher. Si vous voulez passer en revue toute la hiérarchier, 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'interesse 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
$mbox = imap_open("{your.imap.host}","utilisateur","mot_de_passe",OP_HALFOPEN)
     or die("connexion impossible: ".imap_last_error());
$list = imap_getmailboxes($mbox,"{your.imap.host}","*");
if(is_array($list)) {
 reset($list);
 while (list($key, $val) = each($list))
 {
   print "($key) ";
   print imap_utf7_decode($val->name).",";
   print "'".$val->delimiter."',";
   print $val->attributes."<br>\n";
 }
} else
 print "imap_getmailboxes a échoué : ".imap_last_error()."\n";
imap_close($mbox);
?>
```

Voir aussi `imap_getsubscribed` .

6.42.28 imap_getsubscribed() : Liste toutes les boîtes aux lettres souscrites.

`array` **imap_getsubscribed** (`resource imap_stream` , `string ref` , `string pattern`)
[lien librairie]

`imap_getsubscribed` est identique à `imap_getmailboxes` , mais ne retourne que les boîtes aux lettres auxquelles l'utilisateur est inscrit.

6.42.29 imap_header() : Lit l'en−tête d'un message.

`object` **imap_header** (`resource imap_stream` , `int msg_number` , *`int fromlength`* , *`int subjectlength`* , *`string defaulthost`*)
[lien librairie]

`imap_header` est un alias de `imap_headerinfo` et lui est identique en tous points.

6.42.30 imap_headerinfo() : Lit l'en−tête du message

`object` **imap_headerinfo** (`resource imap_stream` , `int msg_number` , *`int fromlength`* , *`int subjectlength`* , *`string defaulthost`*)
[lien librairie]

`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 savor 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
udate (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)
```

6.42.31 imap_headers() : Retourne les en−têtes de tous les messages d'une boîte aux lettres.

`array` **imap_headers** (`resource imap_stream`)
[lien librairie]

`imap_headers` retourne un tableau de chaîne contenant les en−tête des messages. Une chaîne par message.

6.42.32 imap_last_error() : Retourne la dernière erreur (si elle existe) qui est survenu lors de la dernière requête.

`string` **imap_last_error** (`void`)
[lien librairie]

`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.

6.42.33 imap_listmailbox() : Liste les boîtes aux lettres.

`array` **imap_listmailbox** (`resource imap_stream` , `string ref` , `string pat`)
[lien librairie]

`imap_listmailbox` retourne un tableau contenant les noms des boîtes aux lettres.

Exemple avec `imap_listmailbox`

```
<?php
$mbox = imap_open("{your.imap.host}","utilisateur","mot_de_passe",OP_HALFOPEN)
     or die("connexion impossible: ".imap_last_error());
$list = imap_listmailbox($mbox,"{your.imap.host}","*");
if(is_array($list)) {
 reset($list);
 while (list($key, $val) = each($list))
   print imap_utf7_decode($val)."<br>\n";
} else
 print "imap_listmailbox a échoué: ".imap_last_error()."\n";
imap_close($mbox);
?>
```

6.42.34 imap_listsubscribed() : Liste les boîtes aux lettres souscrites.

`array` **imap_listsubscribed** (`resource imap_stream` , `string ref` , `string pattern`)
[lien librairie]

`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.

6.42.35 imap_mail_compose() : Crée un message MIME

`string` **imap_mail_compose** (`array envelope` , `array body`)
[lien librairie]

Exemple `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";
$fp=fopen($filename,"r");
$contents=fread($fp,filesize($filename));
fclose($fp);
$part2["type"]=TYPEAPPLICATION;
$part2["encoding"]=ENCBINARY;
$part2["subtype"]="octet-stream";
$part2["description"]=basename($filename);
$part2["contents.data"]=$contents;
$part3["type"]=TYPETEXT;
$part3["subtype"]="plain";
$part3["description"]="description3";
```

```
$part3["contents.data"]="contents.data3\n\n\n\t";
$body[1]=$part1;
$body[2]=$part2;
$body[3]=$part3;
echo nl2br(imap_mail_compose($envelope,$body));
?>
```

6.42.36 imap_mail_copy() : Copie les messages spécifiés dans une boîte aux lettres.

`int` **imap_mail_copy** (`resource imap_stream` , `string msglist` , `string mbox` , `int flags`)
[lien librairie]

`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.

`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.

6.42.37 imap_mail_move() : Déplace les messages spécifiés dans une boîte aux lettres.

`int` **imap_mail_move** (`resource imap_stream` , `string msglist` , `string mbox` , *`int flags`*)
[lien librairie]

`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.

`flags` est un champs de bit et peut contenir une seule valeur :

- CP_UID − La séquence de nombrs contient UIDS

`imap_mail_move` retourne TRUE en cas de succès, et FALSE en cas d'erreur.

6.42.38 imap_mail() : Envoie un message mail

`string` **imap_mail** (`string to` , `string subject` , `string message` , *`string additional_headers`* , *`string cc`* , *`string bcc`* , *`string rpath`*)
[lien librairie]

`imap_mail` est uniquement disponible sous PHP 3.

6.42.39 imap_mailboxmsginfo() : Lit les informations à propos de la boîte aux lettres courante.

`object` **imap_mailboxmsginfo** (`resource imap_stream`)
[lien librairie]

`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("conexion 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);
?>
```

6.42.40 imap_mime_header_decode() : Décode les éléments MIME d'une en−tête

`array` **imap_mime_header_decode** (`string text`)
[lien librairie]

`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, sil 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.

6.42.41 imap_msgno() : Retourne le numéro de séquence de message pour un UID donné.

`int` **imap_msgno** (`resource imap_stream` , `int uid`)
[lien librairie]

`imap_msgno` retourne le numéro de séquence de message pour l'UID `uid` . C'est la fonction contraire de `imap_uid` .

6.42.42 imap_num_msg() : Retourne le nombre de message dans la boîte aux lettres courante.

`int` **imap_num_msg** (`resource imap_stream`)
[lien librairie]

`imap_num_msg` retourne le nombre de message dans la boîte aux lettres courante.

Voir aussi `imap_num_recent` et `imap_status` .

6.42.43 imap_num_recent() : Retourne le nombre de messages récents dans la boîte aux lettres courante.

`int` **imap_num_recent** (`resource imap_stream`)
[lien librairie]

`imap_num_recent` retourne le nombre de message récents dans la boîte aux lettres courante.

Voir aussi `imap_num_msg` et `imap_status` .

6.42.44 imap_open() : Ouvre un flot IMAP vers une boîte aux lettres.

`int` **imap_open** (`string mailbox` , `string username` , `string password` , *`int flags`*
)
[lien librairie]

imap_open retourne un flot 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 .newsrc 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 flot.

Pour se connecter à un serveur IMAP , on peut utiliser la commande suivante :

```
<?php
$mbox  = 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 :

```
<?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 :

```
<?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 :

```
<?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:

```
<?php
$nntp = imap_open("{localhost:119/nntp}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);
?>
```

6.42.45 imap_ping() : Vérifie que le flot IMAP est toujours actif.

`int` **imap_ping** (`resource imap_stream`)
[lien librairie]

`imap_ping` retourne TRUE si le flot `imap_stream` existe toujours, et FALSE sinon.

`imap_ping` vérifie que le flot 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.

6.42.46 imap_qprint() : Convertit une chaîne à guillemets en une chaîne à 8 bits.

`string` **imap_qprint** (`string string`)
[lien librairie]

`imap_qprint` convertit la chaîne à guillemets `string` en une chaîne à 8 bits.

`imap_qprint` retourne une chaîne 8 bits (binaire).

Voir aussi `imap_8bit` .

6.42.47 imap_renamemailbox() : Renomme une boîte aux lettres.

`int` **imap_renamemailbox** (`resource imap_stream` , `string old_mbox` , `string new_mbox`)
[lien librairie]

`imap_renamemailbox` renomme la boîte aux lettres `old_mbox` en `new_mbox` .

`imap_renamemailbox` retourne `TRUE` en cas de succès, et `FALSE` en cas d'erreur.

Voir aussi `imap_createmailbox` , `imap_deletemailbox` et `imap_open` pour le format de `mbox` .

6.42.48 imap_reopen() : Ouvre un flot IMAP vers une nouvelle boîte aux lettres.

`int` **imap_reopen** (`resource imap_stream` , `string mailbox` , *`string flags`*)
[lien librairie]

`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 .newsrc 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 flot. (voir `imap_delete` et `imap_expunge`).

6.42.49 imap_rfc822_parse_adrlist() : Analyse une chaîne d'adresse.

`string` **imap_rfc822_parse_adrlist** (`string address` , `string default_host`)
[lien librairie]

`address` analyse la chaîne `address` et essaie, pour chaque adresse, de retourner un tableau d'objets. Les 4 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($address_array)) die("une erreur...\n");
reset($address_array);
while(list($key,$val)=each($address_array)){
 print "boîte   : ".$val->mailbox."<br>\n";
```

```
 print "hôte    : ".$val->host."<br>\n";
 print "personnel: ".$val->personal."<br>\n";
 print "adl     : ".$val->adl."<p>\n";
}
?>
```

6.42.50 imap_rfc822_parse_headers() : Analyse une en−tête mail

`object` **imap_rfc822_parse_headers** (`string headers` , *`string defaulthost`*)
[lien librairie]

`imap_rfc822_parse_headers` analyse la chaîne `headers` , et retourne un objet contenant differents éléments, similaires à la fonction `imap_header` , hormis les flags, et autres éléments liés au serveur `IMAP` .

6.42.51 imap_rfc822_write_address() : Retourne une adresse email proprement formatée

`string` **imap_rfc822_write_address** (`string mailbox` , `string host` , `string personal`)
[lien librairie]

`mailbox` retourne une adresse email proprement formatée, 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";
?>
```

6.42.52 imap_scanmailbox() : Lit la liste des boîtes aux lettres, et y recherche une chaîne.

`array` **imap_scanmailbox** (`resource imap_stream` , `string string`)
[lien librairie]

`imap_scanmailbox` retourne un tableau contenant les noms des boîtes aux lettres qui contiennent la chaîne `string` . `imap_scanmailbox` est simialaire à `imap_listmailbox` , mais va aussi rechercher la chaîne `string` dans dans les données de la boîte aux lettres. Reportez−vous à `imap_getmailboxes` pour une description des paramètres `ref` et `pattern` .

6.42.53 imap_search() : Retourne un tableau de message après recherche.

`array` **imap_search** (`resource imap_stream` , `string criteria` , `int flags`)
[lien librairie]

imap_search effectue une recherche dans la boîte aux lettres courante, sur le flot 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 champs 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 champs 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" dan le champs From:
- KEYWORD "string" − tous les messags 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 champs 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 champs Subject:
- TEXT "string" − tous les messages avec le texte "string"
- TO "string" − tous les messages avec la chaîne "string" dans le champs 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 inprécise. (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.

6.42.54 imap_set_quota() : Modifie le quota d'une boîte aux lettres

`int` **imap_set_quota** (`resource imap_stream` , `string quota_root` , `int quota_limit`)
[lien librairie]

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 librairies c−client2000.

imap_stream doit avoir été créé avec la fonction imap_open . Ce flot 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
$mbox = imap_open ("{votre.hote.imap:143}", "mailadmin", "mot de passe");
if(!imap_set_quota($mbox, "user.toto", 3000)) {
   print "Erreur lors de la modification des quotas\n";
   return;
}
imap_close($mbox);
?>
```

Voir aussi imap_open et imap_set_quota .

6.42.55 imap_setflag_full() : Positionne un flag sur un message.

string **imap_setflag_full** (int stream , string sequence , string flag , string options)
[lien librairie]

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 :

```
      ST_UID  la séquence contient des UIDs au lieu de numéro
      de séquence.
```

Exemple avec imap_setflag_full

```
<?php
$mbox = imap_open("{votre.hote.imap:143}","utilisateur","mot_de_passe")
    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);
?>
```

6.42.56 imap_sort() : Trie des messages.

string **imap_sort** (int stream , int criteria , int reverse , int options)
[lien librairie]

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 :

```
      SORTDATE        Date du message
      SORTARRIVAL     Date d'arrivée
      SORTFROM        Nom de la première boîte aux lettres de l'adresse d'origine (From address)
      SORTSUBJECT     Sujet du message
      SORTTO          Nom de la première boîte aux lettres de destination (To address)
      SORTCC          Nom de la boîte aux lettres de copie cachée (cc address)
      SORTSIZE        Taille du message en octets
```

Les flags dont des masques de bits, d'un ou plusieurs des éléments suivants :

```
      SE_UID          Retourne l'UIDs à la place d'une séquence
      de nombres.
      SE_NOPREFETCH   Ne pas pré-télécharger les messages
      trouvés.
```

6.42.57 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`)
[lien librairie]

`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 destatus−>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
  $mbox = imap_open("{your.imap.host}","utilisateur","mot_de_passe",OP_HALFOPEN)
     or die("can't connect: ".imap_last_error());
  $status = imap_status($mbox,"{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);
?>
```

6.42.58 imap_subscribe() : Souscrit à une boîte aux lettres.

`int` **imap_subscribe** (`resource imap_stream` , `string mbox`)
[lien librairie]

`imap_subscribe` souscrit à la boîte aux lettres `mbox` .

`imap_subscribe` retourne TRUE en cas de succès, et FALSE en cas d'échec.

6.42.59 imap_uid() : Retourne l'UID d'un message.

`int` **imap_uid** (`resource imap_stream` , `int msgno`)
[lien librairie]

`imap_uid` retourne l'UID pour le message `msgno` . Un UID est un identifant unique que ne change jamais, alors que le numéro du message dans la lsite 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.

6.42.60 imap_undelete() : Enlève la marque d'effacement d'un message.

`int` **imap_undelete** (`resource imap_stream` , `int msg_number`)
[lien librairie]

`imap_undelete` enlève la marque d'effacement du message `msg_number` , placée avec `imap_delete` .

`imap_subscribe` retourne TRUE en cas de succès, et FALSE en cas d'erreur.

6.42.61 imap_unsubscribe() : Termine la souscription à une boîte aux lettres.

`int` **imap_unsubscribe** (`resource imap_stream` , `string mbox`)
[lien librairie]

`imap_unsubscribe` termine la souscription à la boîte aux lettres `mbox` .

`imap_unsubscribe` retourne TRUE en cas de succès, et FALSE en cas d'erreur.

6.42.62 imap_utf7_decode() : Décode une chaîne modifiée UTF−7.

`string` **imap_utf7_decode** (`string text`)
[lien librairie]

`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).

6.42.63 imap_utf7_encode() : Convertit des données 8bit en texte UTF−7.

`string` **imap_utf7_encode** (`string data`)
[lien librairie]

`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.

6.42.64 imap_utf8() : Convertit du texte en UTF8

`string` **imap_utf8** (`string text`)
[lien librairie]

`imap_utf8` convertit le texte `text` en UTF8 (comme défini dans RFC2044).

6.43 Informix

6.43.1 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.

6.43.2 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.

6.43.3 Installation

Note
Le script de configuration va détecter automatiquement les librairies disponibles, et inclure les dossiers si vous lancer le script avec l'option `−−with−informix=yes` . Vous pouvez ignorer cette détection en spécifiant "IFX_LIBDIR", "IFX_LIBS" et "IFX_INCDIR" dans votre environnement. Le script de configuration va aussi essayer de détecter la version de votre serveur Informix. Il modifiera alors la condition de compilation "HAVE_IFX_IUS" si votre serveur Informix est d'une version plus récente que 9.00.

6.43.4 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 (vérifiéez 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 de détails sur les directives de configuration.

`ifx.allow_persistent` *`boolean`*
: 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 boolean`
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 boolean`
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 boolean`
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 boolean`
Lorsque cette option est activée, les espaces en fin de chaîne de caractères seront conservés lors d'une commande FETCH.

`ifx.nullformat boolean`
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` .

6.43.5 Types de ressources

6.43.6 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
- ifx_free_slob
- ifxus_open_slob
- ifxus_read_slob
- ifxus_seek_slob
- ifxus_tell_slob
- ifxus_write_slob

6.43.8 ifx_blobinfile_mode() : Choisit le mode par défaut des objets BLOB pour toutes les requêtes SELECT.

`void` **ifx_blobinfile_mode** (`int mode`)
[lien librairie]

`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.

6.43.9 ifx_byteasvarchar() : Choisit le mode par défaut des objets BYTE.

`void` **ifx_byteasvarchar** (`int mode`)
[lien librairie]

`ifx_byteasvarchar` modifie le mode par défaut des objets BYTE. Le mode "0" retournera l'identifiant de BLOB, et le mode "1" retourenra le contenu du TEXT sous la forme d'un VARCHAR.

6.43.10 ifx_close() : Ferme une connexion à un serveur Informix.

`int` **ifx_close** (*`int link_identifier`*)
[lien librairie]

`ifx_close` retourne toujours TRUE .

`ifx_close` ferme le lien au serveur de données Informix associé à l'identifant de connexion `link_identifier` . 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
$conn_id = ifx_connect ("mydb&#64;ol_srv", "coucou", "cestmoi");
//... quelques requêtes diverses et variées ...
ifx_close($conn_id);
?>
```

6.43.11 ifx_connect() : Ouvre une connexion à un serveur Informix.

`int` **ifx_connect** (*`string database`* , *`string userid`* , *`string password`*)
[lien librairie]

`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 librairies 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&#64;ol_srv1", "imyself", "mypassword");
?>
```

6.43.12 ifx_copy_blob() : Duplique un objet BLOB.

`int` **ifx_copy_blob** (`int bid`)
[lien librairie]

`ifx_copy_blob` retourne FALSE en cas d'erreur, et sinon, l'identifiant du nouvel objet.

6.43.13 ifx_create_blob() : Crée un objet BLOB.

`int` **ifx_create_blob** (`int type` , `int mode` , `string param`)
[lien librairie]

`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.

6.43.14 ifx_create_char() : Crée un objet char.

`int` **ifx_create_char** (`string param`)
[lien librairie]

`ifx_create_char` crée un objet char. `param` sera le contenu de l'objet.

6.43.15 ifx_do() : Exécute une requête SQL déjà préparée.

`int` **ifx_do** (`int result_id`)
[lien librairie]

`ifx_do` retourne TRUE en cas de succès, FALSE en cas d'erreur.

Exécute une requête qui a déjà été préparée, ou crée un pointeur pour cela.

Ne libère pas `result_id` en cas d'erreur.

De plus, elle fixe la valeur de `result_id` pour accès ultérieur par `ifx_affected_rows` .

Voir aussi `ifx_prepare` (Pour un exemple).

6.43.16 ifx_error() : Retourne le code d'erreur de la dernière requête Informix.

`string` **ifx_error** (`void`)
[lien librairie]

`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` .

6.43.17 ifx_errormsg() : Retourne le message d'erreur de la dernière requête Informix.

`string` **ifx_errormsg** (*`int errorcode`*)
[lien librairie]

`ifx_errormsg` retourne le plus récent message d'erreur ou, lorsque l'option `errorcode` est présent, le message d'erreur associé à `errorcode` .

Voir aussi `ifx_error` .

```
printf("%s\n<br>", ifx_errormsg(−201));
```

6.43.18 ifx_fetch_row() : Retourne une ligne sous la forme d'un tableau énuméré.

`array` **ifx_fetch_row** (`int result_id` , *`mixed position`*)
[lien librairie]

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);
?>
```

6.43.19 ifx_fieldproperties() : Liste les propriétés des champs SQL.

array **ifx_fieldproperties** (int result_id)
[lien librairie]

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 champs 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);
}
?>
```

6.43.20 ifx_fieldtypes() : Liste les champs Informix SQL.

`array` **ifx_fieldtypes** (`int result_id`)
[lien librairie]

`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);
}
?>
```

6.43.21 ifx_free_blob() : Supprime un objet BLOB.

`int` **ifx_free_blob** (`int bid`)
[lien librairie]

`ifx_free_blob` supprime l'objet BLOB `bid` . `ifx_free_blob` retourne `FALSE` en cas d'erreur, et sinon `TRUE` .

6.43.22 ifx_free_char() : Supprime un objet char.

`int` **ifx_free_char** (`int bid`)
[lien librairie]

`ifx_free_char` supprime l'objet char `bid` . `ifx_free_char` retourne FALSE en cas d'erreur, et sinon TRUE .

6.43.23 ifx_free_result() : Libère les ressources prises par un résultat.

`int` **ifx_free_result** (`int result_id`)
[lien librairie]

`ifx_free_result` libère les ressources prises par le résultat `result_id` . `ifx_free_result` retourne FALSE en cas d'erreur.

6.43.24 ifx_get_blob() : Retourne le contenu d'un objet BLOB.

`int` **ifx_get_blob** (`int bid`)
[lien librairie]

`ifx_get_blob` retourne le contenu de l'objet BLOB associé à `bid` .

6.43.25 ifx_get_char() : Retourne le contenu d'un objet char.

`int` **ifx_get_char** (`int bid`)
[lien librairie]

`ifx_get_char` retourne le contenu de l'objet associé à l'identifiant `bid` .

6.43.26 ifx_getsqlca() : Retourne le contenu de la variable sqlca.sqlerrd[0..5] après une requête.

`array` **ifx_getsqlca** (`int result_id`)
[lien librairie]

`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.sqlerrdx')", 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]

```php
<?php
/* On suppose que la première colonne d'une table 'quelconque' est un numéro de série */
$qid = ifx_query("insert into sometable values(0, '2nd column', 'another column' ", $connid);
if (! $qid) {
//    ... erreur ...
}
$sqlca = ifx_getsqlca ($qid);
$serial_value = $sqlca["sqlerrd1"];
echo "Le numéro de série de la valeur insérée est : " . $serial_value . "<br>\n";
?>
```

6.43.27 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`*)
[lien librairie]

`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
<?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);
?>
```

6.43.28 ifx_nullformat() : Modifie le mode par défaut de lecture des valeurs.

`void` **ifx_nullformat** (`int mode`)
[lien librairie]

`ifx_nullformat` modifie le mode par défaut de lecture des valeurs. Le mode "0" retourne "", et le mode "1" retourne "NULL".

6.43.29 ifx_num_fields() : Retourne le nombre de colonnes dans une requête.

`int` **ifx_num_fields** (`int result_id`)
[lien librairie]

`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.

6.43.30 ifx_num_rows() : Compte le nombre de ligne déjà lues dans un résultat.

`int` **ifx_num_rows** (`int result_id`)
[lien librairie]

`ifx_num_rows` compte le nombre de ligne déjà lues dans le résultat `result_id` après `ifx_query` ou `ifx_do` .

6.43.31 ifx_pconnect() : Ouvre une connexion persistante à un serveur Informix.

`int` **ifx_pconnect** (*`string database`* , *`string userid`* , *`string password`*)
[lien librairie]

`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` .

6.43.32 ifx_prepare() : Prépare une requête SQL pour l'exécution.

`int` **ifx_prepare** (`string query` , `int conn_id` , *`int cursor_def`* , `mixed blobidarray`)
[lien librairie]

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 .

6.43.33 ifx_query() : Envoie une requête Informix.

int **ifx_query** (string query , *int link_identifier* , *int cursor_type* , *mixed blobidarray*)
[lien librairie]

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 par4 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

```
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"

```
<?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;
// éxécute la requête
$query = "insert into catalog (stock_num, manu_code, " .
         "cat_descr,cat_picture) values(1,'HRO',?,?)";
$res_id = ifx_query($query, $conn_id, $BLOBidarray);
if (! $res_id) {
//  ... erreur ...
}
// libération du résultat
ifx_free_result($res_id);
?>
```

6.43.34 ifx_textasvarchar() : Choisit le mode par défaut des objets text.

`void` **ifx_textasvarchar** (`int mode`)
[lien librairie]

`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.

6.43.35 ifx_update_blob() : Modifie le contenu d'un objet BLOB.

`boolean` **ifx_update_blob** (`int bid` , `string content`)
[lien librairie]

`ifx_update_blob` modifie le contenu de l'objet BLOB repéré par sont identifiant `bid` . `content` est une chaîne contenant les nouvelles données. `ifx_update_blob` retourne FALSE en cas d'erreur, et sinon, TRUE .

6.43.36 ifx_update_char() : Modifie le contenu d'un objet char.

`int` **ifx_update_char** (`int bid` , `string content`)
[lien librairie]

`ifx_update_char` modifie le contenu de l'objet char repéré par son identifiant `bid` . `content` est une chaîne avec les nouvelles données. `ifx_update_char` retourne FALSE en cas d'erreur, et sinon, TRUE .

6.43.37 ifxus_close_slob() : Ferme un objet SLOB.

`int` **ifxus_close_slob** (`int bid`)
[lien librairie]

`ifxus_close_slob` ferme l'objet SLOB représenté par son identifiant `bid` . `ifxus_close_slob` retourne FALSE en cas d'erreur, et sinon, TRUE .

6.43.38 ifxus_create_slob() : Crée un objet SLOB et l'ouvre.

`int` **ifxus_create_slob** (`int mode`)
[lien librairie]

`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.

6.43.39 ifx_free_slob() : Supprime un objet SLOB.

`int` **ifxus_free_slob** (`int bid`)
[lien librairie]

`ifxus_free_slob` supprime un objet SLOB. `bid` est l'identifiant de l'objet SLOB. `ifxus_free_slob` retourne FALSE en cas d'erreur, et sinon TRUE .

6.43.40 ifxus_open_slob() : Ouvre un objet SLOB.

`int` **ifxus_open_slob** (`long bid` , `int mode`)
[lien librairie]

`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 FALSE en cas d'erreur, et sinon, l'identifiant du nouvel objet.

6.43.41 ifxus_read_slob() : Lit n bytes d'un objet SLOB.

`int` **ifxus_read_slob** (`long bid` , `long nbytes`)
[lien librairie]

`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.

6.43.42 ifxus_seek_slob() : Fixe le fichier courant, ou la position courante.

`int` **ifxus_seek_slob** (`long bid` , `int mode` , `long offset`)
[lien librairie]

`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.

6.43.43 ifxus_tell_slob() : Retourne le fichier courant, ou la position courante.

`int` **ifxus_tell_slob** (`long bid`)
[lien librairie]

`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.

6.43.44 ifxus_write_slob() : Ecrit une chaîne dans un objet SLOB.

`int` **ifxus_write_slob** (`long bid` , `string content`)
[lien librairie]

`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.

6.44 InterBase

6.44.1 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 `php.ini` la directive suivante : `magic_quotes_sybase = On`

6.44.2 Pré−requis

6.44.3 Installation

6.44.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
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

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.44.5 Types de ressources

6.44.6 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 (nombre ˆ virgule flottante)
IBASE_TEXT (nombre ˆ virgule flottante)
IBASE_UNIXTIME (nombre ˆ virgule flottante)
IBASE_READ (nombre ˆ virgule flottante)
IBASE_COMMITTED (nombre ˆ virgule flottante)
IBASE_CONSISTENCY (nombre ˆ virgule flottante)
IBASE_NOWAIT (nombre ˆ virgule flottante)
IBASE_TIMESTAMP (nombre ˆ virgule flottante)
IBASE_DATE (nombre ˆ virgule flottante)
IBASE_TIME (nombre ˆ virgule flottante)

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- ibase_close
- ibase_commit
- ibase_connect
- ibase_errmsg
- ibase_execute
- ibase_fetch_object
- ibase_fetch_row
- ibase_field_info
- ibase_free_query
- ibase_free_result
- ibase_num_fields
- ibase_pconnect
- ibase_prepare
- ibase_query
- ibase_rollback
- ibase_timefmt
- ibase_trans

6.44.8 ibase_commit() : Valide une transaction

`int` **ibase_commit** (*`resource link_identifier`* , `resource trans_number`)
[lien librairie]

`ibase_commit` valide la transaction `trans_number` , qui a été préparée avec `ibase_trans` .

6.44.9 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*)
[lien librairie]

ibase_connect établit une connexin avec un serveur InterBase. database doit être un chemin valide jusqu'à un fichier de base de donnés sur le serveur sur lequel il résde. 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 spécifié dans les directives de configuration du PHP ibase.default_user et ibase.default_password. charset est le jeu de caractère par défaut de la base. buffers est le nombre de buffer de base à allouer pour le cache serveur. S'il est passé à 0 ou omis, le serveur choisira de lui−même. dialect sélectionne le dialecte SQL pour les requêtes exécutées avec cette connexion, et par défaut, il utilise le meilleur dialecte disponible.

Si un deuxième appel est fait avec ibase_connect , en passant les mêmes arguments, une nouvelle connexion ne sera pas ouverte, mais la connexion déjà ouverte sera retournée. La connexion sera fermée dès que le script se termine, à moins qu'elle ne soit fermée explicitement avec ibase_close , durant le script.

Exemple ibase_connect

```
<?php
    $dbh = ibase_connect($host, $username, $password);
    $stmt = 'SELECT * FROM tblname';
    $sth = ibase_query($dbh, $stmt);
    while ($row = ibase_fetch_object($sth)) {
        print $row->email . "\n";
    }
    ibase_close($dbh);
?>
```

Note

buffers a été ajouté en PHP 4−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_pconnect .

6.44.10 ibase_errmsg() : Retourne un message d'erreur

string **ibase_errmsg** (void)
[lien librairie]

ibase_errmsg retourne une chaîne contenant les messages d'erreurs.

6.44.11 ibase_execute() : Exécute une requête préparée.

`resource` **ibase_execute** (`int query` , *`int bind_args`*)
[lien librairie]

`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.

```
<?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);
    }
?>
```

6.44.12 ibase_fetch_object() : Lit une ligne dans une base Interbase dans un objet.

`object` **ibase_fetch_object** (`resource result_id`)
[lien librairie]

`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` .

```
<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` .

6.44.13 ibase_fetch_row() : Lit une ligne dans une base Interbase

`array` **ibase_fetch_row** (`resource result_identifier`)
[lien librairie]

`ibase_fetch_row` retourne la prochaine ligne spécifiée dans le résultat obtenu de `ibase_query` .

6.44.14 ibase_field_info() : Lit les informations sur un champs

`array` **ibase_field_info** (`resource result , int field_number`)
[lien librairie]

`ibase_field_info` retourne un tableau contenant les informations sur le champs numéro `field_number` après une requête de SELECT. Le tableau contient les index name (nom), alias, relation, length (taille), type.

```
<?php
// helio@helio.com.br  08-Dec-2000 02:53
$rs=ibase_query("Select * from unetable");
$coln = ibase_num_fields($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";
    }
?>
```

6.44.15 ibase_free_query() : Libère la mémoire réservée par une requête préparée.

`int` **ibase_free_query** (`resource query`)
[lien librairie]

`ibase_free_query` libère la mémoire réservée par une requête préparée par `ibase_prepare` .

6.44.16 ibase_free_result() : Libère un résultat.

`int` **ibase_free_result** (`resource result_identifier`)
[lien librairie]

`ibase_free_result` libère les ressources liées au résultat `result_identifier` .

6.44.17 ibase_num_fields() : Retourne le nombre de colonnes dans un résultat.

`int` **ibase_num_fields** (`resource result_id`)
[lien librairie]

`ibase_num_fields` retourne le nombre de colonnes dans un résultat.

```
<?php
    $dbh = ibase_connect ($host, $username, $password);
    $stmt = 'SELECT * FROM tblname';
    $sth = ibase_query($dbh, $stmt);
```

```
    if (ibase_num_fields($sth) > 0) {
        while ($row = ibase_fetch_object ($sth)) {
            print $row->email . "\n";
        }
    } else {
        die ("Aucun résultat dans votre requête");
    }
    ibase_close ($dbh);
?>
```

Note
`ibase_timefmt` ne fonctionne pas encore sous PHP4.

6.44.18 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`*)
[lien librairie]

`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 (peristante) 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−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_connect` pour plus de détails sur les arguments de cette fonction.

6.44.19 ibase_prepare() : Prépare une requête pour lier les paramètres et l'éxécuter ultérieurement.

`int` **ibase_prepare** (*`resource link_identifier`* , `string query`)
[lien librairie]

`ibase_prepare` prépare une requête pour l'éxécuter

6.44.20 ibase_query() : Exécute une requête sur une base Interbase

`resource` **ibase_query** (*`resource link_identifier`* , `string query` , *`int bind_args`*)
[lien librairie]

`ibase_query` exécute une requête sur une base Interbase, et retourne un identifiant de résultat, à utiliser avec `ibase_fetch_row` , `ibase_free_result` et/ou `ibase_free_query` .

Note
Bien que ces fonctions supportent la liaison de variables avec des paramètres de requêtes, il n'y a pas d'intérêt spécial à les utiliser. Pour des exemples grandeur réelle, voyez `ibase_prepare` et `ibase_execute` .

6.44.21 ibase_rollback() : Annule une transaction

`int` **ibase_rollback** (*`resource link_identifier`* , `resource trans_number`)
[lien librairie]

`ibase_rollback` annule la transaction `trans_number` qui a été préparée avec `ibase_trans` .

6.44.22 ibase_timefmt() : Fixe le format de date pour les prochaines requêtes.

`int` **ibase_timefmt** (`string format` , *`int columntype`*)
[lien librairie]

`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.

```
<?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.

6.44.23 ibase_trans() : Prépare une transaction

`resource` **ibase_trans** (*`int trans_args`* , *`resource link_identifier`*)
[lien librairie]

`ibase_trans` prépare une transaction

6.45 Ingres II

6.45.1 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 utiliser ce module à vos risques et périls.

6.45.2 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.

6.45.3 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=REP` 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.

6.45.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
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

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.45.5 Types de ressources

Cette extension ne définit aucune ressource.

6.45.6 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` (nombre ˆ virgule flottante)
`INGRES_NUM` (nombre ˆ virgule flottante)
`INGRES_BOTH` (nombre ˆ virgule flottante)

- 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

6.45.8 ingres_close() : Ferme une connexion à un serveur Ingres.

`boolean` **ingres_close** (*`resource link`*)
[lien librairie]

`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` .

6.45.9 ingres_commit() : Valide une transaction.

`boolean` **ingres_commit** (*`resource link`*)
[lien librairie]

`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` .

6.45.10 ingres_connect() : Ouvre une connexion à un serveur Ingres.

`resource` **ingres_connect** (*`string database`* , *`string username`* , *`string password`*)
[lien librairie]

`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 paramétres 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` .

6.45.11 ingres_fetch_array() : Récupère une ligne de résultat dans un tableau.

`array` **ingres_fetch_array** (*`int result_type`* , *`resource link`*)
[lien librairie]

`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.

```
<?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 .

6.45.12 ingres_fetch_object() : Récupère une ligne de résultat dans un objet.

object **ingres_fetch_object** (*int result_type* , *resource link*)
[lien librairie]

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 .

6.45.13 ingres_fetch_row() : Récupère une ligne de résultat dans un tableau énuméré.

array **ingres_fetch_row** (*resource link*)
[lien librairie]

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 pour 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` .

6.45.14 ingres_field_length() : Retourne la taille d'un champ.

`int` **ingres_field_length** (`int index` , *`resource link`*)
[lien librairie]

`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` .

6.45.15 ingres_field_name() : Retourne le nom d'un champ dans le résultat d'une requête.

`string` **ingres_field_name** (`int index` , *`resource link`*)
[lien librairie]

`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` .

6.45.16 ingres_field_nullable() : Teste si un champ est annulable.

`boolean` **ingres_field_nullable** (`int index` , *`resource link`*)
[lien librairie]

`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 .

6.45.17 ingres_field_precision() : Retourne la précision d'un champ.

int **ingres_field_precision** (int index , *resource link*)
[lien librairie]

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 .

6.45.18 ingres_field_scale() : Retourne l'échelle d'un champ.

int **ingres_field_scale** (int index , *resource link*)
[lien librairie]

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 .

6.45.19 ingres_field_type() : Retourne le type d'un champ dans le résultat d'une requête.

string **ingres_field_type** (int index , *resource link*)
[lien librairie]

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 .

6.45.20 ingres_num_fields() : Retourne le nombre de champs renvoyés par la dernière requête.

int **ingres_num_fields** (*resource link*)
[lien librairie]

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 .

6.45.21 ingres_num_rows() : Retourne le nombre de lignes affectées ou retournées par la dernière requête.

int **ingres_num_rows** (*resource link*)
[lien librairie]

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 .

6.45.22 ingres_pconnect() : Ouvre une connexion persistante à un serveur Ingres.

resource **ingres_pconnect** (*string database* , *string username* , *string password*)
[lien librairie]

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 .

6.45.23 ingres_query() : Envoie une requête SQL à un serveur Ingres II.

boolean **ingres_query** (string query , *resource link*)
[lien librairie]

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 pour 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 .

6.45.24 ingres_rollback() : Annule une transaction.

`boolean` **ingres_rollback** (*`resource link`*)
[lien librairie]

`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` .

6.46 IRC

Client pour IRC : Internet Relay Chat

Basé sur IRCG , de Sascha Schumann.

- ircg_channel_mode
- ircg_disconnect
- 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_notice
- ircg_part
- ircg_pconnect
- ircg_register_format_messages
- ircg_set_current
- ircg_topic
- ircg_whois

6.46.2 ircg_disconnect() : Ferme la connexion avec un serveur

`boolean` **ircg_disconnect** (`resource connection` , `string reason`)
[lien librairie]

`ircg_disconnect` ferme la connexion ouverter précédemment avec un serveur grâce à la fonction `ircg-pconnect` , et représentée par `connection` .

Voir aussi `ircg_pconnect` .

6.46.3 ircg_html_encode() : Prépare l'affichage pour le HTML

`boolean` **ircg_html_encode** (`string html_string`)
[lien librairie]

...

Voir aussi

6.46.4 ircg_ignore_add() : Ajoute un utilisateur sur la liste des utilisateurs indésirables

`boolean` **ircg_ignore_add** (`resource connection` , `string nick`)
[lien librairie]

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 .

6.46.5 ircg_ignore_del() : Supprime un utilisateur de la liste des utilisateurs indésirables

boolean **ircg_ignore_del** (resource connection , string nick)
[lien librairie]

ircg_ignore_del supprime l'utilisateur nick de la liste des utilisateurs indésirables, sur le serveur connection .

Voir aussi ircg_ignore_add .

6.46.6 ircg_is_conn_alive() : Vérifie l'état de la connexion

boolean **ircg_is_conn_alive** (resource connection)
[lien librairie]

ircg_is_conn_alive retourne TRUE si la connexion connection est toujours active et fonctionnelle, et FALSE si la connexion n'est plus disponible.

6.46.7 ircg_join() : Rejoint un canal IRC

boolean **ircg_join** (resource connection , string channel)
[lien librairie]

ircg_join rejoint le canal channel sur le serveur sur le serveur représenté par connection .

6.46.8 ircg_kick() : Expulse un utilisateur d'un canal

boolean **ircg_kick** (resource connection , string channel , string nick , string reason)
[lien librairie]

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.

6.46.9 ircg_lookup_format_messages() : Sélectionne un format d'affichage pour les messages IRC

boolean **ircg_lookup_format_messages** (string name)
[lien librairie]

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 .

6.46.10 ircg_msg() : Envoie un message à un canal ou un utilisateur

boolean **ircg_msg** (resource connection , string recipient , string message , *boolean suppress*)
[lien librairie]

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 .

6.46.11 ircg_nick() : Change de nom sur le serveur

boolean **ircg_nick** (resource connection , string nick)
[lien librairie]

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

6.46.12 ircg_notice() : Envoie une note (notice) à un utilisateur

boolean **ircg_notice** (resource connection , string , string message)
[lien librairie]

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.

6.46.13 ircg_part() : Quitte le canal

boolean **ircg_part** (resource connection , string channel)
[lien librairie]

ircg_part quitte le canal channel sur le serveur sur le serveur représenté par connection .

6.46.14 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*)
[lien librairie]

`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` .

6.46.15 ircg_register_format_messages() : Enregistre un nouveau format d'affichage des messages IRC

`boolean` **ircg_register_format_messages** (`string name` , `array messages`)
[lien librairie]

`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
- Elément 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` .

6.46.16 ircg_set_current() : Prépare la connexion courante pour l'affichage

`boolean` **ircg_set_current** (`resource connection`)
[lien librairie]

`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` et créée par `ircg_lookup_format_messages` .

Voir aussi `ircg_register_format_messages` et `ircg_lookup_format_messages` .

6.46.17 ircg_topic() : Modifie le sujet (topic) d'un canal

`boolean` **ircg_topic** (`resource connection` , `string channel` , `string new_topic`)
[lien librairie]

`ircg_topic` change le sujet du canal de `channel` en `new_topic` , sur le serveur `connection` .

6.46.18 ircg_whois() : Requiert les informations sur un utilisateur

`boolean` **ircg_whois** (`resource connection` , `string nick`)
[lien librairie]

Voir aussi

6.47 Java

6.47.1 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 utiliser ce module à vos risques et périls.

6.47.2 Pré−requis

Vous avez besoin d'une Machine virtuelle Java installée sur votre machine pour utiliser cette extension.

6.47.3 Installation

Les instructions d'installation de cette extension sont fournis dans `php4/ext/java/README` .

6.47.4 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	JAVALIB	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.47.5 Types de ressources

Cette extension ne définit aucune ressource.

6.47.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

6.47.7 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 meilleur fonction à utiliser.

  De plus, les noms de méthodes ne sont pas sensibles à la casse en PHP, ce qui augmente le nombre de conflits potentiels.

  Une fois qu'une méthode est sélectionnée, les paramètres sont transtypés, avec une perte d'information potentielle non négligeable (par exemple, les nombres à virgules flottante en double précisions seront convertis en booléen).
- Traditionnellement en PHP, les tableaux et les tables de hashage peuvent être interchangées, et fonctionnent de la même façon. Notez que les tables de hashage de PHP ne peuvent être indexées qu'avec des entiers ou des chaînes, et que le type primitif de tableau de Java ne peut comporter de trous dans les index. Notez aussi que les valeurs sont passées par valeur, ce qui peut être coûteux en mémoire et en temps.

6.47.8 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

6.47.10 java_last_exception_get() : Lit la dernière exception Java

`exception` **java_last_exception_get** `( void )`
[lien librairie]

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 &#62;)
  $result = @$stack->pop();
  $ex = java_last_exception_get();
  if ($ex) print $ex->toString();
  // Efface la dernière exception
  java_last_exception_clear();
?>
```

6.48 LDAP

6.48.1 Introduction

LDAP siginifie : 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
- University of Michigan
- OpenLDAP Project
- LDAP World

Le SDK Netscape contient un guide du programmeur très utile, au format HTML (et en anglais).

6.48.2 Pré−requis

Vous devez télécharger et compiler les librairies clients LDAP, soit celles de University of Michigan ldap−3.3, soit le Netscape Directory SDK 3.0.

6.48.3 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.

6.48.4 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` .

6.48.5 Types de ressources

Cette extension ne définit aucune ressource.

6.48.6 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_TWOWAY_AUTH` (entier)

6.48.7 Exemple complet de code LDAP

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>";
}
?>
```

6.48.8 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
- Exemple complet de code LDAP
- 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

6.48.10 ldap_add() : Ajoute une entrée dans un dossier LDAP

`bool` **ldap_add** (`resource link_identifier` , `string dn` , `array entry`)
[lien librairie]

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`ldap_add` sert à ajouter une entrée dans un dossier LDAP. `dn` est l'identification complète de l'entrée qui sera ajoutée, et `entry` est un tableau avec les informations sur la nouvelle entrée. Ces valeurs sont indexées individuellement. En cas de valeurs multiples pour un attribut, elle sont indexées numériquement, à partir de 0.

```
<?php $entree["attribut1"] = "valeur"; $entree["attribut2"][0] = "valeur1"; $entree["attribut2"][1] = "valeur2"; ?>
```

Exemple complet avec identification

```
<?php
$ds=ldap_connect("localhost");  // on suppose que le serveur LDAP est sur le serveur local

if ($ds) {
    // Connexion avec une identitée 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";
}
?>
```

6.48.11 ldap_bind() : Bind to LDAP directory

`bool` **ldap_bind** (`resource link_identifier` , *`string bind_rdn`* , *`string bind_password`*)
[lien librairie]

ldap_bind s'autentifie auprès du serveur LDAP link_identifier , avec le nom d'utilisateur bind_rdn et le mot de passe bind_password . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_bind effecute une opération de bind avec le serveur. bind_rdn et bind_password sont optionnels. Si ils sont omis, une tentative de connexion anonyme est tentée.

6.48.12 ldap_close() : Ferme une connexion au serveur LDAP

bool **ldap_close** (resource link_identifier)
[lien librairie]

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_close ferme la connexion au serveur LDAP, représentée par link_identifier .

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 .

6.48.13 ldap_compare() : Compare une entrée avec des valeurs d'attributs

bool **ldap_compare** (resource link_identifier , string dn , string attribute , string value)
[lien librairie]

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 entré 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.

6.48.14 ldap_connect() : Se connecte à un serveur LDAP.

`resource` **ldap_connect** (*`string hostname`* , *`int port`*)
[lien librairie]

`ldap_connect` retourne un identifiant positif de serveur LDAP en cas de succès, ou bien TRUE 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. Le support des URL et SSL a été ajouté en PHP 4.0.4.

6.48.15 ldap_count_entries() : Compte le nombre d'entrées après une recherche

`int` **ldap_count_entries** (`resource link_identifier` , `resource result_identifier`)
[lien librairie]

ldap_count_entries retourne le nombre d'entrées trouvées dans le résultat `result_identifier` , sur la connexion `link_identifier` , ou FALSE en cas d'erreur.

`ldap_count_entries` retourne le nombre d'entrées stockées dans le résultat de la recherche précédente. `result_identifier` représente un résultat interne LDAP.

6.48.16 ldap_delete() : Efface une entrée dans un dossier

`bool` **ldap_delete** (`resource link_identifier` , `string dn`)
[lien librairie]

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`ldap_delete` efface l'entrée identifiée par son `dn` sur le serveur identifié par `link_identifier` .

6.48.17 ldap_dn2ufn() : Convert it un DN en format UFN (User Friendly Naming)

`string` **ldap_dn2ufn** (`string dn`)
[lien librairie]

`ldap_dn2ufn` convertit le DN `dn` dans un format plus lisible humainement, en supprimant les types des noms.

6.48.18 ldap_err2str() : Convertit un numéro d'erreur LDAP en message d'erreur

`string` **ldap_err2str** (`int errno`)
[lien librairie]

`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 librairies 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.

Enumerer 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` .

6.48.19 ldap_errno() : Retourne le numéro d'erreur LDAP de la dernière commande exécutée.

`int` **ldap_errno** (`resource link_id`)
[lien librairie]

`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_id` . 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 allons intercepter.
  $ld = ldap_connect("localhost");
  $bind = ldap_bind($ld);
// Erreur de syntaxe dans l'expression du filtre (errno 87),
// ce doit être "objectclass=*"
  $res =  @ldap_search($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` .

6.48.20 ldap_error() : Retourne le message LDAP de la dernière commande LDAP.

`string` **ldap_error** (`resource link_id`)
[lien librairie]

`ldap_error` retourne un message d'erreur.

`ldap_error` retourne le message d'erreur lié à la connexion `link_id` . Même si les numéros d'erreur LDAP sont standardisés, différentes librairies 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` .

6.48.21 ldap_explode_dn() : Sépare les différents composants d'un DN

`array` **ldap_explode_dn** (`string dn` , `int with_attrib`)
[lien librairie]

`ldap_explode_dn` sert à extraire les différents composants du DN `dn` . Chaque composant s'appelle un Nom Distingué Relatif (Relative Distinguished Name ou encore RDN). `ldap_explode_dn` retourne un tableau de tous ces composants. L'argument `with_attrib` sert à préciser si les RDN sont retournés seuls, ou bien avec leurs attributs. Pour obtenir les attributs en même temps que les RDN (au format attribut=valeur), donnez à `with_attrib` la valeur de 0 et sinon, donnez lui la valeur de 1.

6.48.22 ldap_first_attribute() : Return first attribute

`string` **ldap_first_attribute** (`resource link_identifier` , `resource result_entry_identifier` , `int ber_identifier`)
[lien librairie]

`ldap_first_attribute` retourne le premier attribut de l'entrée `result_entry_identifier` , sur le serveur `link_identifier` , ou FALSE en cas d'erreur.

Similairement à la lecture des entrées, les attributs sont lis les uns après les auters, pour une entrée. `ldap_first_attribute` retourne le premier attribut de l'entrée représentée par `result_entry_identifier` . Les autres attributs sont lus grâce à la fonction `ldap_next_attribute` , appelée aussi souvent que nécessaire. Le paramètre `ber_identifier` est l'identifiant de pointeur en mémoire locale. Il est passé par référence. Le même paramètre `ber_identifier` est passé à la fonction `ldap_next_attribute` , qui modifiera ce pointeur.

Voir aussi `ldap_get_attributes` .

6.48.23 ldap_first_entry() : Retourne la première entrée

`resource` **ldap_first_entry** (`resource link_identifier` , `resource result_identifier`)
[lien librairie]

`ldap_first_entry` retourne la première entrée du résultat `result_identifier` , pour la connexion `link_identifier` , 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 `lap_next_entry` , en appelant cette fonction aussi souvent que nécessaire.

Voir aussi `ldap_get_entries` .

6.48.24 ldap_first_reference() : Return first reference

`resource` **ldap_first_reference** (`resource link` , `resource result`)

[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.48.25 ldap_free_result() : Free result memory

`bool` **ldap_free_result** (`resource result_identifier`)
[lien librairie]

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`ldap_free_result` libère toute la mémoire allouée en interne pour stocker le résultat `result_identifier` . Si l'appel de cette fonction est omis, toute la mémoire sera libérée automatiquement à la fin du script.

Typqieuemtn, 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.

6.48.26 ldap_get_attributes() : Lit les attributs d'une entrée

`array` **ldap_get_attributes** (`resource link_identifier` , `resource result_entry_identifier`)
[lien librairie]

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.

```
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
```

Affiche la lsite des attributs d'une entrée

```
<?php
// $ds is the link identifier for the directory

// $sr is a valid search result from a prior call to
```

```
// one of the ldap directory search calls

  $entry = ldap_first_entry($ds, $sr);

  $attrs = ldap_get_attributes($ds, $entry);

  echo $attrs["count"]." attributs définissent cette entrée : <P>";

  for ($i=0; $i<$attrs["count"]; $i++){
    echo $attrs[$i]."<br>";
  }
?>
```

Voir aussi `ldap_first_attribute` et `ldap_next_attribute` .

6.48.27 ldap_get_dn() : Lit le DN d'une entrée

`string` **ldap_get_dn** (`resource link_identifier` , `resource result_entry_identifier`)
[lien librairie]

`ldap_get_dn` lit le DN de l'entrée courante dans le résultat de recherche LDAP `result_entry_identifier` , ou FALSE , en cas d'erreur.

`ldap_get_dn` sert à lire le DN dans un résultat de recherche.

6.48.28 ldap_get_entries() : Lit toutes les entrées du résultat

`array` **ldap_get_entries** (`resource link_identifier` , `resource result_identifier`)
[lien librairie]

`ldap_get_entries` retourne toutes les entrées du résulat `result_identifier` , sous la forme d'un tableau multi−dimensionnel, ou FALSE en casd 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.

```
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−ième entrée
return_value[n]["attribute"][m] = m−ième valeur de la n−ième entrée
```

Voir aussi `ldap_first_entry` et `ldap_next_entry` .

6.48.29 ldap_get_option() : Lit/écrit la valeur courante d'une option

`bool` **ldap_get_option** (`resource link_identifier` , `int option` , `mixed retval`)
[lien librairie]

`ldap_get_option` modifie la valeur de l'option `option` à la valeur `retval` pour la connexion `link_identifier` . 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` .

6.48.30 ldap_get_values_len() : Lit toutes les valeurs binaires d'une entrée

`array` **ldap_get_values_len** (`resource link_identifier` , `resource result_entry_identifier` , `string attribute`)
[lien librairie]

`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_identifier` . L'entré utilisée et l'entrée courante du résultat de recherche LDAP `result_entry_identifier` . 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.

6.48.31 ldap_get_values() : Lit toutes les valeurs d'une entrée

`array` **ldap_get_values** (`resource link_identifier` , `resource result_entry_identifier` , `string attribute`)
[lien librairie]

`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'attibut `attribute` , dans l'entrée courante du résultat de recherche `result_entry_identifier` , pour la connexion `link_identifier` . 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_identifier` 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 addresse emails par personne, tout en n'utilisant qu'une étiquette "mail" :

```
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>";
```

6.48.32 ldap_list() : Recherche dans un niveau

`resource` **ldap_list** (`resource link_identifier` , `string base_dn` , `string filter` , *`array attributes`* , *`int attrsonly`* , *`int sizelimit`* , *`int timelimit`* , *`int deref`*)
[lien librairie]

`ldap_list` retourne un identfiant 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 paramtères optionnels. Voir ldap_search pour connaître le 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 paralleles. Reportez vous à ldap_search pour plus de détails.

6.48.33 ldap_mod_add() : Ajoute un attribut à l'entrée courante

bool **ldap_mod_add** (resource link_identifier , string dn , array entry)
[lien librairie]

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 .

6.48.34 ldap_mod_del() : Efface un attribut à l'entrée courante

bool **ldap_mod_del** (resource link_identifier , string dn , array entry)
[lien librairie]

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 .

6.48.35 ldap_mod_replace() : Remplace un attribut dans l'entrée courante

bool **ldap_mod_replace** (resoure link_identifier , string dn , array entry)
[lien librairie]

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 .

6.48.36 ldap_modify() : Modifie une entrée LDAP

bool **ldap_modify** (resource link_identifier , string dn , array entry)
[lien librairie]

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 .

6.48.37 ldap_next_attribute() : Lit l'attribut suivant.

string **ldap_next_attribute** (resource link_identifier , resource result_entry_identifier , int ber_identifier)
[lien librairie]

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_identifier . Il est passé par référence à la fonction. Le premier appel à ldap_next_attribute est fait avec le result_entry_identifier retourné par ldap_first_attribute .

Voir aussi ldap_get_attributes .

6.48.38 ldap_next_entry() : Lit la prochaine entrée

resource **ldap_next_entry** (resource link_identifier , resource result_entry_identifier)
[lien librairie]

ldap_next_entry retourne un identifiant d'entrée, pour la prochaine entrée dans le résultat result_entry_identifier . 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_identifier . 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_identifier est celui qui a été retourné par la fonction ldap_first_entry .

Voir aussi ldap_get_entries .

6.48.39 ldap_next_reference() : Lit la référence suivante

resource **ldap_next_reference** (resource link , resource entry)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.48.40 ldap_parse_reference() : Extrait les informations d'une référence d'entrée

bool **ldap_parse_reference** (resource link , resource entry , array referrals)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.48.41 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)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.48.42 ldap_read() : Lit une entrée

resource **ldap_read** (resource link_identifier , string base_dn , string filter , *array attributes* , *int attrsonly* , *int sizelimit* , *int timelimit* , *int deref*)
[lien librairie]

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 paralelles. Voyez `ldap_search` poour plus de détails.

6.48.43 ldap_rename() : Modifie le nom d'une entrée

`bool` **ldap_rename** (`resource link_identifier` , `string dn` , `string newrdn` , `string newparent` , `bool deleteoldrdn`)
[lien librairie]

`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.

6.48.44 ldap_search() : Recherche sur le serveur LDAP

`resource` **ldap_search** (`resource link_identifier` , `string base_dn` , `string filter` , *`array attributes`* , *`int attrsonly`* , *`int sizelimit`* , *`int timelimit`* , *`int deref`*)
[lien librairie]

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 paramtère 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 duréee 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 géré durant la rechercheé. Ce paramètre peut prendre l'une des valeurs suivantes :

- LDAP_DEREF_NEVER − (par défaut) les alias ne sont jamais déréférencés.
- LDAP_DEREF_SEARCHING − les alias sont déréférencés durant la recherche, mais pas durant la localisation du dossier de recherche.
- LDAP_DEREF_FINDING − les alias sont déréférencés durant la localisation du dossier de recherche, mais pas durant la recherche.
- LDAP_DEREF_ALWAYS − les alias sont toujours déréférencés.

Note
Ces paramètres optionnels ont été ajoutés en 4.0.2 : `attrsonly` , `sizelimit` , `timelimit` et `deref` .

Le filtre de recherche peut être simple ou avancé, et utiliser ces opérateurs booléen au format décrit dans la documentation LDAP (reportez vous à Netscape Directory SDK pour plus d'informations sur les filtres).

L'exemple ci−dessous lit le nom du service, le nom, le prénom et 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 paralelles. 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 etre de la même taille que le table de ressources de connexion : la première entrée du tableau de ressources sera utilisé avec le premier DN et le premier filtre, etc... Lorsque vous faite une recherche paralelle, 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 paralelle retourne un identifiant.

6.48.45 ldap_set_option() : Modifie la valeur d'une option LDAP

`bool` **ldap_set_option** (`resource link_identifier` , `int option` , `mixed newval`)
[lien librairie]

ldap_set_option modifie la valeur de l'option `option` en remplacant 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 constitués d'un **oid** identifiant le contrôle, une valeur optionnelle **value** , et un flag optionnel de criticalité (**criticality**). En PHP, un contrôle est définit 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 comem 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` .

6.48.46 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.48.47 ldap_sort() : Trie les entrées d'un résultat LDAP

`bool` **ldap_sort** (`resource link` , `resource result` , `string sortfilter`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.48.48 ldap_start_tls() : Démarre TLS

`bool` **ldap_start_tls** (`resource link`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.48.49 ldap_t61_to_8859() : Convertit les caractères t6 en caractères 8859

`string` **ldap_t61_to_8859** `( string value )`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.48.50 ldap_unbind() : Déconnecte d'un serveur LDAP

`bool` **ldap_unbind** `( resource link_identifier )`
[lien librairie]

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`ldap_unbind` assure la déconnexion du serveur LDAP.

6.49 Mail

6.49.1 Introduction

La fonction `mail` permet d'envoyer du mail.

6.49.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.49.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.49.4 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 de détails sur les 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 champs "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.

6.49.5 Types de ressources

Cette extension ne définit aucune ressource.

6.49.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- ezmlm_hash
- mail

6.49.8 mail() : Envoie de mail

`boolean` **mail** (`string to` , `string subject` , `string message` , *`string additional_headers`* , *`string additional_parameters`*)
[lien librairie]

`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 http://www.zend.com/zend/spotlight/sendmimeemailpart1.php 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)

```
<?php
  mail("rasmus@lerdorf.on.ca", "Mon Sujet", "Ligne 1\nLigne 2\nLigne 3");
?>
```

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 `\r\n` pour séparer les en−têtes. Notez aussi que les en−têtes cc: et bcc: sont sensibles à la casse et doivent être écrits `Cc:` et `Bcc:` sous Win32.

Si le cinquième argument `additional_parameters` est fourni, PHP l'utilisera dans son appel du programme d'envoie 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("nobody@aol.com", "Le sujet", $message,
       "From: webmaster@$SERVER_NAME\nReply−To: webmaster@$SERVER_NAME\nX−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("nobody@aol.com", "the subject", $message, "From: webmaster@$SERVER_NAME", "−fwebmaster@$SERV
?>
```

Vous pouvez aussi utiliser des techniques simples de concaténations de chaînes pour construire des messages complexes :

Envoi de mail complexe.

```
<?php
  /* destinataire */
    $recipient .= "Mary <mary@u.college.edu>", " ; //remarquez les virgules
    $recipient .= "Kelly <kelly@u.college.edu>", ";
    $recipient .= "ronabop@php.net";
  /* sujet */
    $subject = "Rappel des anniversaires du mois d'août";
  /* message */
    $message .= "Le mail suivant inclut une table au format ASCII\n";
    $message .= "Jour \t\tMois \t\tAn\n";
    $message .= "3 \t\tAou\t\t1970\n";
    $message .= "17\t\tAou\t\t1973\n";
  /* Vous pouvez ajouter une signature */
    $message .= "−−\r\n";
  //Délimiteur de signature
    $message .= "Rappel d'anniversaire : copyleft par public domain";
  /* D'autres en−têtes : errors, From cc's, bcc's, etc */
    $headers .= "From: Rappel d'anniversaire <birthday@php.net>\n";
    $headers .= "X−Sender: <birthday@php.net>\n";
    $headers .= "X−Mailer: PHP\n"; // maileur
    $headers .= "X−Priority: 1\n"; //  Message urgent!
    $headers .= "Return−Path: <birthday@php.net>\n";  // Re−chemin de retour pour les erreurs
    $headers .= "Content−Type: text/html; charset=iso−8859−1\n" // Type MIME
    $headers .= "Cc:birthdayarchive@php.net\n"; // Champs CC
    $headers .= "Bcc:birthdaycheck@php.net, birthdaygifts@php.net"; // Champs BCCs
  /* et hop, à la poste */
    mail($recipient, $subject, $message, $headers);
```

```
?>
```

Note

Assurez−vous qu'il n'y ait aucune nouvelle ligne (ou d'autres espaces ou caractères blancs) dans les paramètres `to` ou `subject` , car cela peut avoir des effets secondaires irrationnels.

6.50 Traitement de mail

6.50.1 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 utiliser ce module à vos risques et périls.

- Introduction
- 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

6.50.3 mailparse_msg_create() : Retourne un analyseur destiné au traitement d'un message

`int` **mailparse_msg_create** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.4 mailparse_msg_extract_part_file() : Extrait et décode une section de message

`string` **mailparse_msg_extract_part_file** (`resource rfc2045` , `string filename` , *`string callbackfunc`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.5 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`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.6 mailparse_msg_free() : Détruit un analyseur créé par mailparse_msg_crea()

`void` **mailparse_msg_free** (`resource rfc2045buf`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.7 mailparse_msg_get_part_data() : Retourne un tableau associatif avec des informations sur le message

`array` **mailparse_msg_get_part_data** (`resource rfc2045`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.8 mailparse_msg_get_part() : Retourne une ressource sur une section d'un message MIME

`int` **mailparse_msg_get_part** (`resource rfc2045` , `string mimesection`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.9 mailparse_msg_get_structure() : Retourne un tableau avec les différentes sections MIME du message

`array` **mailparse_msg_get_structure** (`resource rfc2045`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.10 mailparse_msg_parse_file() : Analyse un fichier et retourne une ressource représentant la structure du fichier

`resource` **mailparse_msg_parse_file** (`string filename`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.11 mailparse_msg_parse() : Analyse incrémentalement des données dans un buffer

`void` **mailparse_msg_parse** (`resource rfc2045buf` , `string data`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.12 mailparse_rfc822_parse_addresses() : Analyse les adresses et retourne un tableau contenant ces données

`array` **mailparse_rfc822_parse_addresses** (`string addresses`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.13 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`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.50.14 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`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.51 Mathématiques

6.51.1 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 .

6.51.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.51.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.51.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.51.5 Types de ressources

Cette extension ne définit aucune ressource.

6.51.6 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 verisons de PHP jusqu'à 4.0.0. Toutes les autres constantes on été ajoutés 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 Prédefinies
- 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

6.51.8 Acos() : arc cosinus

`float` **acos** (`float arg`)
[lien librairie]

`acos` retourne l'arc cosinus de `arg` (`arg` en radians).

Voir aussi `asin` et `atan` .

6.51.9 acosh() : Arc cosinus hyperbolique

`float` **acosh** (`float arg`)
[lien librairie]

`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` , `asin` et `atan` .

6.51.10 Asin() : arc sinus

`float` **asin** (`float arg`)
[lien librairie]

`asin` retourne l'arc sinus de `arg` (`arg` en radians).

Voir aussi `acos` et `atan` .

6.51.11 asinh() : Arc sinus hyperbolique

`float` **asinh** (`float arg`)
[lien librairie]

`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 , acos et atan .

6.51.12 Atan() : arc tangent

float **atan** (float arg)
[lien librairie]

atan retourne l'arc tangent de arg (arg en radians).

Voir aussi acos et atan .

6.51.13 Atan2() : arc tangent de deux variables

float **atan2** (float y , float x)
[lien librairie]

atan2 retourne l'arc tangent de deux variables x et y . La formule est : '' arc tangent (y / x) '', et les signes des arguments sont utilisés pour déterminer le quadrant du résultat.

atan2 retourne un résultat en radians, entre −PI et PI (inclus).

Voir aussi acos et atan .

6.51.14 atanh() : Arc tangeant hyperbolique

float **atanh** (float arg)
[lien librairie]

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 , asin et acos .

6.51.15 base_convert() : Convertit un nombre entre des bases arbitraires.

string **base_convert** (string number , int frombase , int tobase)
[lien librairie]

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.

base_convert

```php
<?php
  $binary = base_convert($hexadecimal, 16, 2);
?>
```

6.51.16 BinDec() : Convertit de binaire en décimal

`int` **bindec** `( string binary_string )`
[lien librairie]

`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` .

6.51.17 Ceil() : Arrondit au nombre supérieur

`float` **ceil** `( float number )`
[lien librairie]

`ceil` retourne l'entier supérieur du nombre `number` . 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.

```php
<?php
  $x = ceil(4.25);
//  ce qui donne $x=5
?>
```

NOTE: `ceil` sous PHP/FI 2 retournait un nombre à virgule flottante. Utilisez: `$new = (float)ceil($number);` pour retrouver le comportement traditionnel.

Voir aussi `floor` et `round` .

6.51.18 Cos() : cosinus

`float` **cos** `( float arg )`
[lien librairie]

`cos` retourne le cosinus de `arg` (`arg` en radians).

Voir aussi `sin` et `tan` .

6.51.19 cosh() : Cosinus hyperbolic

`float` **cosh** (`float arg`)
[lien librairie]

`cosh` retourne le cosinus hyperbolique de `arg` , définit comme `(exp(arg) + exp(−arg))/2` .

Voir aussi `cos` , `acosh` , `sin` et `tan` .

6.51.20 DecBin() : Convertit de décimal en binaire

`string` **decbin** (`int number`)
[lien librairie]

`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` .

6.51.21 DecHex() : Convertit de décimal en hexadécimal

`string` **dechex** (`int number`)
[lien librairie]

`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` .

6.51.22 DecOct() : Convertit de décimal en octal

`string` **decoct** (`int number`)
[lien librairie]

`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` .

6.51.23 deg2rad() : Convertit un nombre de degrés en radians

`float` **deg2rad** (`float number`)
[lien librairie]

`deg2rad` convertit `number` de degrés en radians.

Voir aussi `rad2deg` .

6.51.24 Exp() : exponentielle

`float` **exp** (`float arg`)
[lien librairie]

`exp` retourne l'exponentielle de `arg` , c'est−à−dire e élevé à la puissance `arg` .

Voir aussi `pow` .

6.51.25 expm1() : Retourne le résultat de exp(number) − 1, calculé de manière précise, même si val est proche de 0

`float` **expm1** (`float number`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.51.26 Floor() : Arrondi à l'entier inférieur

`floats` **floor** (`float number`)
[lien librairie]

`floor` retourne l'entier inférieur du nombre `number` . La valeur retournée est un nombre à virgule flottante, (`float`) car ces nombres peuvent être plus grands que les entiers.

NOTE: `floor` sous PHP/FI retournait un float. Utilisez: `$new = (float)floor($number);` pour retrouver le comportement traditionnel.

Voir aussi `ceil` et `round` .

6.51.27 fmod() : Retourne le reste de la division

`float` **fmod** (`float x` , `float y`)
[lien librairie]

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
```

6.51.28 getrandmax() : Plus grande valeur aléatoire possible.

`int` **getrandmax** (`void`)
[lien librairie]

`getrandmax` retourne la plus grande valeur aléatoire possible retournée par `rand` .

Voir aussi `rand` , `srand` `mt_rand` , `mt_srand` et `mt_getrandmax` .

6.51.29 hexdec() : Convertit de hexadécimal en décimal

`int` **hexdec** (`string hex_string`)
[lien librairie]

`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−héxadecimal 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` .

6.51.30 hypot() : Retourne le résultat de sqrt(num1*num1 + num2*num2)

`float` **hypot** (`float num1` , `float num2`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.51.31 is_finite() : Indique si un nombre est fini

`bool` **is_finite** (`float val`)
[lien librairie]

`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.

6.51.32 is_infinite() :

`bool` **is_infinite** (`float val`)
[lien librairie]

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.

6.51.33 is_nan() : Indique si une valeur est un nombre

`bool` **is_nan** (`float val`)
[lien librairie]

`is_nan` retourne TRUE si `val` n'est pas un nombre, comme par exemple, le résultat de `acos(1.01)` .

6.51.34 lcg_value() : Générateur de congruence combinée linéaire

`float` **lcg_value** (`void`)
[lien librairie]

`lcg_value` retourne un nombre pseudo−aléatoire, compris entre 0 et 1. `lcg_value` combine deux générateurs de congruence, de périodes respectives `2^31 − 85` et `2^31 − 249` . La période de cette fonction est le produit de ces deux nombres premiers (soit `(2^31 − 85)*(2^31 − 249)`).

6.51.35 Log() : Logarithme naturel (népérien)

`float` **log** (`float arg` , *`float base`*)
[lien librairie]

`log` retourne le logarithme naturel (ou népérien) de `arg` . Si le paramètre optionnel `base` est spécifié, `log` retourn alors le logarithme en base `base` .

Note
Le paramètre `base` a été ajouté en PHP version 4.3.0.

6.51.36 Log10() : logarithme en base 10.

`float` **log10** (`float arg`)
[lien librairie]

`log10` retourne le logarithme en base 10 de `arg` .

6.51.37 log1p() : Retourne le résultat de log(1 + number), calculé de manière précise, même si val est proche de zéro.

`float` **log1p** (`float number`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.51.38 max() : La plus grande valeur.

`number` **max** (`number arg1` , `number arg2` , `number argn`)
[lien librairie]

`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.

6.51.39 min() : La plus petite valeur.

`number` **min** (`number arg1` , `number arg2` , `number argn`)
[lien librairie]

`min` retourne la plus petite valeur numérique parmi les valeurs passées en paramètres.

Si le premier paramètre est un tableau, `min` retourne la plus petite valeur de ce tableau. Si le premier paramètre est un entier, une chaîne ou un nombre à virgule flottante, `min` requiert au moins deux paramètres, et retournera alors le plus petit 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.

6.51.40 mt_getrandmax() : La plus grand valeur aléatoire possible.

`int` **mt_getrandmax** (`void`)
[lien librairie]

`mt_getrandmax` retourne la plus grand valeur aléatoire possible que peut retourner `mt_rand` .

Voir aussi `mt_rand` , `mt_srand` `rand` , `srand` et `getrandmax` .

6.51.41 mt_rand() : Génère une meilleure valeur aléatoire.

`int` **mt_rand** (`int min` , `int max`)
[lien librairie]

De nombreux générateurs de nombres aléatoires provenant de vieilles bibliothèques libcs ont des comportement 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 . Une version optimisée des sources de MT est disponible à http://www.scp.syr.edu/~marc/hawk/twister.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)` .

N'oubliez pas d'initialiser le générateur de nombres aléatoires avec `mt_srand` .

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` , `srand` , `rand` et `getrandmax` .

6.51.42 mt_srand() : Initialise une meilleure valeur aléatoire

`void` **mt_srand** (`int seed`)
[lien librairie]

`mt_srand` initialise une meilleure valeur aléatoire avec `seed` .

```
<?php
// initialise avec les microsecondes depuis la dernière seconde entière
  mt_srand((float) microtime()*1000000);
```

```
  $randval = mt_rand();
?>
```

Voir aussi mt_rand , mt_getrandmax , srand , rand et getrandmax .

6.51.43 OctDec() : Convertit d'octal en décimal.

int **octdec** (string octal_string)
[lien librairie]

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 .

6.51.44 pi() : Retourne la valeur de pi

float **pi** (void)
[lien librairie]

pi retourne la valeur de pi.

```
<?php
  echo pi();
//  3.1415926535898
?>
```

6.51.45 pow() : Puissance

number **pow** (number base , number exp)
[lien librairie]

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);
  // erreur
?>
```

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 .

6.51.46 rad2deg() : Convertit de radians en degrés

float **rad2deg** (float number)
[lien librairie]

rad2deg convertit number (supposé en radians) en degrés.

Voir aussi deg2rad .

6.51.47 rand() : Génère une valeur aléatoire.

int **rand** (int min , int max)
[lien librairie]

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) .

N'oubliez pas d'initialiser le générateur de nombres aléatoires avec srand .

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 , getrandmax , mt_rand , mt_srand et mt_getrandmax .

6.51.48 round() : Arrondi.

float **round** (float val , *int precision*)
[lien librairie]

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 .

```
<?php
  $foo = round( 3.4 );   // $foo == 3.0
  $foo = round( 3.5 );   // $foo == 4.0
  $foo = round( 3.6 );   // $foo == 4.0
```

```
?>
```

Note
Le paramètre `precision` est disponible uniquement en PHP 4.

Voir aussi `ceil` et `floor` .

6.51.49 Sin() : Sinus

`float` **sin** (`float arg`)
[lien librairie]

`sin` retourne le sinus de `arg` (`arg` in radians).

Voir aussi `cos` et `tan` .

6.51.50 sinh() : Sinyus hyperbolique

`float` **sinh** (`float arg`)
[lien librairie]

`sinh` retourne le sinus hyperbolique de `arg` , défini comme `(exp(arg) - exp(-arg))/2` .

Voir aussi `sin` , `asinh` , `cos` et `tan` .

6.51.51 Sqrt() : Racine carrée.

`float` **sqrt** (`float arg`)
[lien librairie]

`sqrt` retourne la racine carrée de `arg` .

6.51.52 srand() : Initialise le générateur de nombres aléatoires

`void` **srand** (`int seed`)
[lien librairie]

`srand` initialise le générateur de nombres aléatoires avec `seed` .

```
<?php
// initialise avec les microsecondes depuis la dernière seconde entière
  srand((float) microtime()*1000000);
  $randval = rand();
?>
```

Voir aussi `rand` , `getrandmax` , `mt_rand` , `mt_srand` et `mt_getrandmax` .

6.51.53 Tan() : Tangente

`float` **tan** (`float arg`)
[lien librairie]

`tan` retourne la tangente de `arg` (`arg` en radians).

Voir aussi `sin` et `cos` .

6.51.54 tanh() : Tangente hyperbolique

`float` **tanh** (`float arg`)
[lien librairie]

`tanh` retourne la tangeante hyperbolique de `arg` , définie comme `sinh(arg)/cosh(arg)` .

Voir aussi `tan` , `atanh` , `sin` et `cos` .

6.52 Chaînes de caractères multi−octets

6.52.1 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 erronnés 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

`Jeus 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.

6.52.2 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 expérimentelement 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. automatically.

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.

6.52.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
mbstring.language	NULL	PHP_INI_ALL
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.script_encoding	NULL	PHP_INI_ALL
mbstring.substitute_character	NULL	PHP_INI_ALL
mbstring.func_overload	"0"	PHP_INI_SYSTEM
mbstring.encoding_translation	"0"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à `ini_set` .

Voici de détails sur les directives de configuration.

- `mbstring.language` 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` active la détection et la traduction des données d'entrées HTTP vers le codage interne mbstring.
- `mbstring.internal_encoding` définit l'encodage interne par défaut.
- `mbstring.http_input` définit l'encodage de reception HTTP par défaut.
- `mbstring.http_output` définit l'encodage d'affichage HTTP par défaut.
- `mbstring.detect_order` définit l'ordre de détection des encodages par défaut. Voir aussi `mb_detect_order` .
- `mbstring.substitute_character` définit l'encodage de substitution par défaut : il est utilisé pour les caractères invalides.
- `mbstring.func_overload` 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 suffisament intelligent pour détecter les jeux de caractères dans le HTML. Si `charset` est activé dans les entêtes, 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 peux 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

```
;; Inactive 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

;; Acitvation de la converstion 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
```

6.52.4 Types de ressources

Cette extension ne définit aucune ressource.

6.52.5 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.

MB_OVERLOAD_MAIL (nombre ˆ virgule flottante)
MB_OVERLOAD_STRING (nombre ˆ virgule flottante)
MB_OVERLOAD_REGEX (nombre ˆ virgule flottante)

6.52.6 Entrées/Sorties HTTP

La conversion automatiques 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` .

  Inactive la conversion HTTP dans le `php.ini`

  ```
  ;; Inactive la conversion HTTP
  mbstring.http_input = pass
  ;; Inactive la conversion HTTP (PHP 4.3.0 ou plus récent)
  mbstring.encoding_translation = Off
  ```

  Lorsque vous utilisez PHP comme module Apache, il est possible d'annuler la configuration du `php.ini` pour chaque Virtual Host dans le fichier `httpd.conf` ou par dossier avec le fichier `.htaccess` . Reportez vous à la section de configuration ainsi qu'au manuel Apache.
- Sorties HTTP

  Il y a plusieurs moyens d'activer la conversion en sortie de script PHP. L'un d'entre eux utilise `php.ini` , un autre utilise `ob_start` avec la fonction `mb_output_handler` comme fonction de call−back.

  Note

  Pour les utilisateurs PHP3−i18n, le système de conversion de `mbstring` diffère de celui de

PHP3−i18n. Le jeu de caractère est converti avec un buffer de sortie.

Exemple de configuration de mbstring dans `php.ini`

```
;; Active la conversion de sortie pour 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 mbstring

```
<?php

// Active la conversion de caractère uniquement pour cette page

// Choisi le jeu de caractères SJIS
mb_http_output('SJIS');

// Commence la bufferisation et spécifie "mb_output_handler"
// comme fonction de callback
ob_start('mb_output_handler');
?>
```

6.52.7 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 caractèrs 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 fichiers `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.

6.52.8 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 par 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, certains 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 mbstring.func_overload	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`

6.52.9 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.

6.52.10 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 ressouces complémentaires :

- Unicode/UTF/UCS/etc

  `http://www.unicode.org/`
- Japanais/coréen/Chinois

  `ftp://ftp.ora.com/pub/examples/nutshell/ujip/doc/cjk.inf`

- Introduction
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- Entrées/Sorties HTTP
- Jeux de caractères supportés
- Exploitation des chaînes multi−octets en PHP
- Cas des caractères japonais
- Références
- 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_http_input
- mb_http_output
- mb_internal_encoding
- mb_language
- mb_output_handler
- mb_parse_str
- mb_preferred_mime_name
- mb_send_mail
- mb_strcut
- mb_strimwidth
- mb_strlen
- mb_strpos
- mb_strrpos

- mb_strwidth
- mb_substitute_character
- mb_substr

6.52.12 mb_convert_kana() : Convertit entre les différents "kana"

`string` **mb_convert_kana** (`string str` , `string option` , *`mixed encoding`*)
[lien librairie]

`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.

```
Options de conversions possibles :
"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 les espaces "zen−kaku" en "han−kaku" (U+3000 −> U+0020)
"S" :  Convertit les espaces "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
<?php
/* Convertit tous les "kana" en "zen−kaku" "kata−kana" */
$str = mb_convert_kana($str, "KVC");
/* Convertit "han−kaku" "kata−kana" en "zen−kaku" "kata−kana"
   et "zen−kaku" alpha−numeric en "han−kaku" */
$str = mb_convert_kana($str, "KVa");
?>
```

6.52.13 mb_convert_variables() : Convertit l'encodage de variables

`string` **mb_convert_variables** (`string to−encoding` , `mixed from−encoding` , `mixed vars`)
[lien librairie]

`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 fournit 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);
?>
```

6.52.14 mb_decode_mimeheader() : Décode une en-tête MIME

`string` **mb_decode_mimeheader** (`string str`)
[lien librairie]

`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` .

6.52.15 mb_decode_numericentity() : Décode les entités HTML en caractères

`string` **mb_decode_numericentity** (`string str` , `array convmap` , *`string encoding`*)
[lien librairie]

`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 entite numérique
?>
```

Voir aussi mb_encode_numericentity .

6.52.16 mb_detect_encoding() : Détecte un encodage

string **mb_detect_encoding** (string str , *mixed encoding-list*)
[lien librairie]

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  */
$ary[] = "ASCII";
$ary[] = "JIS";
$ary[] = "EUC-JP";
echo mb_detect_encoding($str, $ary);
?>
```

Voir aussi mb_detect_order .

6.52.17 mb_detect_order() : Lit/modifie l'ordre de détection des encodages

array **mb_detect_order** (*mixed encoding-list*)
[lien librairie]

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 .

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 */
$ary[] = "ASCII";
$ary[] = "JIS";
$ary[] = "EUC-JP";
mb_detect_order($ary);
/* 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`

6.52.18 mb_encode_mimeheader() : Encode une chaîne pour une en-tête MIME

`string` **mb_encode_mimeheader** (`string str` , *`string charset`* , *`string transfer-encoding`* , *`string linefeed`*)
[lien librairie]

`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 . "&#64;" . $doma . ">";
echo $addr;
?>
```

Voir aussi `mb_decode_mimeheader` .

6.52.19 mb_encode_numericentity() : Encode des entités HTML

`string` **mb_encode_numericentity** (`string str` , `array convmap` , *`string encoding`*)
[lien librairie]

`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 entite 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 .

6.52.20 mb_http_input() : Détecte le type d'encodage d'un caractère HTTP

string **mb_http_input** (*string type*)
[lien librairie]

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

6.52.21 mb_http_output() : Lit/modifie l'encodage d'affichage

`string` **mb_http_output** (*`string encoding`*)
[lien librairie]

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`

6.52.22 mb_internal_encoding() : Lit/modifie l'encodage interne

`string` **mb_internal_encoding** (*`string encoding`*)
[lien librairie]

`mb_internal_encoding` modifie l'encodage interne courant en le remplacant par `encoding` . Si ce paramètre est omis, l'encodage interne courant est retourné.

`encoding` sert lors des conversions des chaînes en provenance et en direction du web, ainsi que lors de la création de chaînes avec le module mbstring.

`encoding` : Nom d'encodage.

Valeur retournée : si `encoding` est fourni, `mb_internal_encoding` retourne TRUE en cas de succès, et FALSE sinon. Si `encoding` est omis, `mb_internal_encoding` retourne le nom de l'encodage courant.

Exemple avec `mb_internal_encoding`

```
<?php
/* Utilise l'encodage interne UTF-8 */
mb_internal_encoding("UTF-8");
/* Affiche l'encodage interne courant */
echo mb_internal_encoding();
?>
```

Voir aussi `mb_http_input` , `mb_http_output` et `mb_detect_order`

6.52.23 mb_language() : Lit/modifie le langage courant

`string` **mb_language** (*`string language`*)
[lien librairie]

`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 .

6.52.24 mb_output_handler() : Fonction de traitement des affichages web

string **mb_output_handler** (string contents , int status)
[lien librairie]

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 .

6.52.25 mb_parse_str() : Analyse les données HTTP GET/POST/COOKIE et assigne les variables globales

string **mb_parse_str** (string encoded_string , *array result*)
[lien librairie]

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 .

6.52.26 mb_preferred_mime_name() : Détecte l'encodage MIME

string **mb_preferred_mime_name** (string encoding)
[lien librairie]

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));
?>
```

6.52.27 mb_send_mail() : Envoie un mail encodé ISO-2022-JP (mail japonais)

boolean **mb_send_mail** (string to , string subject , string message , *string additional_headers* , *string additional_parameter*)
[lien librairie]

mb_send_mail envoie un courrier électronique. Les en-têtes et le corps du mesage 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).

mb_send_mail retourne TRUE en cas de succès, et FALSE en cas d'erreur.

Voir aussi mail .

6.52.28 mb_strcut() : Coupe une partie de chaîne

`string` **mb_strcut** (`string str` , `int start` , *`int length`* , *`string encoding`*)
[lien librairie]

`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` .

6.52.29 mb_strimwidth() : Tronque une chaîne

`string` **mb_strimwidth** (`string str` , `int start` , `int width` , `string trimmarker` , *`string encoding`*)
[lien librairie]

`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'offet 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` .

6.52.30 mb_strlen() : Retourne la taille d'une chaîne

`string` **mb_strlen** (`string str` , *`string encoding`*)
[lien librairie]

`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 .

6.52.31 mb_strpos() : Repère la première occurence d'un caractère dans une chaîne

string **mb_strpos** (string haystack , string needle , *int offset* , *string encoding*)
[lien librairie]

mb_strpos retourne la position numérique de la première occurence 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_strrpos accepte des chaînes comme argument needle , alors que strrpos 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

6.52.32 mb_strrpos() : Repère la dernière occurence d'un caractère dans une chaîne

string **mb_strrpos** (string haystack , string needle , *string encoding*)
[lien librairie]

mb_strrpos retourne la position numérique de la première occurence du caractère needle dans la chaîne haystack . Si needle est introuvable, mb_strrpos retourne FALSE .

mb_strrpos 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_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 .

6.52.33 mb_strwidth() : Retourne la largeur d'une chaîne

`int` **mb_strwidth** (`string str` , *`string encoding`*)
[lien librairie]

`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.

```
Largeur de caractères :
U+0000 − U+0019   0
U+0020 − U+1FFF   1
U+2000 − U+FF60   2
U+FF61 − U+FF9F   1
U+FFA0 −          2
```

`encoding` est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi `mb_strimwidth` et `mb_internal_encoding` .

6.52.34 mb_substitute_character() : Lit/modifie les caractères de substitution

`mixed` **mb_substitute_character** (*`mixed substrchar`*)
[lien librairie]

`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 héxadécimal.

Ce paramétrage affecte `mb_detect_encoding` et `mb_send_mail` .

`substchar` spécifie une valeur Unicode sous la forme d'un entier, ou bien une chaîne sous ces formes :

- "none" : pas d'affichage
- "long" : affiche la valeur hexadécimale (Par exemple : U+3000,JIS+7E7E)

SI `substchar` est fourni, `mb_substitute_character` retourne `TRUE` en cas de succès, et `FALSE` en cas d'erreur. Si `substchar` est omis, `mb_substitute_character` retourne une valeur Unicode, ou bien " `none` "/" `long` ".

Exemple avec `mb_substitute_character`

```php
<?php
/* Configure le caractère de substitution avec U+3013 (GETA MARK) */
mb_substitute_character(0x3013);
/* Configure le caractère de substitution avec un format héxadécimal */
mb_substitute_character("long");
/* Affiche la configuration courante */
echo mb_substitute_character();
```

```
?>
```

6.52.35 mb_substr() : Lit une sous−chaîne

`string` **mb_substr** (`string str` , `int start` , *`int length`* , *`string encoding`*)
[lien librairie]

`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` .

6.53 MCAL

6.53.1 Introduction

MCAL signifie Modular Calendar Access Library (librairie calendaire modulaire).

Libmcal est une librairie 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 librairie.

Les événements peuvent être lus, sélectionnés et enregistrés. Il y a aussi la possibilité d'ajouter des alarmes, et de placer des événéments récurents.

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 librairie originale et l'extension PHP ne sont plus supportées. La solution de remplacement est MCAL.

6.53.2 Pré−requis

Pour faire fonctionner cette librairie, vous devez compiler PHP avec l'option `−−with−mcal` . Il vous faudra alors avoir installé la librairie mcal. Téléchargez la dernière version à http://mcal.chek.com/ et compilez−la, puis installez−la.

6.53.3 Installation

Après avoir installé la librairie mcal, vous devez compiler PHP avec l'option `−−with−mcal` .

6.53.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.53.5 Types de ressources

Cette extension ne définit aucune ressource.

6.53.6 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 (nombre ˆ virgule flottante)
MCAL_MONDAY (nombre ˆ virgule flottante)
MCAL_TUESDAY (nombre ˆ virgule flottante)
MCAL_WEDNESDAY (nombre ˆ virgule flottante)
MCAL_THURSDAY (nombre ˆ virgule flottante)
MCAL_FRIDAY (nombre ˆ virgule flottante)
MCAL_SATURDAY (nombre ˆ virgule flottante)
MCAL_JANUARY (nombre ˆ virgule flottante)
MCAL_FEBRUARY (nombre ˆ virgule flottante)
MCAL_MARCH (nombre ˆ virgule flottante)
MCAL_APRIL (nombre ˆ virgule flottante)
MCAL_MAY (nombre ˆ virgule flottante)
MCAL_JUNE (nombre ˆ virgule flottante)
MCAL_JULY (nombre ˆ virgule flottante)
MCAL_AUGUST (nombre ˆ virgule flottante)
MCAL_SEPTEMBER (nombre ˆ virgule flottante)
MCAL_OCTOBER (nombre ˆ virgule flottante)
MCAL_NOVEMBER (nombre ˆ virgule flottante)
MCAL_DECEMBER (nombre ˆ virgule flottante)
MCAL_RECUR_NONE (nombre ˆ virgule flottante)
MCAL_RECUR_DAILY (nombre ˆ virgule flottante)
MCAL_RECUR_WEEKLY (nombre ˆ virgule flottante)
MCAL_RECUR_MONTHLY_MDAY (nombre ˆ virgule flottante)
MCAL_RECUR_MONTHLY_WDAY (nombre ˆ virgule flottante)
MCAL_RECUR_YEARLY (nombre ˆ virgule flottante)
MCAL_M_SUNDAY (nombre ˆ virgule flottante)
MCAL_M_MONDAY (nombre ˆ virgule flottante)
MCAL_M_TUESDAY (nombre ˆ virgule flottante)
MCAL_M_WEDNESDAY (nombre ˆ virgule flottante)
MCAL_M_THURSDAY (nombre ˆ virgule flottante)
MCAL_M_FRIDAY (nombre ˆ virgule flottante)
MCAL_M_SATURDAY (nombre ˆ virgule flottante)
MCAL_M_WEEKDAYS (nombre ˆ virgule flottante)
MCAL_M_WEEKEND (nombre ˆ virgule flottante)
MCAL_M_ALLDAYS (nombre ˆ virgule flottante)

- 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

- mcal_days_in_month
- mcal_delete_calendar
- mcal_delete_event
- mcal_event_add_attribute
- mcal_event_init
- mcal_event_set_alarm
- mcal_event_set_category
- mcal_event_set_class
- mcal_event_set_description
- mcal_event_set_end
- mcal_event_set_recur_daily
- mcal_event_set_recur_monthly_mday
- mcal_event_set_recur_monthly_wday
- mcal_event_set_recur_none
- mcal_event_set_recur_weekly
- mcal_event_set_recur_yearly
- mcal_event_set_start
- mcal_event_set_title
- mcal_expunge
- mcal_fetch_current_stream_event
- mcal_fetch_event
- mcal_is_leap_year
- mcal_list_alarms
- mcal_list_events
- mcal_next_recurrence
- mcal_open
- mcal_popen
- mcal_rename_calendar
- mcal_reopen
- mcal_snooze
- mcal_store_event
- mcal_time_valid
- mcal_week_of_year

6.53.8 mcal_close() : Ferme une connexion MCAL.

`int` **mcal_close** (`resource mcal_stream` , `int flags`)
[lien librairie]

`mcal_close` ferme la connexion `mcal_stream` .

6.53.9 mcal_create_calendar() : Crée un nouveau calendrier

`string` **mcal_create_calendar** (`resource mcal_stream` , `string calendar`)
[lien librairie]

`mcal_create_calendar` crée un nouveau calendrier nommé `calendar` .

6.53.10 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`)
[lien librairie]

`mcal_date_compare` compares les deux dates données, et retourne <0, 0, > 0 si a<b, a==b, a>b respectivement.

6.53.11 mcal_date_valid() : Valide une date.

`int` **mcal_date_valid** (`int year` , `int month` , `int day`)
[lien librairie]

`mcal_date_valid` retourne `TRUE` si la date (constituée par l'année `year` , le mois `month` et la date `day`) est valide, et `FALSE` sinon.

6.53.12 mcal_day_of_week() : Le jour de la semaine.

`int` **mcal_day_of_week** (`int year` , `int month` , `int day`)
[lien librairie]

`mcal_day_of_week` retourne le jour de la semaine, pour la date constituée par l'année `year` , le mois `month` et la date `day` .

6.53.13 mcal_day_of_year() : Le jour de l'année.

`int` **mcal_day_of_year** (`int year` , `int month` , `int day`)
[lien librairie]

`mcal_day_of_year` retourne le numéro de jour dans l'année pour la date constituée par l'année `year` , le mois `month` et la date `day` .

6.53.14 mcal_days_in_month() : Retourne le nombre de jour d'un mois.

`int` **mcal_days_in_month** (`int month` , `int leap_year`)
[lien librairie]

`mcal_days_in_month` retourne le nombre de jour du mois `month` , et prend en compte le fait que l'année est bissextile avec le paramètre `leap_year` .

6.53.15 mcal_delete_calendar() : Efface un calendrier

`string` **mcal_delete_calendar** (`resource mcal_stream` , `string calendar`)
[lien librairie]

`mcal_delete_calendar` efface le calendrier `calendar` .

6.53.16 mcal_delete_event() : Efface un événement dans un calendrier MCAL.

`int` **mcal_delete_event** (`resource mcal_stream` , *`int event_id`*)
[lien librairie]

`mcal_delete_event` efface l'événement d'identifiant `uid` .

Retourne `TRUE` .

6.53.17 mcal_event_add_attribute() : Ajoute un attribut et une valeur à la structure globale

`void` **mcal_event_add_attribute** (`resource mcal_stream` , `string attribute` , `string value`)
[lien librairie]

`mcal_event_add_attribute` ajoute l'attribut `attribute` à la structure globale, avec la valeur `value` .

6.53.18 mcal_event_init() : Initialise la structure globale d'un flot.

`int` **mcal_event_init** (`resource mcal_stream`)
[lien librairie]

`mcal_event_init` initialise la structure globale d'un flot. Cela remet tous les éléments de la structure à 0, ou à leur valeur par défaut.

`mcal_event_init` retourne `TRUE` .

6.53.19 mcal_event_set_alarm() : Fixe l'alarme de la structure globale.

`int` **mcal_event_set_alarm** (`resource mcal_stream` , `int alarm`)
[lien librairie]

`mcal_event_set_alarm` fixe l'alarme de la structure globale,à un nombre de minutes avant déclenchement.

`mcal_event_set_alarm` retourne `TRUE` .

6.53.20 mcal_event_set_category() : Fixe la catégorie de la structure globale.

`int` **mcal_event_set_category** (`resource mcal_stream` , `string category`)
[lien librairie]

`mcal_event_set_category` fixe la catégorie de la structure globale à la valeur de `category` .

mcal_event_set_category retourne TRUE .

6.53.21 mcal_event_set_class() : Fixe la classe de la structure globale.

int **mcal_event_set_class** (resource mcal_stream , int class)
[lien librairie]

mcal_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.

mcal_event_set_class retourne TRUE .

6.53.22 mcal_event_set_description() : Fixe la description de la structure globale.

int **mcal_event_set_description** (resource mcal_stream , string description)
[lien librairie]

mcal_event_set_description fixe la catégorie de la structure globale à la valeur de description .

mcal_event_set_description retourne TRUE .

6.53.23 mcal_event_set_end() : Fixe la date de fin de la structure globale.

int **mcal_event_set_end** (resource mcal_stream , int year , int month , *int day* , *int hour* , *int min* , *int sec*)
[lien librairie]

mcal_event_set_end fixe la date de fin de la structure globale.

mcal_event_set_end retourne TRUE .

6.53.24 mcal_event_set_recur_daily() : Fixe la récurrence quotidienne.

int **mcal_event_set_recur_daily** (resource mcal_stream , int year , int month , int day , int interval)
[lien librairie]

mcal_event_set_recur_daily fixe la récurrence quotidienne de la structure globale, jusqu'à la date passée en paramètre. (la date de début est celle de la structure).

6.53.25 mcal_event_set_recur_monthly_mday() : Fixe la récurrence.

int **mcal_event_set_recur_monthly_mday** (resource mcal_stream , int year , int month , int day , int interval)
[lien librairie]

mcal_event_set_recur_monthly_mday fixe la récurrence de la structure globale, jusqu'à la date passée en paramètre. (la date de début est celle de la structure).

6.53.26 mcal_event_set_recur_monthly_wday() : Fixe la récurrence mensuelle.

int **mcal_event_set_recur_monthly_wday** (resource mcal_stream , int year , int month , int day , int interval)
[lien librairie]

mcal_event_set_recur_monthly_wday fixe la récurrence mensuelle de la structure globale, jusqu'à la date passée en paramètre. (la date de début est celle de la structure).

6.53.27 mcal_event_set_recur_none() : Supprime la récurrence de la structure globale.

int **mcal_event_set_recur_none** (resource mcal_stream)
[lien librairie]

mcal_event_set_recur_none supprime la récurrence de la structure globale (event−>recur_type est mis à MCAL_RECUR_NONE).

6.53.28 mcal_event_set_recur_weekly() : Fixe la récurrence hebdomadaire.

int **mcal_event_set_recur_weekly** (resource mcal_stream , int year , int month , int day , int interval , int weekdays)
[lien librairie]

mcal_event_set_recur_weekly fixe la récurrence hebdomadaire de la structure globale, jusqu'à la date passée en paramètre. (la date de début est celle de la structure).

6.53.29 mcal_event_set_recur_yearly() : Fixe la récurrence annuelle.

int **mcal_event_set_recur_yearly** (resource mcal_stream , int year , int month , int day , int interval)
[lien librairie]

mcal_event_set_recur_yearly fixe la récurrence annuelle de la structure globale, jusqu'à la date passée en paramètre. (la date de début est celle de la structure).

6.53.30 mcal_event_set_start() : Fixe les dates de début et de fin de la structure globale.

int **mcal_event_set_start** (resource mcal_stream , int year , int month , *int day* , *int hour* , *int min* , *int sec*)
[lien librairie]

mcal_event_set_start fixe la date de début de la structure globale.

mcal_event_set_start retourne TRUE .

6.53.31 mcal_event_set_title() : Fixe le titre de la structure globale.

int **mcal_event_set_title** (resource mcal_stream , string title)
[lien librairie]

mcal_event_set_title fixe le titre de la structure globale à la valeur de title .

mcal_event_set_title retourne TRUE .

6.53.32 mcal_expunge() : Supprime tous les événements marqués pour l'effacement

int **mcal_expunge** (resource mcal_stream)
[lien librairie]

mcal_expunge supprime tous les événements marqués pour l'effacement.

6.53.33 mcal_fetch_current_stream_event() : Retourne un objet contenant la structure de date pour le flot courant.

object **mcal_fetch_current_stream_event** (resource mcal_stream)
[lien librairie]

mcal_fetch_current_stream_event retourne la structure de la date du flot 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_endate − date de fin de récurrence
- int recur_data − données de récurrence

Tous les objets de date et heure sont construits comme suit :

- int year − année
- int month − mois
- int mday − jour du mois
- int hour − heure
- int min − minutes

- int sec − secondes
- int alarm − nombre de minutes avant de déclencher l'alarme

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

6.53.34 mcal_fetch_event() : Recherche un événement dans le calendrier.

`object` **mcal_fetch_event** (`resource mcal_stream` , `int event_id` , *`int options`*)
[lien librairie]

`mcal_fetch_event` recherche un évenement 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_endate − date de fin de récurrence
- int recur_data − données de récurrence

Tous les objets de date et heure sont construits comme suit :

- int year − année
- int month − mois
- int mday − jour du mois
- int hour − heure
- int min − minutes
- int sec − secondes
- int alarm − nombre de minutes avant de déclencher l'alarme

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

6.53.35 mcal_is_leap_year() : Vérifie que l'année est bissextile.

`int` **mcal_is_leap_year** (`int year`)
[lien librairie]

`mcal_is_leap_year` retoune 1 si l'année `year` est bissextile, et 0 sinon.

6.53.36 mcal_list_alarms() : Retourne une liste d'événements qui ont une alarme prévue à une date.

`array` **mcal_list_alarms** (`resource mcal_stream` , *`int begin_year`* , *`int begin_month`* , *`int begin_day`* , *`int end_year`* , *`int end_month`* , *`int end_day`*)
[lien librairie]

`mcal_list_events` retourne un tableau d'identifiants, qui ont une alarme de prévue à la date `alarm_date` . Si seul le flot MCAL est donné, la date de début et de fin de la structure globale sera utilisée.

`mcal_list_events` prend une date, et retourne un tableau d'identifiants.

6.53.37 mcal_list_events() : Retourne une liste d'événement entre deux dates.

`array` **mcal_list_events** (`resource mcal_stream` , `object begin_date` , *`object end_date`*)
[lien librairie]

`mcal_list_events` retourne un tableau d'identifiants d'événéments, compris entre deux dates.

`mcal_list_events` prend une date de début et une date de fin. Un tableau d'identifiants est retourné.

6.53.38 mcal_next_recurrence() : Retourne la prochaine occurrence d'un événement.

`int` **mcal_next_recurrence** (`resource mcal_stream` , `int weekstart` , `array next`)
[lien librairie]

`mcal_next_recurrence` retourne un objet contenant la prochaine date de l'événement, ou la date de l'événement suivant la date. `mcal_next_recurrence` retourne un objet date vide si l'événement n'a pas de réoccurrence, ou si quelquechose est invalide. Utilisez `weekstart` pour déterminer le premier jour.

6.53.39 mcal_open() : Ouvre une connexion MCAL.

`resource` **mcal_open** (`string calendar` , `string username` , `string password` , *`int options`*)
[lien librairie]

`mcal_open` retourne un flot MCAL en cas de succès, et FALSE en cas d'erreur.

`mcal_open` ouvre une connexion MCAL au serveur `calendar` . Si `options` est spécifié, passe aussi `options` à la boîte aux lettres (???). La structure interne du flot MCAL est initialisée à la connexion.

6.53.40 mcal_popen() : Ouvre une connexion persistante MCAL.

`resource` **mcal_popen** (`string calendar` , `string username` , `string password` , *`int options`*)
[lien librairie]

`mcal_popen` retourne un flot MCAL en cas de succès, et FALSE sinon.

`mcal_popen` ouvre une connexion MCAL au serveur de calendrier `calendar` . Si les options `options` sont spécifiées, elles sont aussi passé à cette boîte au lettre. La structure interne du flot est aussi initialisée.

6.53.41 mcal_rename_calendar() : Renomme un calendrier

`string` **mcal_rename_calendar** (`resource mcal_stream` , `string old_name` , `string new_name`)
[lien librairie]

`mcal_rename_calendar` renomme le calendrier `old_name` en `new_name` .

6.53.42 mcal_reopen() : Réouvre une connexion MCAL

`resource` **mcal_reopen** (`string calendar` , *`int options`*)
[lien librairie]

`mcal_reopen` réouvre une connexion MCAL.

`mcal_reopen` réouvre une connexion MCAL avec le serveur `calendar` . Si les options `options` sont spécifiées, elles sont aussi passé à cette boîte aux lettres.

6.53.43 mcal_snooze() : Eteint l'alarme d'un événement.

`int` **mcal_snooze** (`int id`)
[lien librairie]

`mcal_snooze` éteint l'alarme de l'événement identifié par l'UID `uid` .

mcal_snooze retourne TRUE .

6.53.44 mcal_store_event() : Modifie un événement dans un calendrier MCAL.

int **mcal_store_event** (resource mcal_stream)
[lien librairie]

mcal_store_event enregistre l'événement global dans le calendrier MCAL mcal_stream .

mcal_store_event retourne l'identifiant de l'événement modifié en cas de succès, et FALSE en cas d'erreur.

6.53.45 mcal_time_valid() : Valide une heure.

int **mcal_time_valid** (int hour , int minutes , int seconds)
[lien librairie]

mcal_time_valid retourne TRUE si l'heure (constituée par l'heure hour , les minutes minutes et les secondes seconds) est une heure valide, et FALSE sinon.

6.53.46 mcal_week_of_year() : Retourne le numéro de semaine dans l'année pour une date donnée

int **mcal_week_of_year** (int day , int month , int year)
[lien librairie]

6.54 Chiffrage mcrypt

6.54.1 Introduction

Ces fonctions permettent d'accéder à la librairie mcrypt, qui dispose d'une grande variété d'algorithmes de chiffrage, 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".

6.54.2 Pré−requis

Ces fonctions utilisent mcrypt .

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 (chiffrage), PANAMA, RC4 et WAKE. Avec `libmcrypt 2.4.x` un autre mode de chiffrement est disponible : nOFB.

6.54.3 Installation

Pour l'utiliser, téléchargez la librairie `libmcrypt−x.x.tar.gz` par ici et suivez les instructions d'installations incluses. Vous aurez aussi besoin de compiler PHP avec le paramètre `−−with−mcrypt` pour activer cette extension.

6.54.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Changeable
mcrypt.algorithms_dir	NULL	PHP_INI_ALL
mcrypt.modes_dir	NULL	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.54.5 Types de ressources

Cette extension ne définit aucune ressource.

6.54.6 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 chiffrage (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)

6.54.7 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 TWOFISHxxx 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 CFB 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) pour plus de détails sur le sujet).

6.54.8 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
<?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

6.54.10 mcrypt_cfb() : Chiffre/déchiffre des données en mode CFB

`string` **mcrypt_cfb** (`int cipher` , `string key` , `string data` , `int mode` , `string iv`)
[lien librairie]
`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` chiffre ou déchiffre (suivant le `mode` sélectionné) les données `data` avec le chiffrement `cipher` et la clé `key` en mode CFB et retourne la chaîne résultant.

`Cipher` est une des constantes MCRYPT_ciphername

`Key` est la clé fournie à l'algorithme. Elle doit être tenue secrète.

`Data` sont les données à traiter.

`Mode` vaut MCRYPT_ENCRYPT ou MCRYPT_DECRYPT.

`IV` est le vecteur d'initialisation (optionnel).

Voir aussi: `mcrypt_cbc` , `mcrypt_ecb` , et `mcrypt_ofb` .

6.54.11 mcrypt_create_iv() : Crée un vecteur d'initialisation à partir d'une source aléatoire.

`string` **mcrypt_create_iv** (`int size` , `int source`)
[lien librairie]

`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
$cipher = MCRYPT_TripleDES;
$block_size = mcrypt_get_block_size($cipher);
```

```
$iv = mcrypt_create_iv($block_size, MCRYPT_DEV_RANDOM);
?>
```

6.54.12 mcrypt_decrypt() : Déchiffre un texte

`string` **mcrypt_decrypt** (`string cipher` , `string key` , `string data` , `string mode` , *`string iv`*)
[lien librairie]

`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` '.

6.54.13 mcrypt_ecb() : Chiffre/déchiffre des données en mode ECB

`string` **mcrypt_ecb** (`int cipher` , `string key` , `string data` , `int mode`)
[lien librairie]
`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` chiffre ou déchiffre (suivant le `mode` sélectionné) les données `data` avec le chiffrement `cipher` et la clé `key` en mode CFB et retourne la chaîne résultant.

`Cipher` est une des constantes MCRYPT_ciphername

`Key` est la clé fournie à l'algorithme. Elle doit être tenue secrète.

`Data` sont les données à traiter.

`Mode` vaut MCRYPT_ENCRYPT ou MCRYPT_DECRYPT.

`IV` est le vecteur d'initialisation (optionnel).

Voir aussi: `mcrypt_cbc` , `mcrypt_cfb` , et `mcrypt_ofb` .

6.54.14 mcrypt_enc_get_algorithms_name() : Retourne le nom de l'algorithme

`string` **mcrypt_enc_get_algorithms_name** (`resource td`)
[lien librairie]

`mcrypt_enc_get_algorithms_name` retourne le nom de l'algorithme désigné par `td` .

6.54.15 mcrypt_enc_get_block_size() : Retourne la taille de blocs d'un algorithme

`int` **mcrypt_enc_get_block_size** (`resource td`)
[lien librairie]

`mcrypt_enc_get_block_size` retourne la taille de blocs d'un algorithme, en octets.

6.54.16 mcrypt_enc_get_iv_size() : Retourne la taille du VI d'un algorithme

`int` **mcrypt_enc_get_iv_size** (`resource td`)
[lien librairie]

`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.

6.54.17 mcrypt_enc_get_key_size() : Retourne la taille maximale de la clé pour un mode

`int` **mcrypt_enc_get_key_size** (`resource td`)
[lien librairie]

`mcrypt_enc_get_key_size` retourne la taille maximale de clé acceptée par le mode désigné par `td` , en octets.

6.54.18 mcrypt_enc_get_modes_name() : Retourne le nom du mode

`string` **mcrypt_enc_get_modes_name** (`resource td`)
[lien librairie]

`mcrypt_enc_get_modes_name` retourne le nom du mode désigné par `td` .

6.54.19 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`)

[lien librairie]

`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.

6.54.20 mcrypt_enc_is_block_algorithm_mode() : Teste le chiffrage par blocs d'un mode

`int` **mcrypt_enc_is_block_algorithm_mode** `( resource td )`
[lien librairie]

`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).

6.54.21 mcrypt_enc_is_block_algorithm() : Teste le chiffrage par blocs d'un algorithme

`int` **mcrypt_enc_is_block_algorithm** `( resource td )`
[lien librairie]

`mcrypt_enc_is_block_algorithm` retourne 1 si l'algorithme utilisé est un algorithme par blocs, et 0 si c'est un algorithme par flot.

6.54.22 mcrypt_enc_is_block_mode() : Teste si le mode retourne les données par blocs

`int` **mcrypt_enc_is_block_mode** `( resource td )`
[lien librairie]

`mcrypt_enc_is_block_mode` retourne 1 si le mode retourne des blocs d'octets, ou bien 0 s'il retourne des octets (par flot). (i.e. 1 pour cbc et ecb, et 0 pour cfb et stream).

6.54.23 mcrypt_enc_self_test() : Teste un module ouvert

`int` **mcrypt_enc_self_test** `( resource td )`
[lien librairie]

`mcrypt_enc_self_test` effectue un test du module ouvert et désigné par `td` . Si le test est concluant, elle retourne 0, sinon, 1.

6.54.24 mcrypt_encrypt() : Chiffre un texte

`string` **mcrypt_encrypt** `( string cipher , string key , string data , string mode , string iv )`
[lien librairie]

`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";
$crypttext = mcrypt_encrypt (MCRYPT_RIJNDAEL_256, $key, $text, MCRYPT_MODE_ECB, $iv);
echo strlen ($crypttext)."\n";
?>
```

L'exemple ci−dessus affichera :
42
64

6.54.25 mcrypt_generic_deinit() : Prépare le module pour le déchargement

`bool` **mcrypt_generic_deinit** (`resource td`)
[lien librairie]

`mcrypt_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 `mcrypt_module_close` vous même (mais PHP le fera pour vous a la fin du script). `mcrypt_generic_deinit` retourne FALSE en cas d'erreur, et TRUE sinon.

Pour un exemple, voir `mcrypt_module_open` et l'entrée sur `mcrypt_generic_init` .

6.54.26 mcrypt_generic_end() : Termine un chiffrage

`boolean` **mcrypt_generic_end** (`resource td`)
[lien librairie]

mcrypt_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.

6.54.27 mcrypt_generic_init() : Initialise tous les buffers nécessaires

int **mcrypt_generic_init** (resource td , string key , string iv)
[lien librairie]

La taille maximale de la clé doit être cette retournée par mcrypt_enc_get_key_size et toutes les valeurs inférieures seront aussi valides. Le vecteur d'initialisation (VI) doit avoir la taille d'un bloc, mais vous devez lire sa taille en appelant mcrypt_enc_get_iv_size . IV est ignoré en mode ECB. IV DOIT exister en modes CFB, CBC, STREAM, nOFB et OFB. Il doit être aléatoire et unique (mais pas secret). Le même VI doit être utilisé pour le chiffrage et le déchiffrage. 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 mcrypt_generic_init avant chaque appel à mcrypt_generic ou mdecrypt_generic .

6.54.28 mcrypt_generic() : Chiffre

string **mcrypt_generic** (resource td , string data)
[lien librairie]

mcrypt_generic chiffre des données. Les données sont complétées par des " \0 " pour obtenir une taille de n fois la taille d'un bloc. Elle retourne les données encryptées. Notez que la longueur de la chaîne retournée peut être plus longue que celle passée en argument, à cause du complément.

6.54.29 mcrypt_get_block_size() : Retourne la taille de blocs d'un chiffrement.

int **mcrypt_get_block_size** (int cipher)
[lien librairie]
int **mcrypt_get_block_size** (string cipher , string module)

mcrypt_get_block_size sert à lire la taille de blocs du chiffrement cipher .

mcrypt_get_block_size prend comme argument le chiffrement cipher et retourne une taille en octets.

Voir aussi : mcrypt_get_key_size .

6.54.30 mcrypt_get_cipher_name() : Lit le nom du chiffrement utilisé.

string **mcrypt_get_cipher_name** (int cipher)
[lien librairie]
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 donner :
TripleDES

6.54.31 mcrypt_get_iv_size() : Retourne la taille du VI utilisé par un couple chiffrement/mode

int **mcrypt_get_iv_size** (string cipher , string mode)
[lien librairie]
int **mcrypt_get_iv_size** (resource td)

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_modename qui peut valoir : "ecb", "cbc", "cfb", "ofb", "nofb" ou "stream".

6.54.32 mcrypt_get_key_size() : Retourne la taille de la clé d'un chiffrement.

int **mcrypt_get_key_size** (int cipher)
[lien librairie]
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.

Voir aussi: mcrypt_get_block_size .

6.54.33 mcrypt_list_algorithms() : Liste tous les algorithmes de chiffrement supportés

`array` **mcrypt_list_algorithms** (*`string  lib_dir`*)
[lien librairie]

`mcrypt_list_algorithms` sert à lister tous les algorithmes de chiffrement de `lib_dir` . `mcrypt_list_algorithms` prend un argument optionnel, qui spécifie le dossier qui contient tous les algorithmes. S'il est omis, la valeur de mcrypt.algorithms_dir dans `php.ini` est utilisée.

Exemple avec `mcrypt_list_algorithms`

```
<?php
$algorithms = mcrypt_list_algorithms ("/usr/local/lib/libmcrypt");
foreach ($algorithms as $cipher) {
    echo $cipher."/n";
}
?>
```

L'exemple ci−dessus va afficher tous les algorithmes supportés dans le dossier " `/usr/local/lib/libmcrypt` ".

6.54.34 mcrypt_list_modes() : Liste tous les modes de chiffrement supportés

`array` **mcrypt_list_modes** (*`string  lib_dir`*)
[lien librairie]

`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/libmcrypt` ".

6.54.35 mcrypt_module_close() : Décharge le module de chiffrement

`bool` **mcrypt_module_close** (`resource td`)
[lien librairie]

`mcrypt_module_close` décharge le module `td` .

Voir mcrypt_module_open .

6.54.36 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*)
[lien librairie]

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.

6.54.37 mcrypt_module_get_algo_key_size() : Retourne la taille maximale de clé

int **mcrypt_module_get_algo_key_size** (string algorithm , *string lib_dir*)
[lien librairie]

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.

6.54.38 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*)
[lien librairie]

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 utiliés pour ouvrir les modules de chiffrement.

6.54.39 mcrypt_module_is_block_algorithm_mode() : Indique si un mode fonctionne par blocs

boolean **mcrypt_module_is_block_algorithm_mode** (string mode , *string lib_dir*)
[lien librairie]

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.

6.54.40 mcrypt_module_is_block_algorithm() : Indique si un algorithme fonctionne par blocs

`boolean` **mcrypt_module_is_block_algorithm** (`string algorithm` , *`string lib_dir`*)
[lien librairie]

`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.

6.54.41 mcrypt_module_is_block_mode() : Indique si un mode travaille par blocs

`boolean` **mcrypt_module_is_block_mode** (`string mode` , *`string lib_dir`*)
[lien librairie]

`mcrypt_module_is_block_mode` retourne `TRUE` si ce mode fournit des blocs d'octets, ou bien un flot 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.

6.54.42 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`)
[lien librairie]

`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 utilisé 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.

6.54.43 mcrypt_module_self_test() : Teste un mode

`boolean` **mcrypt_module_self_test** (`string algorithm` , *`string lib_dir`*)
[lien librairie]

`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.

6.54.44 mcrypt_ofb() : Chiffre/déchiffre des données en mode OFB

`string` **mcrypt_ofb** (`int cipher` , `string key` , `string data` , `int mode` , `string iv`)
[lien librairie]
`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` chiffre ou déchiffre (suivant le `mode` sélectionné) les données `data` avec le chiffrement `cipher` et la clé `key` en mode OFB et retourne la chaîne résultant.

`Cipher` est une des constantes MCRYPT_ciphername

`Key` est la clé fournie à l'algorithme. Elle doit être tenue secrète.

`Data` sont les données à traiter.

`Mode` vaut MCRYPT_ENCRYPT ou MCRYPT_DECRYPT.

`IV` est le vecteur d'initialisation (optionnel).

Voir aussi: `mcrypt_cbc` , `mcrypt_cfb` , et `mcrypt_ecb` .

6.54.45 mdecrypt_generic() : Déchiffre

`string` **mdecrypt_generic** (`resource td` , `string data`)
[lien librairie]

`mdecrypt_generic` déchiffre les données `data` . Notez que la longueur de la chaîne décryptée peut être plus longue que la chaîne originale, car elle peut avoir été complétée par des " `\0` ".

Exemple avec `mdecrypt_generic`

```
<?php
$iv_size = mcrypt_enc_get_iv_size ($td));
$iv = @mcrypt_create_iv ($iv_size, MCRYPT_RAND);
if (@mcrypt_generic_init ($td, $key, $iv) != -1)
{
    $c_t = mcrypt_generic ($td, $plain_text);
    @mcrypt_generic_init ($td, $key, $iv);
    $p_t = mdecrypt_generic ($td, $c_t);
```

```
}
if (strncmp ($p_t, $plain_text, strlen($plain_text)) == 0)
    echo "ok";
else
    echo "erreur";
?>
```

L'exemple ci−dessus montre comment vérifier que les données avant chiffrage sont bien les mêmes que celles après chiffrage/déchiffrage.

6.55 MCVE (Paiement)

6.55.1 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, 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.

6.55.2 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 www.mcve.com .

6.55.3 Voir aussi

Plus de documentation sur l'API PHP de MCVE est disponible à http://www.mcve.com/docs/phpapi.pdf . La documentation éditée par Main Street est complète et est la source primaire d'information.

- Introduction
- Installation
- Voir aussi
- mcve_adduser
- mcve_adduserarg
- mcve_bt
- mcve_checkstatus
- mcve_chkpwd
- mcve_chngpwd
- mcve_completeauthorizations
- mcve_connect
- mcve_connectionerror

- mcve_deleteresponse
- 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_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

6.55.5 mcve_adduserarg() : Ajoute une valeur dans une structure de configuration d'utilisateur

`int` **mcve_adduserarg** (`resource usersetup` , `int argtype` , `string argval`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.6 mcve_bt() : Lit le total des paiements non finalisés en mode batch

`int` **mcve_bt** (`resource conn` , `string username` , `string password`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.7 mcve_checkstatus() : Vérifie si une transaction a été complétée

`int` **mcve_checkstatus** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.8 mcve_chkpwd() : Verifie le mot de passe

`int` **mcve_chkpwd** (`resource conn` , `string username` , `string password`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.9 mcve_chngpwd() : Modifie le mot de passe administrateur

`int` **mcve_chngpwd** (`resource conn` , `string admin_password` , `string new_password`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.10 mcve_completeauthorizations() : Retourne le nombre de transactions dans la queue d'autorisation, ainsi que la liste des identifiants

`int` **mcve_completeauthorizations** (`resource conn` , `int`)
`[ lien librairie ]`

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.11 mcve_connect() : Etablit la connexion avec le système MCVE

`int` **mcve_connect** (`resource conn`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.12 mcve_connectionerror() : Lit un message représentant la raison de l'échec de la connexion

`string` **mcve_connectionerror** (`resource conn`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.13 mcve_deleteresponse() : Efface une transaction spécifiée dans une structure MCVE_CONN

`bool` **mcve_deleteresponse** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.14 mcve_deletetrans() : Efface une transaction spécifiée dans une structure MCVE_CONN

`bool` **mcve_deletetrans** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.15 mcve_deleteusersetup() : Supprime une structure usersetup

`void` **mcve_deleteusersetup** (`resource usersetup`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.16 mcve_deluser() : Efface un compte d'utilisateur MCVE

`int` **mcve_deluser** (`resource conn` , `string admin_password` , `string username`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.17 mcve_destroyconn() : Termine une connexion MCVE_CONN

`void` **mcve_destroyconn** (`resource conn`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.18 mcve_destroyengine() : Libère la mémoire associée à la connectivité IP/SSL

`void` **mcve_destroyengine** (`void`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.19 mcve_disableuser() : Inactive un utilisateur MCVE actif

`int` **mcve_disableuser** (`resource conn` , `string admin_password` , `string username`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.20 mcve_edituser() : Modifie la structure usersetup d'un utilisateur

`int` **mcve_edituser** (`resource conn` , `string admin_password` , `int usersetup`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.21 mcve_enableuser() : Active un utilisateur MCVE inactif

`int` **mcve_enableuser** (`resource conn` , `string admin_password` , `string username`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.22 mcve_force() : Envoie une commande FORCE à MCVE. (généralement une autorisation par téléphone)

`int` **mcve_force** (`resiurce conn` , `string username` , `string password` , `string trackdata` , `string account` , `string expdate` , `float amount` , `string authcode` , `string comments` , `string clerkid` , `string stationid` , `int ptrannum`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.23 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.24 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.25 mcve_getcommadelimited() : Lit une ligne brute, au format CSV retournée par MCVE

`string` **mcve_getcommadelimited** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.26 mcve_getheader() : Lit le nom d'une colonne dans une réponse au format CSV

`string` **mcve_getheader** (`resource conn` , `int identifier` , `int column_num`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.27 mcve_getuserarg() : Lit une valeur dans une structure usersetup

`string` **mcve_getuserarg** (`resource usersetup` , `int argtype`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.28 mcve_getuserparam() : Lit un paramètre de réponse utilisateur

`string` **mcve_getuserparam** (`resource conn` , `long identifier` , `int key`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.29 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.30 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.31 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.32 mcve_initconn() : Crée et initialise une structure MCVE_CONN

`resource` **mcve_initconn** (`void`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.33 mcve_initengine() : Prépare le client pour la connexion IP/SSL

`int` **mcve_initengine** (`string location`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.34 mcve_initusersetup() : Initialise une structure pour stocker des données utilisateurs

`resource` **mcve_initusersetup** (`void`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.35 mcve_iscommadelimited() : Vérifie qu'une réponse est au format MVCE

`int` **mcve_iscommadelimited** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.36 mcve_liststats() : Liste les statistiques de tous les utilisateurs MCVE

`int` **mcve_liststats** (`resource conn` , `string admin_password`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.37 mcve_listusers() : Liste de tous les utilisateurs du système MCVE

`int` **mcve_listusers** (`resource conn` , `string admin_password`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.38 mcve_maxconntimeout() : La durée maximum de tentative de connexion à MCVE

`bool` **mcve_maxconntimeout** (`resource conn` , `int secs`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.39 mcve_monitor() : Communique avec MCVE (envoie et reçoit des données) sans blocage

`int` **mcve_monitor** (`resource conn`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.40 mcve_numcolumns() : Nombre de colonnes retournées dans une ligne au format CSV

`int` **mcve_numcolumns** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.41 mcve_numrows() : Nombre de lignes retournées dans une réponse au format CSV

`int` **mcve_numrows** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.42 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.43 mcve_parsecommadelimited() : Analyse une au format CSV pour que mcve_getcell et consorts fonctionnent

`int` **mcve_parsecommadelimited** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.44 mcve_ping() : Envoie une requête de ping à MCVE

`int` **mcve_ping** (`resource conn`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.45 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.46 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.47 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.48 mcve_responseparam() : Lit un paramètre de réponse personnalisé

`string` **mcve_responseparam** (`resource conn` , `long identifier` , `string key`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.49 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.50 mcve_returncode() : Lit le code retourné exact de la transaction

`int` **mcve_returncode** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.51 mcve_returnstatus() : Vérifie si la transaction a réussi

`int` **mcve_returnstatus** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.52 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.53 mcve_setblocking() : Configure la connexion en mode bloquant/non bloquant

`int` **mcve_setblocking** (`resource conn` , `int tf`)

[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.54 mcve_setdropfile() : Configure la connexion en mode Drop−File

`int` **mcve_setdropfile** (`resource conn` , `string directory`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.55 mcve_setip() : Configure la connexion en mode IP

`int` **mcve_setip** (`resource conn` , `string host` , `int port`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.56 mcve_setssl() : Configure la connexion en mode SSL

`int` **mcve_setssl** (`resource conn` , `string host` , `int port`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.57 mcve_settimeout() : Configure le temps maximum d'une transaction (par transaction)

`int` **mcve_settimeout** (`resource conn` , `int seconds`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.58 mcve_settle() : Emet une résolution pour faire un dépôt batch

`int` **mcve_settle** (`resource conn` , `string username` , `string password` , `string batch`)
[lien librairie]

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.59 mcve_text_avs() : Lit une représentation textuelle de return_avs

`string` **mcve_text_avs** (`string code`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.60 mcve_text_code() : Lit une représentation textuelle de eturn_code

`string` **mcve_text_code** (`string code`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.61 mcve_text_cv() : Lit une représentation textuelle de return_cv

`string` **mcve_text_cv** (`int code`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.62 mcve_transactionauth() : Lit le numéro d'autorisation de la transaction (alpha−numérique)

`string` **mcve_transactionauth** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.63 mcve_transactionavs() : Lit l'adresse de vérification (Address Verification) dans le statut retourné

`int` **mcve_transactionavs** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.64 mcve_transactionbatch() : Lit le numéro de batch associé à la transaction

`int` **mcve_transactionbatch** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.65 mcve_transactioncv() : Lit le statut de CVC2/CVV2/CID

`int` **mcve_transactioncv** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.66 mcve_transactionid() : Lit l'identifiant système unique d'une transaction

`int` **mcve_transactionid** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.67 mcve_transactionitem() : Lit le numéro d'ITEM d'une transaction dans un batch

`int` **mcve_transactionitem** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.68 mcve_transactionssent() : Vérifie si le buffer de sortie est vide ou pas

`int` **mcve_transactionssent** (`resource conn`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.69 mcve_transactiontext() : Lit le message retourné par MCVE ou par l'institution de traitement

`string` **mcve_transactiontext** (`resource conn` , `int identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.70 mcve_transinqueue() : Nombre de transactions dans la queue client

`int` **mcve_transinqueue** (`resource conn`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.71 mcve_transnew() : Démarre une nouvelle transaction

`int` **mcve_transnew** (`resource conn`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.72 mcve_transparam() : Ajoute un paramètre à une transaction

`int` **mcve_transparam** (`resource conn` , `long identifier` , `int key`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.73 mcve_transsend() : Finalise et envoie une transaction

`int` **mcve_transsend** (`resource conn` , `long identifier`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.74 mcve_ub() : Lit une liste des paiements en attente en mode batch

`int` **mcve_ub** (`resource conn` , `string username` , `string password`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.75 mcve_uwait() : Attend x microsecondes

`int` **mcve_uwait** (`long microsecs`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.76 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.77 mcve_verifysslcert() : Configure MCVE pour qu'il vérifie ou pas le certificat SSL

`bool` **mcve_verifysslcert** (`resource conn` , `int tf`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.55.78 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`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.56 Hash

6.56.1 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.

6.56.2 Pré−requis

Pour l'utiliser, téléchargez les distributions de mhash depuis le site web ici et suivez les instructions d'installation incluses.

6.56.3 Installation

Vous aurez besoin de recompiler PHP avec l'option `−−with−mhash[=DIR]` pour activer cette extension.

6.56.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.56.5 Types de ressources

Cette extension ne définit aucune ressource.

6.56.6 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

6.56.7 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";
?>
```


- 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

6.56.9 mhash_get_block_size() : Retourne la taille de bloc du hash.

`int` **mhash_get_block_size** (`int hash`)
[lien librairie]

`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.

6.56.10 mhash_get_hash_name() : Retourne le nom du hash.

`string` **mhash_get_hash_name** (`int hash`)
[lien librairie]

`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 `mhash_get_hash_name`

```
<?php
  $hash = MHASH_MD5;
  print mhash_get_hash_name($hash);
?>
```

L'exemple ci−dessus va afficher :
`MD5`

6.56.11 mhash_keygen_s2k() : Génère une clé

`string` **mhash_keygen_s2k** (`int hash` , `string password` , `string salt` , `int bytes`)
[lien librairie]

`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 génèrez, 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.

6.56.12 mhash() : Calcule un hash.

`string` **mhash** (`int hash` , `string data`)
[lien librairie]

`mhash` applique la fonction de hash `hash` aux données `data` et retourne le résultat.

6.57 Mimetype

6.57.1 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 réalise un bon travail.

Cette extension est dérivée du moduel 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.

6.57.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.57.3 Installation

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` .

6.57.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
mime_magic.magicfile	"/usr/share/misc/magic.mime"	PHP_INI_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.57.5 Types de ressources

Cette extension ne définit aucune ressource.

6.57.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- mime_content_type

6.58 Microsoft SQL Server

6.58.1 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.

6.58.2 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.

6.58.3 Installation

L'extension MSSQL est activée par la ligne `extension=php_mssql.dll` dans le fichier `php.ini` .

6.58.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
mssql.allow_persistent	"1"	PHP_INI_SYSTEM
mssql.max_persistent	"−1"	PHP_INI_SYSTEM
mssql.max_links	"−1"	PHP_INI_SYSTEM
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
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
mssql.secure_connection	"0"	PHP_INI_SYSTEM
mssql.max_procs	"25"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.58.5 Types de ressources

6.58.6 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 (nombre ˆ virgule flottante)
MSSQL_NUM (nombre ˆ virgule flottante)
MSSQL_BOTH (nombre ˆ virgule flottante)
SQLTEXT (nombre ˆ virgule flottante)
SQLVARCHAR (nombre ˆ virgule flottante)
SQLCHAR (nombre ˆ virgule flottante)
SQLINT1 (nombre ˆ virgule flottante)
SQLINT2 (nombre ˆ virgule flottante)
SQLINT4 (nombre ˆ virgule flottante)
SQLBIT (nombre ˆ virgule flottante)
SQLFLT8 (nombre ˆ virgule flottante)

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- mssql_close
- mssql_connect
- 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_free_result
- mssql_get_last_message
- mssql_min_error_severity
- mssql_min_message_severity
- fbsql_next_result
- mssql_num_fields
- mssql_num_rows
- mssql_pconnect
- mssql_query
- mssql_result
- mssql_select_db

6.58.8 mssql_connect() : Ouvre une connexion à un serveur MS SQL server.

resource **mssql_connect** (*string servername* , *string username* , *string password*)
[lien librairie]

mssql_connect r etourne 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 .

6.58.9 mssql_data_seek() : Déplace le pointeur interne de ligne.

int **mssql_data_seek** (resource result_identifier , int row_number)
[lien librairie]

mssql_data_seek retourne TRUE en cas de succès, FALSE en cas d'échec.

mssql_data_seek déplace le pointeur interne de ligne, dans le résultat result_identifier , jusqu'à la ligne row_number . Le prochain appel à mssql_fetch_row retournera cette ligne.

Voir aussi mssql_data_seek .

6.58.10 mssql_fetch_array() : Lit une ligne dans un tableau.

int **mssql_fetch_array** (resource result)
[lien librairie]

mssql_fetch_array retourne un tableau qui contient les valeurs de la ligne lues, 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 apréciable.

Pour plus de détails, voyez mssql_fetch_row .

6.58.11 mssql_fetch_field() : Lit les inforamtions sur le champs.

object **mssql_fetch_field** (resource result , *int field_offset*)
[lien librairie]

mssql_fetch_field retourne un objet contenant les informations sur un champs.

mssql_fetch_field sert à lire des informations spécifiques à un champs, dans un résultat de requête. Si l'offset du champs field_offset n'est pas précisé, le prochain champs 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 .

6.58.12 mssql_fetch_object() : Retourne une ligne sous la forme d'un objet.

object **mssql_fetch_object** (resource result)
[lien librairie]

mssql_fetch_object retourne un objet dont les propriétés contiennent les valeurs de la ligne, ou FALSE si il n'y a plus de ligne.

mssql_fetch_object est similaire à mssql_fetch_array , avec un 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).

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 .

6.58.13 mssql_fetch_row() : Lit une ligne comme un tableau.

array **mssql_fetch_row** (resource result)
[lien librairie]

mssql_fetch_row retourne un tableau qui contient les valeurs de la ligne à lire, 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 et mssql_result .

6.58.14 mssql_field_length() : Lit la longueur d'un champs.

`int` **mssql_field_length** (`resource result` , *`int offset`*)
[lien librairie]

6.58.15 mssql_field_name() : Lit le nom d'un champs.

`int` **mssql_field_name** (`resource result` , *`int offset`*)
[lien librairie]

6.58.16 mssql_field_seek() : Fixe l'offset du pointeur de champs.

`int` **mssql_field_seek** (`resource result` , `int field_offset`)
[lien librairie]

`mssql_field_seek` modifie la valeur du pointeur de champs. Le prochain appel à `mssql_fetch_field` qui ne précisera pas de numéro de champs, le champs fixé par `mssql_field_seek` sera retournée.

Voir aussi `mssql_fetch_field` .

6.58.17 mssql_field_type() : Lit le nom d'un champs.

`string` **mssql_field_type** (`resource result` , *`int offset`*)
[lien librairie]

6.58.18 mssql_free_result() : Libère la mémoire.

`int` **mssql_free_result** (`resource result`)
[lien librairie]

`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` .

6.58.19 mssql_get_last_message() : Retourne le dernier message d'erreur du serveur (min_message_severity?).

`string` **mssql_get_last_message** (`void`)
[lien librairie]

6.58.20 mssql_min_error_severity() : Fixe le niveau de sévérité des erreurs.

`void` **mssql_min_error_severity** (`int severity`)
[lien librairie]

6.58.21 mssql_min_message_severity() : Fixe le niveau de sévérité des messages d'erreurs.

`void` **mssql_min_message_severity** (`int severity`)
[lien librairie]

6.58.22 fbsql_next_result() : Déplace le pointeur interne vers le résultat suivant

`bool` **mssql_next_result** (`resource result_id`)
[lien librairie]

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 jeu 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 disponbile, 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);
?>
```

6.58.23 mssql_num_fields() : Retourne le nombre de champs dans un résultat.

`int` **mssql_num_fields** (`resource result`)
[lien librairie]

`mssql_num_fields` retourne le nombre de champs dans un résultat.

Voir aussi `mssql_query` , `mssql_fetch_field` et `mssql_num_rows` .

6.58.24 mssql_num_rows() : Retourne le nombre de lignes dans un résultat.

`int` **mssql_num_rows** (`resource result`)
[lien librairie]

mssql_num_rows retourne le nombre de lignes dans un résultat.

Voir aussi mssql_query et mssql_fetch_row .

6.58.25 mssql_pconnect() : Ouvre une connexion persistante à un serveur MS SQL.

resource **mssql_pconnect** (*string servername* , *string username* , *string password*)
[lien librairie]

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 lieux 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'.

6.58.26 mssql_query() : Envoie une requête SQL.

resource **mssql_query** (string query , *resource link_identifier*)
[lien librairie]

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_identifier (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 .

6.58.27 mssql_result() : Lit les données d'un résultat.

int **mssql_result** (mssql_query result , int i , mixed field)
[lien librairie]

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 champs peut être son nom littéral ou son offet, ou encore, le nom de la table + "." + le nom du champs, 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` .

6.58.28 mssql_select_db() : Sélectionne la base de données MS SQL.

`int` **mssql_select_db** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`mssql_select_db` retourne TRUE en cas de succès, et FALSE en cas d'erreur.

`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.

Voir aussi `mssql_connect` , `mssql_pconnect` et `mssql_query` .

6.59 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 utiliser ce module à vos risques et périls.

6.59.1 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.

6.59.2 Pré−requis

Pour utiliser Ming avec PHP, vous devez d'abord installer la librairy 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.

6.59.3 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.

6.59.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.59.5 Types de ressources

6.59.6 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.

`SWFBUTTON_HIT` (nombre ˆ virgule flottante)
`SWFBUTTON_DOWN` (nombre ˆ virgule flottante)
`SWFBUTTON_OVER` (nombre ˆ virgule flottante)
`SWFBUTTON_UP` (nombre ˆ virgule flottante)
`SWFBUTTON_MOUSEUPOUTSIDE` (nombre ˆ virgule flottante)
`SWFBUTTON_DRAGOVER` (nombre ˆ virgule flottante)
`SWFBUTTON_DRAGOUT` (nombre ˆ virgule flottante)
`SWFBUTTON_MOUSEUP` (nombre ˆ virgule flottante)
`SWFBUTTON_MOUSEDOWN` (nombre ˆ virgule flottante)
`SWFBUTTON_MOUSEOUT` (nombre ˆ virgule flottante)
`SWFBUTTON_MOUSEOVER` (nombre ˆ virgule flottante)
`SWFFILL_RADIAL_GRADIENT` (nombre ˆ virgule flottante)
`SWFFILL_LINEAR_GRADIENT` (nombre ˆ virgule flottante)
`SWFFILL_TILED_BITMAP` (nombre ˆ virgule flottante)
`SWFFILL_CLIPPED_BITMAP` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_HASLENGTH` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_NOEDIT` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_PASSWORD` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_MULTILINE` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_WORDWRAP` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_DRAWBOX` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_NOSELECT` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_HTML` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_ALIGN_LEFT` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_ALIGN_RIGHT` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_ALIGN_CENTER` (nombre ˆ virgule flottante)
`SWFTEXTFIELD_ALIGN_JUSTIFY` (nombre ˆ virgule flottante)
`SWFACTION_ONLOAD` (nombre ˆ virgule flottante)
`SWFACTION_ENTERFRAME` (nombre ˆ virgule flottante)
`SWFACTION_UNLOAD` (nombre ˆ virgule flottante)

SWFACTION_MOUSEMOVE (nombre ˆ virgule flottante)
SWFACTION_MOUSEDOWN (nombre ˆ virgule flottante)
SWFACTION_MOUSEUP (nombre ˆ virgule flottante)
SWFACTION_KEYDOWN (nombre ˆ virgule flottante)
SWFACTION_KEYUP (nombre ˆ virgule flottante)
SWFACTION_DATA (nombre ˆ virgule flottante)

6.59.7 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 utilisez, 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
- SWFAction
- SWFBitmap−>getHeight
- SWFBitmap−>getWidth
- SWFBitmap
- SWFbutton−>addAction
- SWFbutton−>addShape
- SWFbutton−>setAction
- SWFbutton−>setDown
- SWFbutton−>setHit
- SWFbutton−>setOver
- SWFbutton−>setUp
- SWFbutton
- SWFDisplayItem−>addColor
- SWFDisplayItem−>move
- SWFDisplayItem−>moveTo
- SWFDisplayItem−>multColor
- SWFDisplayItem−>remove
- SWFDisplayItem−>Rotate

- SWFDisplayItem->rotateTo
- SWFDisplayItem->scale
- SWFDisplayItem->scaleTo
- SWFDisplayItem->setDepth
- SWFDisplayItem->setName
- SWFDisplayItem->setRatio
- SWFDisplayItem->skewX
- SWFDisplayItem->skewXTo
- SWFDisplayItem->skewY
- SWFDisplayItem->skewYTo
- SWFDisplayItem
- SWFFill->moveTo
- SWFFill->rotateTo
- SWFFill->scaleTo
- 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

6.59.9 SWFBitmap−>getHeight() : Retourne la hauteur d'une bitmap.

`int` **swfbitmap−>getheight** (`void`)
[lien librairie]

`swfbitmap−>getheight` retourne la hauteur d'une bitmap, en pixels.

Voir aussi `swfbitmap−>getwidth` .

6.59.10 SWFBitmap−>getWidth() : Retourne la largeur d'une bitmap.

`int` **swfbitmap−>getwidth** (`void`)
[lien librairie]

`swfbitmap−>getwidth` retourne la largeur d'une bitmap, en pixels.

Voir aussi `swfbitmap−>getheight` .

6.59.11 SWFBitmap() : Crée un objet bitmap

`new` **swfbitmap** (`string filename` , *`int alphafilename`*)
[lien librairie]

`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();
?>
```

6.59.12 SWFbutton->addAction() : Ajoute une action au bouton

`void` **swfbutton->addaction** (`ressource action` , `int flags`)
[lien librairie]

`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` .

6.59.13 SWFbutton->addShape() : Ajoute une forme à un bouton

`void` **swfbutton->addshape** (`ressource shape` , `int flags`)
[lien librairie]

`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 déssiné est considéré comme accessible.

6.59.14 SWFbutton->setAction() : Assigne l'action du bouton

`void` **swfbutton->setaction** (`ressource action`)
[lien librairie]

`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` .

6.59.15 SWFbutton->setDown() : Alias de addShape(shape, SWFBUTTON_DOWN)

`void` **swfbutton->setdown** (`ressource shape`)
[lien librairie]

`swfbutton->setdown` est un alias pour `addShape(shape, SWFBUTTON_DOWN)` .

Voir aussi : `swfbutton->addshape` et `swfaction` .

6.59.16 SWFbutton−>setHit() : Alias de addShape(shape, SWFBUTTON_HIT)

`void` **swfbutton−>sethit** (`ressource shape`)
[lien librairie]

`swfbutton−>sethit` est un alias pour `addShape(shape, SWFBUTTON_HIT)` .

Voir aussi : `swfbutton−>addshape` et `swfaction` .

6.59.17 SWFbutton−>setOver() : Alias de addShape(shape, SWFBUTTON_OVER)

`void` **swfbutton−>setover** (`ressource shape`)
[lien librairie]

`swfbutton−>setover` est un alias pour `addShape(shape, SWFBUTTON_OVER)` .

Voir aussi : `swfbutton−>addshape` et `swfaction` .

6.59.18 SWFbutton−>setUp() : Alias de SWFbutton−>addShape(shape, SWFBUTTON_UP)

`void` **swfbutton−>setup** (`ressource shape`)
[lien librairie]

`swfbutton−>setup` est un alias pour `SWFbutton−>addShape(shape, SWFBUTTON_UP)` .

Voir aussi : `swfbutton−>addshape` et `swfaction` .

6.59.19 SWFbutton() : Crée un nouveau bouton.

`new` **swfbutton** (`void`)
[lien librairie]

`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);
  $i = $m->add($b);
  $i->moveTo(400,900);
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

Cet exemple simple illustre le déplacement d'un gros bouton rouge dans la fenêtre. Ce n'est pas du tirer−déposer, mais juste du tirer.

Exemple avec swfbutton−>addaction

```
<?php
  $s = new SWFShape();
  $s->setRightFill($s->addFill(0xff, 0, 0));
  $s->drawLine(1000,0);
  $s->drawLine(0,1000);
  $s->drawLine(−1000,0);
  $s->drawLine(0,−1000);
  $b = new SWFButton();
  $b->addShape($s, SWFBUTTON_HIT | SWFBUTTON_UP | SWFBUTTON_DOWN | SWFBUTTON_OVER);
  $b->addAction(new SWFAction("startDrag('/test', 0);"),
        SWFBUTTON_MOUSEDOWN);
// '0' signifie : ne pas verrouiller la souris
  $b->addAction(new SWFAction("stopDrag();"),
        SWFBUTTON_MOUSEUP | SWFBUTTON_MOUSEUPOUTSIDE);
  $p = new SWFSprite();
  $p->add($b);
  $p->nextFrame();
  $m = new SWFMovie();
  $i = $m->add($p);
  $i->setName('test');
  $i->moveTo(1000,1000);
  header('Content−type: application/x−shockwave−flash');
  $m->output();
?>
```

6.59.20 SWFDisplayItem−>addColor() : Ajoute une couleur à une transoformation.

`void` **swfdisplayitem−>addcolor** (*`int red`* , *`int green`* , *`int blue`* , *`int a`*)
[lien librairie]

`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` .

6.59.21 SWFDisplayItem−>move() : Déplace un objet en coordonnées relatives.

`void` **swfdisplayitem−>move** (`int dx` , `int dy`)
[lien librairie]

`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` .

6.59.22 SWFDisplayItem−>moveTo() : Déplace un objet en coordonnées globales.

`void` **swfdisplayitem−>moveto** (`int x` , `int y`)
[lien librairie]

`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` .

6.59.23 SWFDisplayItem−>multColor() : Multiplie la couleur de transformation.

`void` **swfdisplayitem−>multcolor** (*`int red`* , *`int green`* , *`int blue`* , *`int a`*)
[lien librairie]

`swfdisplayitem−>multcolor` multiplie la couleur de transformation par les valeurs données.

L'objet peut être `swfshape` , `swfbutton` , `swftext` ou `swfsprite` . Il doit avoir été ajouté à une animation avec la fonction `swfmovie−>add` .

Cet exemple simple modifie l'atmosphère de votre image, et en fait une scène d'Halloween (utilisez un paysage ou une image claire pour un meilleur effet)

Exemple avec `swfdisplayitem−>multcolor`

```
<?php
  $b = new SWFBitmap("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();
?>
```

6.59.24 SWFDisplayItem−>remove() : Supprime un objet d'une animation

`void` **swfdisplayitem−>remove** (`void`)
[lien librairie]

`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` .

6.59.25 SWFDisplayItem−>Rotate() : Fait tourner une forme relativement.

`void` **swfdisplayitem−>rotate** (`double ddegrees`)
[lien librairie]

`swfdisplayitem−>rotate` fait tourner la forme de `ddegrees` degrés, en plus de sa rotation courante.

L'objet peut être `swfshape` , `swfbutton` , `swftext` ou `swfsprite` . Il doit avoir été ajouté à une animation avec la fonction `swfmovie−>add` .

Voir aussi `swfdisplayitem−>rotateto` .

6.59.26 SWFDisplayItem−>rotateTo() : Tourne un objet en angle absolu.

`void` **swfdisplayitem−>rotateto** (`double degrees`)
[lien librairie]

`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 utilisez $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` .

6.59.27 SWFDisplayItem->scale() : Etire un objet relativement.

`void` **swfdisplayitem->scale** (`int dx` , `int dy`)
[lien librairie]

`swfdisplayitem->scale` étire un objet de (`dx` , `dy`), à partir de sa taille courante.

L'objet peut être `swfshape` , `swfbutton` , `swftext` ou `swfsprite` . Il doit avoir été ajouté à une animation avec la fonction `swfmovie->add` .

Voir aussi `swfdisplayitem->scaleto` .

6.59.28 SWFDisplayItem−>scaleTo() : Etire un objet en coordonnées globales.

`void` **swfdisplayitem−>scaleto** (`int x` , `int y`)
[lien librairie]

`swfdisplayitem−>scaleto` 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` .

6.59.29 SWFDisplayItem−>setDepth() : Modifie la place en profondeur (z−order)

`void` **swfdisplayitem−>setdepth** (`double depth`)
[lien librairie]

`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` .

6.59.30 SWFDisplayItem−>setName() : Nomme un objet

`void` **swfdisplayitem−>setname** (`string name`)
[lien librairie]

`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` .

6.59.31 SWFDisplayItem−>setRatio() : Modifie le ratio de l'objet.

`void` **swfdisplayitem−>setratio** (`double ratio`)
[lien librairie]

`swfdisplayitem−>setratio` modifie le ratio de l'objet, et le fixe à `ratio` . Uniquement utile pour les morphings.

L'objet peut être `swfshape` , `swfbutton` , `swftext` ou `swfsprite` . Il doit avoir été ajouté à une animation avec la fonction `swfmovie−>add` .

Cet exemple simple effectue un morphing délicat de trois cercles concentriques.

Exemple swfdisplayitem->setname

```
<?php
  $p = new SWFMorph();
  $g = new SWFGradient();
  $g->addEntry(0.0, 0, 0, 0);
  $g->addEntry(0.16, 0xff, 0xff, 0xff);
  $g->addEntry(0.32, 0, 0, 0);
  $g->addEntry(0.48, 0xff, 0xff, 0xff);
  $g->addEntry(0.64, 0, 0, 0);
  $g->addEntry(0.80, 0xff, 0xff, 0xff);
  $g->addEntry(1.00, 0, 0, 0);
  $s = $p->getShape1();
  $f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
  $f->scaleTo(0.05);
  $s->setLeftFill($f);
  $s->movePenTo(-160, -120);
  $s->drawLine(320, 0);
  $s->drawLine(0, 240);
  $s->drawLine(-320, 0);
  $s->drawLine(0, -240);
  $g = new SWFGradient();
  $g->addEntry(0.0, 0, 0, 0);
  $g->addEntry(0.16, 0xff, 0, 0);
  $g->addEntry(0.32, 0, 0, 0);
  $g->addEntry(0.48, 0, 0xff, 0);
  $g->addEntry(0.64, 0, 0, 0);
  $g->addEntry(0.80, 0, 0, 0xff);
  $g->addEntry(1.00, 0, 0, 0);
  $s = $p->getShape2();
  $f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
  $f->scaleTo(0.05);
  $f->skewXTo(1.0);
  $s->setLeftFill($f);
  $s->movePenTo(-160, -120);
  $s->drawLine(320, 0);
  $s->drawLine(0, 240);
  $s->drawLine(-320, 0);
  $s->drawLine(0, -240);
  $m = new SWFMovie();
  $m->setDimension(320, 240);
  $i = $m->add($p);
  $i->moveTo(160, 120);
  for($n=0; $n<=1.001; $n+=0.01)
  {
    $i->setRatio($n);
    $m->nextFrame();
  }
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

6.59.32 SWFDisplayItem->skewX() : Incline suivant les X relativement.

`void` **swfdisplayitem->skewx** (`double ddegrees`)
[lien librairie]

`swfdisplayitem->skewx` ajoute `ddegrees` à l'inclinaison courante (x-skew).

L'objet peut être `swfshape` , `swfbutton` , `swftext` ou `swfsprite` . Il doit avoir été ajouté à une animation avec la fonction `swfmovie->add` .

Voir aussi swfdisplayitem->skewx , swfdisplayitem->skewy and swfdisplayitem->skewyto .

6.59.33 SWFDisplayItem->skewXTo() : Incline suivant les X.

void **swfdisplayitem->skewxto** (double degrees)
[lien librairie]

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 .

6.59.34 SWFDisplayItem->skewY() : Incline suivant les Y relativement.

void **swfdisplayitem->skewy** (double ddegrees)
[lien librairie]

swfdisplayitem->skewy ajoute ddegrees 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 .

6.59.35 SWFDisplayItem->skewYTo() : Incline suivant les Y.

void **swfdisplayitem->skewyto** (double degrees)
[lien librairie]

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 .

6.59.36 SWFDisplayItem() : Crée un nouvel objet d'affichage displayitem.

new **swfdisplayitem** (void)
[lien librairie]

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−>scaleto , swfdisplayitem−>scale , swfdisplayitem−>rotate , swfdisplayitem−>rotateto , swfdisplayitem−>skewxto , swfdisplayitem−>skewx , swfdisplayitem−>skewyto swfdisplayitem−>skewyto , swfdisplayitem−>setdepth swfdisplayitem−>remove , swfdisplayitem−>setname swfdisplayitem−>setratio , swfdisplayitem−>addcolor et swfdisplayitem−>multcolor .

6.59.37 SWFFill−>moveTo() : Déplace l'origine de l'objet SWFFill

void **swffill−>moveto** (int x , int y)
[lien librairie]

swffill−>moveto déplace l'origine de la forme jusqu'au point de coordonnées globales (x , y).

6.59.38 SWFFill−>rotateTo() : Tourne la forme

void **swffill−>rotateto** (double degrees)
[lien librairie]

swffill−>rotateto tourne la forme depuis sont orientation initiale jusqu'à un angle de degrees degrés.

6.59.39 SWFFill−>scaleTo() : Modifie l'échelle de la forme

void **swffill−>scaleto** (int x , int y)
[lien librairie]

swffill−>scaleto modifie l'échelle de la forme de x dans le sens des abscisses et y dans le sens des ordonnées.

6.59.40 SWFFill−>skewXTo() : Incline (abscisses)

void **swffill−>skewxto** (double x)
[lien librairie]

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.

6.59.41 SWFFill−>skewYTo() : Incline (ordonnées)

`void` **swffill−>skewyto** (`double y`)
[lien librairie]

`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.

6.59.42 SWFFill() : Crée un objet de remplissage

`swffill` vous permet de transformer une image bitmap ou un gradient. Les objets `swffill` sont créé par `swfshape−>addfill` .

[lien librairie]

SWFFill a les méthodes suivantes : `swffill−>moveto` , `swffill−>scaleto` , `swffill−>rotateto` , `swffill−>skewxto` et `swffill−>skewyto` .

6.59.43 swffont−>getwidth() : Retourne la taille de la chaîne

`int` **swffont−>getwidth** (`string string`)
[lien librairie]

`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.

6.59.44 SWFFont() : Charge une police

`new` **swffont** (`string filename`)
[lien librairie]

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` .

6.59.45 SWFGradient->addEntry() : Ajoute une couleur à la liste du gradient.

`void` **swfgradient->addentry** (`double ratio` , `int red` , `int green` , `int blue` , *`int a`*)
[lien librairie]

`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.

6.59.46 SWFGradient() : Crée un objet gradient

`new` **swfgradient** (`void`)
[lien librairie]

`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();
?>
```

6.59.47 SWFMorph->getshape1() : Sélectionne la forme de départ

`mixed` **swfmorph->getshape1** (`void`)
[lien librairie]

`swfmorph->getshape1` sélectionne la forme de début de morphing. `swfmorph->getshape1` retourne un objet `swfshape` .

6.59.48 SWFMorph->getshape2() : Sélectionne la forme de fin

`mixed` **swfmorph->getshape2** (`void`)
[lien librairie]

`swfmorph->getshape2` sélectionne la forme de début de morphing. `swfmorph->getshape2` retourne un objet `swfshape` .

6.59.49 SWFMorph() : Crée un morphing.

`new` **swfmorph** (`void`)
[lien librairie]

`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();
?>
```

6.59.50 SWFMovie->add() : Ajoute un objet dans une animation.

`void` **swfmovie->add** (`ressource instance`)
[lien librairie]

`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 ajouté à une animation pour être utilisé.

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` .

6.59.51 SWFMovie−>nextframe() : Passe à l'image suivante.

`void` **swfmovie−>nextframe** (`void`)
[lien librairie]

`swfmovie−>setframes` passe à l'image suivante de l'animation.

6.59.52 SWFMovie−>output() : Envoie votre animation au navigateur.

`void` **swfmovie−>output** (`void`)
[lien librairie]

`swfmovie−>output` envoie votre animation au navigateur. En PHP, faîte 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.

6.59.53 SWFMovie−>remove() : Supprime un objet d'une animation.

`void` **swfmovie−>remove** (`ressource instance`)
[lien librairie]

`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` .

6.59.54 SWFMovie−>save() : Sauve dans un fichier.

`void` **swfmovie−>save** (`string filename`)
[lien librairie]

`swfmovie−>save` sauve votre animation dans le fichier `filename` .

Voir aussi `swfmovie−>output` .

6.59.55 SWFMovie−>setbackground() : Modifie la couleur de fond.

`void` **swfmovie−>setbackground** (`int red` , `int green` , `int blue`)
[lien librairie]

`swfmovie−>setbackground` modifie la couleur de fond. Pourquoi est−ce que cette fonction n'accepte pas de canal alpha? (réflechissez 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.

6.59.56 SWFMovie−>setdimension() : Modifie les dimensions de l'animation.

`void` **swfmovie−>setdimension** (`int width` , `int height`)
[lien librairie]

`swfmovie−>setdimension` modifie les dimensions de l'animation : `width` est la largeur et `height` la hauteur.

6.59.57 SWFMovie−>setframes() : Modifie le nombre total d'images dans l'animation.

`void` **swfmovie−>setframes** (`string numberofframes`)
[lien librairie]

`swfmovie−>setframes` modifie le nombre total d'images dans l'animation, et le fixe à `numberofframes` .

6.59.58 SWFMovie−>setrate() : Modifie la vitesse de l'animation.

`void` **swfmovie−>setrate** (`int rate`)
[lien librairie]

`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.

6.59.59 SWFMovie−>streammp3() : Envoie un fichier MP3 en streaming.

`void` **swfmovie−>streammp3** (`string mp3FileName`)
[lien librairie]

`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 l'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();
?>
```

6.59.60 SWFMovie() : Crée un objet 'animation'.

`new` **swfmovie** (`void`)
[lien librairie]

`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.

6.59.61 SWFShape->addFill() : Ajoute un remplissage plein à la forme.

`void` **swfshape->addfill** (`int red` , `int green` , `int blue` , *`int a`*)
[lien librairie]
`void` **swfshape->addfill** (`SWFbitmap bitmap` , *`int flags`*) `void` **swfshape->addfill** (`SWFGradient gradient` , *`int flags`*)

`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 défault, 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
  $p = new SWFMorph();
  $b = new SWFBitmap("alphafill.jpg");
  // utilisez vos propres bitmaps!
  $width = $b->getWidth();
  $height = $b->getHeight();
  $s = $p->getShape1();
  $f = $s->addFill($b, SWFFILL_TILED_BITMAP);
  $f->moveTo(-$width/2, -$height/4);
  $f->scaleTo(1.0, 0.5);
  $s->setLeftFill($f);
  $s->movePenTo(-$width/2, -$height/4);
  $s->drawLine($width, 0);
  $s->drawLine(0, $height/2);
  $s->drawLine(-$width, 0);
  $s->drawLine(0, -$height/2);
  $s = $p->getShape2();
  $f = $s->addFill($b, SWFFILL_TILED_BITMAP);
  // ces déplacements n'ont aucun effet
  $f->moveTo(-$width/4, -$height/2);
  $f->scaleTo(0.5, 1.0);
  $s->setLeftFill($f);
  $s->movePenTo(-$width/4, -$height/2);
  $s->drawLine($width/2, 0);
  $s->drawLine(0, $height);
  $s->drawLine(-$width/2, 0);
  $s->drawLine(0, -$height);
  $m = new SWFMovie();
  $m->setDimension($width, $height);
  $i = $m->add($p);
  $i->moveTo($width/2, $height/2);
  for($n=0; $n<1.001; $n+=0.03)
  {
    $i->setRatio($n);
    $m->nextFrame();
  }
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

6.59.62 SWFShape->drawCurve() : Dessine une courbe relativement.

`void` **swfshape->drawcurve** (`int controldx` , `int controldy` , `int anchordx` , `int anchordy`)

[lien librairie]

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 dirigie sur le point d'ancrrage.

Voir aussi swfshape->drawlineto , swfshape->drawline , swfshape->movepento et swfshape->movepen .

6.59.63 SWFShape->drawCurveTo() : Dessine une courbe.

void **swfshape->drawcurveto** (int controlx , int controly , int anchorx , int anchory)
[lien librairie]

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 dirigie sur le point d'ancrrage.

Voir aussi swfshape->drawlineto , swfshape->drawline , swfshape->movepento et swfshape->movepen .

6.59.64 SWFShape->drawLine() : Dessine une ligne relativement.

void **swfshape->drawline** (int dx , int dy)
[lien librairie]

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 .

6.59.65 SWFShape->drawLineTo() : Dessine une ligne.

void **swfshape->drawlineto** (int x , int y)
[lien librairie]

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 .

6.59.66 SWFShape−>movePen() : Déplace le stylo relativement.

`void` **swfshape−>movepen** (`int dx` , `int dy`)
[lien librairie]

`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` .

6.59.67 SWFShape−>movePenTo() : Déplace le stylo.

`void` **swfshape−>movepento** (`int x` , `int y`)
[lien librairie]

`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` .

6.59.68 SWFShape−>setLeftFill() : Modifie la couleur de rastérisation de gauche.

`void` **swfshape−>setleftfill** (`swfgradient fill`)
[lien librairie]
`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 rastérisation 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 défnissez 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` .

6.59.69 SWFShape−>setLine() : Modifie le style de ligne de la forme.

`void` **swfshape−>setline** (`int width` , *`int red`* , *`int green`* , *`int blue`* , *`int a`*)
[lien librairie]

`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, '&#64;');
  $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("!#%*&#64;")/2, 1000+$f->getAscent()/2);
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

6.59.70 SWFShape−>setRightFill() : Modifie la couleur de rastériation de droite.

`void` **swfshape−>setrightfill** (`swfgradient fill`)
[lien librairie]
`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]));` .

6.59.71 SWFShape() : Crée une nouvelle forme.

`new` **swfshape** (`void`)
[lien librairie]

`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();
?>
```

6.59.72 SWFSprite−>add() : Ajoute un objet à un sprite

`mixed` **swfsprite−>add** (`ressource object`)
[lien librairie]

`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.

6.59.73 SWFSprite−>nextframe() : Va à la prochaine image du sprite.

`void` **swfsprite−>nextframe** (`void`)
[lien librairie]

`swfsprite−>setframes` se déplace à la prochaine image du sprite.

6.59.74 SWFSprite−>remove() : Supprime un objet dans un sprite

`void` **swfsprite−>remove** (`ressource object`)
[lien librairie]

`swfsprite−>remove` supprime une `swfshape` , un `swfbutton` , un `swftext` , une `swfaction` ou un `swfsprite` du sprite courant.

6.59.75 SWFSprite−>setframes() : Fixe le nombre maximum d'image dans le sprite.

`void` **swfsprite−>setframes** (`int numberofframes`)
[lien librairie]

`swfsprite−>setframes` fixe le nombre total d'images de l'animation à `numberofframes` .

6.59.76 SWFSprite() : Crée un sprite

`new` **swfsprite** (`void`)
[lien librairie]

`swfsprite` sont aussi connue sous le nom de "clip" : ils permettent la création d'objet animé dans une animation, avec un scénario propre. De ce fait, un sprite a les mêmes méthodes qu'une animation.

`swfsprite` a les méthodes suivantes : `swfsprite−>add` , `swfsprite−>remove` , `swfsprite−>nextframe` et `swfsprite−>setframes` .

Ce exemple pratique fait tourner un superbe carré rouge.

Exemple de `swfsprite`

```
<?php
  $s = new SWFShape();
  $s->setRightFill($s->addFill(0xff, 0, 0));
  $s->movePenTo(-500,-500);
  $s->drawLineTo(500,-500);
  $s->drawLineTo(500,500);
  $s->drawLineTo(-500,500);
  $s->drawLineTo(-500,-500);
  $p = new SWFSprite();
  $i = $p->add($s);
```

```
  $p->nextFrame();
  $i->rotate(15);
  $p->nextFrame();
  $i->rotate(15);
  $p->nextFrame();
  $i->rotate(15);
  $p->nextFrame();
  $i->rotate(15);
  $p->nextFrame();
  $i->rotate(15);
  $p->nextFrame();
  $m = new SWFMovie();
  $i = $m->add($p);
  $i->moveTo(1500,1000);
  $i->setName("blah");
  $m->setBackground(0xff, 0xff, 0xff);
  $m->setDimension(3000,2000);
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

6.59.77 SWFText->addString() : Ajoute du texte

`void` **swftext->addstring** (`string string`)
[lien librairie]

`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.

6.59.78 SWFText->getWidth() : Calcule la longueur d'une chaîne

`void` **swftext->addstring** (`string string`)
[lien librairie]

`swftext->addstring` retourne la taille de la chaîne `string` , une fois qu'elle est dessinée avec la police et l'espaceement courant.

6.59.79 SWFText->moveTo() : Déplace le stylo de texte

`void` **swftext->moveto** (`int x` , `int y`)
[lien librairie]

`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.

6.59.80 SWFText->setColor() : Modifie la couleur de la police

`void` **swftext->setcolor** (`int red` , `int green` , `int blue` , *`int a`*)
[lien librairie]

`swftext->setspacing` change la couleur de la police courante. Par défaut, c'est noir. La couleur est représentée avec la convention RGB.

6.59.81 SWFText->setFont() : Sélectionne la police courante

`void` **swftext->setfont** (`string font`)
[lien librairie]

`swftext->setfont` remplace la police courante par `font` .

6.59.82 SWFText->setHeight() : Modifie la hauteur de la police courante

`void` **swftext->setheight** (`int height`)
[lien librairie]

`swftext->setheight` fixe la hauteur courante de la police courante à `height` . Par défaut, c'est 240.

6.59.83 SWFText->setSpacing() : Modifie l'espacement de police

`void` **swftext->setspacing** (`double spacing`)
[lien librairie]

`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.

6.59.84 SWFText() : Crée un nouvel objet texte.

`new` **swftext** (`void`)
[lien librairie]

`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.

Exemle 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();
?>
```

6.59.85 SWFTextField->addstring() : Ajoute au texte

`void` **swftextfield->addstring** (`string string`)
[lien librairie]

`swftextfield->setname` concatène la chaîne `string` avec la chaîne courante.

6.59.86 SWFTextField->align() : Modifie l'alignement du texte

`void` **swftextfield->align** (`int alignement`)
[lien librairie]

`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.

6.59.87 SWFTextField->setbounds() : Sélectionne la largeur et hauteur du champs

`void` **swftextfield->setbounds** (`int width` , `int height`)
[lien librairie]

`swftextfield->setbounds` fixe la longueu du champs 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!!).

6.59.88 SWFTextField->setcolor() : Modifie la couleur du champs texte

`void` **swftextfield->setcolor** (`int red` , `int green` , `int blue` , *`int a`*)
[lien librairie]

`swftextfield->setcolor` modifie la couleur du champs texte, en la remplaçant par la couleur fournie. Par défaut, c'est noir opaque. Les couleurs sont représentées en convention RGB.

6.59.89 SWFTextField->setFont() : Modifie la police du champs

`void` **swftextfield->setfont** (`string font`)
[lien librairie]

`swftextfield->setfont` remplace la police courante par la police `font` (police client?).

6.59.90 SWFTextField−>setHeight() : Modifie la hauteur de la police du champs texte.

`void` **swftextfield−>setheight** (`int height`)
[lien librairie]

`swftextfield−>setheight` modifie la hauteur de la police du champs texte par `height` . Par défaut, c'est 240.

6.59.91 SWFTextField−>setindentation() : Modifie l'indentation de la première ligne.

`void` **swftextfield−>setindentation** (`int width`)
[lien librairie]

`swftextfield−>setindentation` modifie l'indentation de la première ligne du champs texte, en la fixant à `width` .

6.59.92 SWFTextField−>setLeftMargin() : Modifie la marge de gauche.

`void` **swftextfield−>setleftmargin** (`int width`)
[lien librairie]

`swftextfield−>setleftmargin` modifie la marge de gauche du champs texte à `width` . Par défaut, c'est 0.

6.59.93 SWFTextField−>setLineSpacing() : Modifie l'espacement de lignes.

`void` **swftextfield−>setlinespacing** (`int height`)
[lien librairie]

`swftextfield−>setlinespacing` modifie l'espacement de lignes, en le fixant à `height` . Par défaut, c'est 40.

6.59.94 SWFTextField−>setMargins() : Modifie les marges du champs texte.

`void` **swftextfield−>setmargins** (`int left` , `int right`)
[lien librairie]

`swftextfield−>setmargins` modifie les deux marges du champs texte : `left` sera la nouvelle largeur de la marge de gauche, et `right` , celle de gauche. Par défaut, elles sont toutes les deux à 0.

6.59.95 SWFTextField−>setname() : Nomme le champs texte

`void` **swftextfield−>setname** (`string name`)
[lien librairie]

`swftextfield−>setname` baptise le champs texte `name` . Cela servira pour les formulaires et les actions.

6.59.96 SWFTextField−>setrightMargin() : Modifie la marge de droite.

`void` **swftextfield−>setrightmargin** (`int width`)
[lien librairie]

`swftextfield−>setrightmargin` modifie la marge de gauche du champs texte à `width` . Par défaut, c'est

6.59.97 SWFTextField() : Crée un nouveau champs texte

`new` **swftextfield** (*`int flags`*)
[lien librairie]

`swftextfield` crée un nouveau champs 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 champs. Ils peuvent prendre les valeurs suivantes :

- SWFTEXTFIELD_NOEDIT : indique que le champs ne doit pas être éditable.
- SWFTEXTFIELD_PASSWORD : indique que c'est un champs mot de passe
- SWFTEXTFIELD_DRAWBOX : dessine le contour du champs
- SWFTEXTFIELD_MULTILINE : autorise les lignes multiples
- SWFTEXTFIELD_WORDWRAP : autorise la mise en forme du texte
- SWFTEXTFIELD_NOSELECT : rend le champs non−sélectionnable

Les flags peuvent être combinés avec l'opérateur OR . Par exemple :

```
<?php
$t = newSWFTextField(SWFTEXTFIELD_PASSWORD | SWFTEXTFIELD_NOEDIT);
?>
```

crée un champs de mot de passe totalement inéditable (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` .

6.60 Fonctions diverses

6.60.1 Introduction

Ces fonctions ont été placées là, car elles ne rentraient dans aucune catégorie adéquate.

6.60.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.60.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.60.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
ignore_user_abort	"0"	PHP_INI_ALL
highlight.string	#CC0000	PHP_INI_ALL
highlight.comment	#FF9900	PHP_INI_ALL
highlight.keyword	#006600	PHP_INI_ALL
highlight.bg	#FFFFFF	PHP_INI_ALL
highlight.default	#0000CC	PHP_INI_ALL
highlight.html	#000000	PHP_INI_ALL
browscap	NULL	PHP_INI_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

`ignore_user_abort` `boolean`
: 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` .

6.60.5 Types de ressources

Cette extension ne définit aucune ressource.

6.60.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

- 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
- leak
- pack
- show_source
- sleep
- uniqid
- unpack
- usleep

6.60.8 connection_status() : Retourne les bits de status de la connexion.

`int` **connection_status** (`void`)
[lien librairie]

`connection_status` retourne les bits de statut de la connexion. Reportez−vous à la section gestion des connexions pour plus de détails.

6.60.9 connection_timeout() : Indique si le script a expiré.

`bool` **connection_timeout** (`void`)
[lien librairie]

`connection_timeout` retourne `TRUE` si le script a expiré. Reportez−vous à la section gestion des connexions pour plus de détails.

6.60.10 constant() : Retourne la valeur d'une constante

`mixed` **constant** (`string  name`)
[lien librairie]

`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 son nom que durant l'exécution du script. Par exemple, ce nom peut être le résultat d'une fonction.

Exempe avec `constant`

```
<?php
  define ("MAXSIZE", 100);
  echo MAXSIZE;
  echo constant("MAXSIZE");   // identique à la ligne précédente
?>
```

Voir aussi `define` , `defined` et la section sur les constantes .

6.60.11 define() : Définit une constante.

`boolean` **define** (`string  name` , `mixed  value` , *`int  case_insensitive`*)
[lien librairie]

`define` définit une constante, de la même façon qu'une variable, sauf que :

- Les constantes ne commencent pas par le signe '$'
- Les constantes sont accessibles partout, de manière globale.
- Les constantes ne peuvent pas être redéfinies, ou indéfinies, une fois qu'elles ont été définies.
- Les constantes ne représentent que des valeurs scalaires : il n'est pas possible de définir des tableaux ou des objets.

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("CONSTANTE", "Bonjour le monde.");
  echo CONSTANTE;
  // affiche "Bonjour le monde."
?>
```

`define` retoune `TRUE` en cas de succès et `FALSE` sinon.

Voir aussi `defined` et la section sur les constantes .

6.60.12 defined() : Vérifie qu'une constante existe.

`int` **defined** `( string name )`
[lien librairie]

`defined` retourne `TRUE` si la constante nommée `name` a été définie, et `FALSE` sinon.

Voir aussi `define` et la section sur les constantes .

6.60.13 die() : Alias de la fonction `exit`

`void` **die** `( string message )`
[lien librairie]

`die` est un alias de `exit` .

6.60.14 eval() : Evalue une chaîne comme un script PHP.

`void` **eval** `( string code_str )`
[lien librairie]

`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.

Exemple avec `eval` − inclusion 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.
```

6.60.15 exit() : Termine le script courant.

`void` **exit** (`string status`)
[lien librairie]
`void` **exit** (`int status`)

Note
`exit` n'est pas une véritable fonction, mais un élément de langage.

`exit` termine l'éxécution 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");
?>
```

6.60.16 get_browser() : Indique de quoi est capable le navigateur client.

`object` **get_browser** (*`string user_agent`*)
[lien librairie]

`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
function list_array ($array) {
    while (list ($key, $value) = each ($array)) {
        $str .= "<B>$key:</B> $value<br>\n";
    }
    return $str;
}
```

```
echo "$HTTP_USER_AGENT<hr>\n";
$browser = get_browser();
echo list_array ((array) $browser);
?>
```

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\.5.*<br>

<B>parent:</B> Netscape 4.0<br>

<B>platform:</B> Unknown<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 doit mener au fichier `browscap.ini` .

Pour plus d'informations, (y compris pour les endroits où charger le fichier `browscap.ini`), suivez la FAQ PHP à http://www.php.net/FAQ.html .

Note
Browscap a été ajouté en PHP 3.0b2.

6.60.17 highlight_file() : Colorisation de la syntaxe d'un fichier

`boolean` **highlight_file** (`string filename`)
[lien librairie]

`highlight_file` affiche la syntaxe colorisée du fichier `filename` , en utilisant les couleurs définies dans le moteur interne de PHP.

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.

Voir aussi `highlight_string` et `show_source` .

6.60.18 highlight_string() : Applique la syntaxe colorisée à du code PHP

`mixed` **highlight_string** (`string str` , *`bool return`*)
[lien librairie]

`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 immédaitement. 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` .

6.60.19 ignore_user_abort() : Active l'option décidant si, lors de la déconnexion du client, le script doit poursuivre son exécution ou non.

`int` **ignore_user_abort** (*`int setting`*)
[lien librairie]

`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 gestion des connexions dans le chapitre caractéristiques pour une description plus complète des manipulations de connexion en PHP.

6.60.20 leak() : Fuite de mémoire.

`void` **leak** (`int bytes`)
[lien librairie]

`leak` crée une fuite de mémoire.

`leak` est pratique pour débugger le gestionnaire de mémoire, qui doit nettoyer automatiquement les fuites de mémoire après chaque requête.

6.60.21 pack() : Compacte des données dans une chaîne binaire.

`string` **pack** (`string format` , *`mixed args`*)
[lien librairie]

`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 cours non signé (toujours 16 bits, ordre des bits little endian)
- i entier signé (taille et ordre des bits dépendants de la machine)
- I entier non signé (taille et ordre des bits dépendants de la machine)
- l entier long signé (toujours 32 bits, ordre des bits dépendant de la machine)
- L entier long non signé (toujours 32 bits, ordre des bits dépendant de la machine)
- N entier long non signé (toujours 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);
?>
```

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.

6.60.22 show_source() : Colorisation de la syntaxe d'un fichier

`void` **show_source** (`string filename`)
[lien librairie]

`show_source` affiche la syntaxe colorisée du fichier `filename` , en utilisant les couleurs définies dans le moteur interne de PHP.

Note
`show_source` est un alias de `highlight_file`

Voir aussi `highlight_string` et `highlight_file` .

6.60.23 sleep() : Retarde l'exécution.

`void` **sleep** (`int seconds`)
[lien librairie]

`sleep` retarde l'exécution du programme pendant `seconds` secondes.

Voir aussi `usleep` .

6.60.24 uniqid() : Génère un identifiant unique.

`int` **uniqid** (`string prefix` , *`boolean lcg`*)
[lien librairie]

`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 :

```php
<?php
  $token = md5 (uniqid (""));
  // pas de section aléatoire.
  $better_token = md5 (uniqid (rand()));
  // mieux, difficile à deviner
?>
```

Ceci va créer un identifiant de 32 caractères (un nombre hexadécimal de 128) qui sera très difficile à prédire.

6.60.25 unpack() : Déconditionne des données depuis une chaîne binaire.

`array` **unpack** (`string format` , `string data`)
[lien librairie]

`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
<?php
  $array = unpack ("c2chars/nint", $binarydata);
?>
```

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 ".

6.60.26 usleep() : Retarde l'exécution en micro−secondes

`void` **usleep** (`int micro_seconds`)
[lien librairie]

`sleep` retarde l'exécution du programme pendant `micro_seconds` micro−secondes.

Voir aussi `sleep` .

Note

`sleep` est inopérante sous Windows

6.61 mnoGoSearch

6.61.1 Introduction

Ces fonctions donnent l'accès à mnoGoSearch (anciennement UdmSearch), moteur de recherche du monde libre. mnoGoSearch est une 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 référéncés, 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/ .

6.61.2 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.

6.61.3 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 : `−−with−mnogosearch −−with−mysql=/usr` .

6.61.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.61.5 Types de ressources

6.61.6 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 (nombre ˆ virgule flottante)
UDM_FIELD_URL (nombre ˆ virgule flottante)
UDM_FIELD_CONTENT (nombre ˆ virgule flottante)
UDM_FIELD_TITLE (nombre ˆ virgule flottante)
UDM_FIELD_KEYWORDS (nombre ˆ virgule flottante)
UDM_FIELD_DESC (nombre ˆ virgule flottante)
UDM_FIELD_DESCRIPTION (nombre ˆ virgule flottante)
UDM_FIELD_TEXT (nombre ˆ virgule flottante)
UDM_FIELD_SIZE (nombre ˆ virgule flottante)
UDM_FIELD_RATING (nombre ˆ virgule flottante)
UDM_FIELD_SCORE (nombre ˆ virgule flottante)
UDM_FIELD_MODIFIED (nombre ˆ virgule flottante)
UDM_FIELD_ORDER (nombre ˆ virgule flottante)
UDM_FIELD_CRC (nombre ˆ virgule flottante)
UDM_FIELD_CATEGORY (nombre ˆ virgule flottante)
UDM_FIELD_LANG (nombre ˆ virgule flottante)
UDM_FIELD_CHARSET (nombre ˆ virgule flottante)
UDM_PARAM_PAGE_SIZE (nombre ˆ virgule flottante)
UDM_PARAM_PAGE_NUM (nombre ˆ virgule flottante)
UDM_PARAM_SEARCH_MODE (nombre ˆ virgule flottante)
UDM_PARAM_CACHE_MODE (nombre ˆ virgule flottante)
UDM_PARAM_TRACK_MODE (nombre ˆ virgule flottante)
UDM_PARAM_PHRASE_MODE (nombre ˆ virgule flottante)
UDM_PARAM_CHARSET (nombre ˆ virgule flottante)
UDM_PARAM_LOCAL_CHARSET (nombre ˆ virgule flottante)
UDM_PARAM_BROWSER_CHARSET (nombre ˆ virgule flottante)
UDM_PARAM_STOPTABLE (nombre ˆ virgule flottante)
UDM_PARAM_STOP_TABLE (nombre ˆ virgule flottante)
UDM_PARAM_STOPFILE (nombre ˆ virgule flottante)
UDM_PARAM_STOP_FILE (nombre ˆ virgule flottante)
UDM_PARAM_WEIGHT_FACTOR (nombre ˆ virgule flottante)
UDM_PARAM_WORD_MATCH (nombre ˆ virgule flottante)
UDM_PARAM_MAX_WORD_LEN (nombre ˆ virgule flottante)
UDM_PARAM_MAX_WORDLEN (nombre ˆ virgule flottante)
UDM_PARAM_MIN_WORD_LEN (nombre ˆ virgule flottante)
UDM_PARAM_MIN_WORDLEN (nombre ˆ virgule flottante)
UDM_PARAM_ISPELL_PREFIXES (nombre ˆ virgule flottante)
UDM_PARAM_ISPELL_PREFIX (nombre ˆ virgule flottante)
UDM_PARAM_PREFIXES (nombre ˆ virgule flottante)
UDM_PARAM_PREFIX (nombre ˆ virgule flottante)
UDM_PARAM_CROSS_WORDS (nombre ˆ virgule flottante)
UDM_PARAM_CROSSWORDS (nombre ˆ virgule flottante)
UDM_PARAM_VARDIR (nombre ˆ virgule flottante)
UDM_PARAM_DATADIR (nombre ˆ virgule flottante)
UDM_PARAM_HLBEG (nombre ˆ virgule flottante)
UDM_PARAM_HLEND (nombre ˆ virgule flottante)
UDM_PARAM_SYNONYM (nombre ˆ virgule flottante)
UDM_PARAM_SEARCHD (nombre ˆ virgule flottante)
UDM_PARAM_QSTRING (nombre ˆ virgule flottante)

UDM_PARAM_REMOTE_ADDR (nombre ˆ virgule flottante)
UDM_LIMIT_CAT (nombre ˆ virgule flottante)
UDM_LIMIT_URL (nombre ˆ virgule flottante)
UDM_LIMIT_TAG (nombre ˆ virgule flottante)
UDM_LIMIT_LANG (nombre ˆ virgule flottante)
UDM_LIMIT_DATE (nombre ˆ virgule flottante)
UDM_PARAM_FOUND (nombre ˆ virgule flottante)
UDM_PARAM_NUM_ROWS (nombre ˆ virgule flottante)
UDM_PARAM_WORDINFO (nombre ˆ virgule flottante)
UDM_PARAM_WORD_INFO (nombre ˆ virgule flottante)
UDM_PARAM_SEARCHTIME (nombre ˆ virgule flottante)
UDM_PARAM_SEARCH_TIME (nombre ˆ virgule flottante)
UDM_PARAM_FIRST_DOC (nombre ˆ virgule flottante)
UDM_PARAM_LAST_DOC (nombre ˆ virgule flottante)
UDM_MODE_ALL (nombre ˆ virgule flottante)
UDM_MODE_ANY (nombre ˆ virgule flottante)
UDM_MODE_BOOL (nombre ˆ virgule flottante)
UDM_MODE_PHRASE (nombre ˆ virgule flottante)
UDM_CACHE_ENABLED (nombre ˆ virgule flottante)
UDM_CACHE_DISABLED (nombre ˆ virgule flottante)
UDM_TRACK_ENABLED (nombre ˆ virgule flottante)
UDM_TRACK_DISABLED (nombre ˆ virgule flottante)
UDM_PHRASE_ENABLED (nombre ˆ virgule flottante)
UDM_PHRASE_DISABLED (nombre ˆ virgule flottante)
UDM_CROSS_WORDS_ENABLED (nombre ˆ virgule flottante)
UDM_CROSSWORDS_ENABLED (nombre ˆ virgule flottante)
UDM_CROSS_WORDS_DISABLED (nombre ˆ virgule flottante)
UDM_CROSSWORDS_DISABLED (nombre ˆ virgule flottante)
UDM_PREFIXES_ENABLED (nombre ˆ virgule flottante)
UDM_PREFIX_ENABLED (nombre ˆ virgule flottante)
UDM_ISPELL_PREFIXES_ENABLED (nombre ˆ virgule flottante)
UDM_ISPELL_PREFIX_ENABLED (nombre ˆ virgule flottante)
UDM_PREFIXES_DISABLED (nombre ˆ virgule flottante)
UDM_PREFIX_DISABLED (nombre ˆ virgule flottante)
UDM_ISPELL_PREFIXES_DISABLED (nombre ˆ virgule flottante)
UDM_ISPELL_PREFIX_DISABLED (nombre ˆ virgule flottante)
UDM_ISPELL_TYPE_AFFIX (nombre ˆ virgule flottante)
UDM_ISPELL_TYPE_SPELL (nombre ˆ virgule flottante)
UDM_ISPELL_TYPE_DB (nombre ˆ virgule flottante)
UDM_ISPELL_TYPE_SERVER (nombre ˆ virgule flottante)
UDM_MATCH_WORD (nombre ˆ virgule flottante)
UDM_MATCH_BEGIN (nombre ˆ virgule flottante)
UDM_MATCH_SUBSTR (nombre ˆ virgule flottante)
UDM_MATCH_END (nombre ˆ virgule flottante)

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- udm_add_search_limit
- Udm_Alloc_Agent
- udm_api_version

- udm_cat_list
- udm_cat_path
- udm_clear_search_limits
- 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_set_agent_param

6.61.8 Udm_Alloc_Agent() : Alloue une session mnoGoSearch

`int` **udm_alloc_agent** (`string dbaddr` , *`string dbmode`*)
[lien librairie]

`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 sotckés 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, plutot que des 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.

6.61.9 udm_api_version() : Lit la version des API mnoGoSearch.

`int` **udm_api_version** (`void`)
[lien librairie]

`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`

```
if (Udm_Api_Version() >= 30111) {
  print  "Total number of urls in database: ".Udm_Get_Doc_Count($udm)."<br>\n";
}
```

6.61.10 udm_cat_list() : Liste toutes les catégories soeurs d'une catégorie.

`array` **udm_cat_list** (`int  agent` , `string  category`)
[lien librairie]

`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
  ...
```

```
<?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];
    $cat_list .= "<a href=\"$PHP_SELF?cat=$path\">$name</a><br>";
  }
>
```

6.61.11 udm_cat_path() : Lit le chemin de la catégorie courante.

`array` **udm_cat_path** (`int agent` , `string category`)
[lien librairie]

`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 le chemin de la catégorie courante avec le format suivant : '> Root > Sport > Foot > PSG'

```
<?php
  $cat_path_arr=Udm_Cat_Path($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];
    $cat_path .= " > <a href=\"$PHP_SELF?cat=$path\">$name</a> ";
  }
>
```

6.61.12 udm_clear_search_limits() : Annule toutes les limitations de recherche

`int` **udm_clear_search_limits** (`int agent`)
[lien librairie]

udm_clear_search_limits annule toutes les limitations de recherche imposées, et retourne TRUE .

6.61.13 Udm_Errno() : Numéro d'erreur mnoGoSearch

int **udm_errno** (int agent)
[lien librairie]

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 .

6.61.14 Udm_Error() : Message d'erreur mnoGoSearch

string **udm_error** (int agent)
[lien librairie]

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 .

6.61.15 Udm_Find() : Effectue une recherche

int **udm_find** (int agent , string query)
[lien librairie]

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 recheche retourne la lsite 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 booléean 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 | msql) & ~postgres". Le mode par requête est simple et puissant à la fois. Vous pouvez utiliser les commandes booléennes habituelles avec ce mode.

6.61.16 Udm_Free_Agent() : Détruit une session mnoGoSearch

`int` **udm_free_agent** (`int agent`)
[lien librairie]

`udm_free_res` retourne `TRUE` en cas de succès, `FALSE` sinon.

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` .

6.61.17 udm_free_ispell_data() : Libère la mémoire allouée pour ispell

`int` **udm_free_ispell_data** (`int agent`)
[lien librairie]

`udm_free_ispell_data` retourne toujours `TRUE` .

`agent` − Agent mnoGoSearch obtenu après un appel à `udm_alloc_agent` .

Note
`udm_free_ispell_data` est supportée à partir de la version 3.1.12 de mnoGoSearch et elle ne fait strictement rien avec les versions précédentes.

6.61.18 Udm_Free_Res() : Libère un résultat mnoGoSearch

`int` **udm_free_res** (`int res`)
[lien librairie]

`udm_free_res` retourne `TRUE` en cas de succès, `FALSE` sinon.

Le paramètre `res` est un identifiant de résultat, obtenu après un appel à `Udm−Find` .

`udm_free_res` libère la mémoire de tous les résultats générés.

6.61.19 udm_get_doc_count() : Lit le nombre total de documents dans les bases.

`int` **udm_get_doc_count** (`int agent`)
[lien librairie]

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
udm_get_doc_count est supporté à partir de la version mnoGoSearch 3.1.11 ou plus récent.

6.61.20 Udm_Get_Res_Field() : Lit un champs de résultat mnoGoSearch

int **udm_get_res_field** (int res , int row , int field)
[lien librairie]

udm_alloc_agent retourne la valeur du champs 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 champs, 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.

6.61.21 Udm_Get_Res_Param() : Lit les paramètres de résultats mnoGoSearch

int **udm_get_res_param** (int res , int param)
[lien librairie]

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 libre" 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.

6.61.22 udm_load_ispell_data() : Charge les données ispell

`int` **udm_load_ispell_data** (`int agent` , `int var` , `string val1` , `string val2` , `int flag`)
[lien librairie]

`udm_load_ispell_data` charge des données ispell. `udm_load_ispell_data` retourne TRUE en cas de succès, et FALSE en cas d'erreur.

`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 `udm_load_ispell_data` , et il ne vaut pas la peine de trier les valeurs après chaque appel, mais uniquement à la fin. Etant donné qu'en mode DB, toutes les données sont chargées en une seule fois, ce paramètre doit avoir la valeur de `1` . Dans ce mode, en cas d'erreur, par exemple si la table ispell est absente, la fonction retournera FALSE et le code d'erreur, avec son message, seront accessibles avec `udm_error` et `udm_errno` .

  Exemple avec `udm_load_ispell_data`

```
if (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_DB,'','',1)) {
  printf("Error #%d: '%s'\n",Udm_Errno($udm),Udm_Error($udm));
  exit;
}
```

- 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 udm_load_ispell_data

```
        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($udm,UDM_ISPELL_TYPE_SPELL,'en','/opt/ispell/en.dict',
            (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_SPELL,'ru','/opt/ispell/ru.dict',
           exit;
        }
```

Note
flag prend la valeur 1 si c'est le dernier appel à cette fonction.

- UDM_ISPELL_TYPE_SPELL indique que les données ispell doivent être chargées depuis un fichier, et initie le chargement du dictionnaire. Dans ce cas, val1 définit le code langue sur deux lettres, et val2 le chemin du fichier. Notez que si vous utilisez un chemin relatif, le module recherche les fichiers non pas dans UDM_CONF_DIR, mais directement avec le chemin courant, où le script est exécuté. En cas d'erreur avec ce mode, si le fichier est absent, la fonction retourne FALSE , et un message d'erreur sera affiché. Les messages d'erreur ne sont pas accessibles avec udm_error et udm_errno , puisque ces fonctions ne traitent que les messages SQL. Reportez−vous à la description du paramètre flag .

  Exemple avec udm_load_ispell_data

```
        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($udm,UDM_ISPELL_TYPE_SPELL,'en','/opt/ispell/en.dict',
            (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_SPELL,'ru','/opt/ispell/ru.dict',
           exit;
        }
```

Note
flag prend la valeur 1 si c'est le dernier appel à cette fonction.

- UDM_ISPELL_TYPE_SERVER active le support des serveurs ispell. val1 indique alors l'adresse de l'hôte qui supporte le serveur ispall. 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 Spelld lisent les données d'orthographe dans une configuration séparée (par défaut /usr/local/mnogosearch/etc/spelld.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'éxécution, 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

```
if (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_SERVER,'','',1)) {
    printf("Error loading ispell data from server<br>\n");
    exit;
}
```

6.61.23 udm_set_agent_param() : Modifie les paramètres de l'agent mnoGoSearch

int **udm_set_agent_param** (int agent , int var , string val)
[lien librairie]

udm_set_agent_param retourne TRUE en cas de succès et FALSE sinon.
udm_set_agent_param définit les paramètres de l'agent mnoGoSearch.

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. Lorque deux requête 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 requete 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 utiliser 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 héxadécimaux, sous la forme ABCDE. Chaque chiffre est un facteur correspondant à un poids affecté à une partie du document. Por la situation décrite ci−dessus,

  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.

6.62 mSQL

6.62.1 Introduction

Ces fonctions vous permettent d'accéder aux bases de données mSQL. Plus d'informations sur mSQL à http://www.hughes.com.au/ .

6.62.2 Pré−requis

6.62.3 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'` .

6.62.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
msql.allow_persistent	"On"	PHP_INI_SYSTEM
msql.max_persistent	"−1"	PHP_INI_SYSTEM
msql.max_links	"−1"	PHP_INI_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

`msql.allow_persistent` *`boolean`*
: 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.

6.62.5 Types de ressources

6.62.6 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.

`MSQL_ASSOC` *(integer)*
`MSQL_NUM` *(integer)*
`MSQL_BOTH` *(integer)*

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- msql_affected_rows
- msql_close
- msql_connect
- msql_create_db
- msql_createdb
- msql_data_seek
- msql_dbname
- msql_drop_db
- msql_dropdb
- msql_error
- msql_fetch_array
- msql_fetch_field
- msql_fetch_object
- msql_fetch_row
- msql_field_seek
- msql_fieldflags
- msql_fieldlen
- msql_fieldname
- msql_fieldtable
- msql_fieldtype
- msql_free_result
- msql_freeresult
- msql_list_dbs
- msql_list_fields
- msql_list_tables
- msql_listdbs
- msql_listfields
- msql_listtables
- msql_num_fields
- msql_num_rows
- msql_numfields
- msql_numrows
- msql_pconnect
- msql_query
- msql_regcase
- msql_result
- msql_select_db
- msql_selectdb
- msql_tablename
- msql

6.62.8 msql_close() : Ferme une connexion mSQL.

`int` **msql_close** (`resource link_identifier`)
[lien librairie]

`msql_close` retourne `TRUE` en cas de succès, `FALSE` en cas d'erreur.

msql_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.

msql_close ne peut pas fermer les connexions persistantes, générées par msql_pconnect .

Voir aussi msql_connect et msql_pconnect .

6.62.9 msql_connect() : Ouvre une connexion mSQL.

resource **msql_connect** (*string hostname* , *string hostname:port* , *string username* , *string password*)
[lien librairie]

msql_connect retourne un identifiant de connexion positif en cas de succès, et FALSE sinon.

msql_connect établit une connexion à un serveur mSQL. Le nom d'hôte est optionnel, et lorsqu'il manque, localhost est utilisé.

Si un deuxième appel est fait à msql_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 msql_close .

Voir aussi msql_pconnect et msql_close .

6.62.10 msql_create_db() : Crée une base de données mSQL.

int **msql_create_db** (string database_name , *resource link_identifier*)
[lien librairie]

msql_create_db essaie de créer une nouvelle base de données nommée database_name sur le serveur référencé par l'identifiant link_identifier .

Voir aussi msql_drop_db .

6.62.11 msql_createdb() : Crée une base de données mSQL.

int **msql_createdb** (string database_name , *resource link_identifier*)
[lien librairie]

Identique à msql_create_db .

6.62.12 msql_data_seek() : Déplace le pointeur interne.

`bool` **msql_data_seek** (`resource query_identifier` , `int row_number`)
[lien librairie]

`msql_data_seek` retourne `TRUE` en cas de succès, et `FALSE` en cas d'échec.

`msql_data_seek` déplace le pointeur interne de résultat mSQL, et le place à l'offset donné. Le prochain appel à la fonction to `msql_fetch_row` retournera cette ligne.

Voir aussi `msql_fetch_row` .

6.62.13 msql_dbname() : Lit le nom de la base de données courante.

`string` **msql_dbname** (`resource query_identifier` , `int i`)
[lien librairie]

`msql_dbname` retourne le nom de la base de données enregistré en position `i` du pointeur de résultat retourné par la fonction `msql_listdbs` . La fonction `msql_numrows` peut être utilisée pour déterminer le nombre de bases disponibles.

6.62.14 msql_drop_db() : Efface une base de données mSQL.

`int` **msql_drop_db** (`string database_name` , `int link_identifier`)
[lien librairie]

`msql_drop_db` retourne `TRUE` en cas de succès, et `FALSE` en cas d'échec.

`msql_drop_db` essaie d'effacer une base de données entière sur le serveur référencé par l'identifiant fourni.

Voir aussi `msql_create_db` .

6.62.15 msql_dropdb() : Efface une base de données mSQL.

Voir `msql_drop_db` .

[lien librairie]

6.62.16 msql_error() : Retourne le message d'erreur

`string` **msql_error** (`void`)
[lien librairie]

Les erreurs générées par mSQL ne sont plus traitées comme des alertes. Au lieu de cela, elles sont stockées, et accessibles à partir de cette fonction.

6.62.17 msql_fetch_array() : Lit une ligne sous la forme d'un tableau.

`int` **msql_fetch_array** (`resource query_identifier` , *`int result_type`*)
[lien librairie]

`msql_fetch_array` retourne un tableau qui contient la ligne demandée, ou FALSE , si il n'y a pas d'autres lignes.

`msql_fetch_array` est une version évoluée de `msql_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 `msql_fetch_array` est une constante, et peut prendre les valeurs suivantes : MSQL_ASSOC, MSQL_NUM, et MSQL_BOTH.

Méfiez vous des requêtes qui retournent une ligne qui ne contient qu'un champs de valeur 0 (ou `NULL` , ou chaîne vide).

Il est important de noter que `msql_fetch_array` est marginalement plus lent que `msql_fetch_row` , alors qu'elle apporte un confort d'utilisation appréciable.

Voir aussi `msql_fetch_row` .

6.62.18 msql_fetch_field() : Lit la valeur d'un champs.

`object` **msql_fetch_field** (`resource query_identifier` , `int field_offset`)
[lien librairie]

`msql_fetch_field` retourne un objet contenant les informations sur un champs.

`msql_fetch_field` sert à lire les informations sur les champs, dans certaines requêtes. Si l'offset du champs n'est pas spécifié, le prochain champs 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 `msql_field_seek` .

6.62.19 msql_fetch_object() : Lit une ligne sous la forme d'un objet.

`int` **msql_fetch_object** (`resource query_identifier` , *`int result_type`*)
[lien librairie]

`msql_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.

msql_fetch_object est identique à msql_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 msql_fetch_array est une constante qui peut prendre les valeurs suivantes : MSQL_ASSOC, MSQL_NUM, et MSQL_BOTH.

msql_fetch_object est aussi rapide que msql_fetch_array , et marginalement plus lente que msql_fetch_row (la différence est non significative).

Voir aussi msql_fetch_array et msql_fetch_row .

6.62.20 msql_fetch_row() : Retourne une ligne sous la forme d'un objet.

array **msql_fetch_row** (int query_identifier)
[lien librairie]

msql_fetch_row retourne un tableau qui contient la ligne demandée, ou FALSE , si il n'y a plus de lignes à lire.

msql_fetch_row retourne une ligne, extraite du résultat associé à l'identifiant de résultat query_identifier . La ligne est retournée sous la forme d'un tableau. Chaque résultat est enregistré dans un champs, indexé numériquement, à partir de 0.

Les appels ultérieurs à msql_fetch_row retourneront les lignes suivantes, ou FALSE , lorsqu'il n'y aura plus de ligne.

Voir aussi msql_fetch_array , msql_fetch_object , msql_data_seek et msql_result .

6.62.21 msql_field_seek() : Fixe d'offset d'un champs.

int **msql_field_seek** (resource query_identifier , int field_offset)
[lien librairie]

msql_field_seek recherche l'offset du champs field_offset . Le prochain appel à msql_fetch_field sans l'argument field_offset , retournera ce champs.

Voir aussi msql_fetch_field .

6.62.22 msql_fieldflags() : Retourne le flag d'un champs.

string **msql_fieldflags** (resource query_identifier , int i)
[lien librairie]

msql_fieldflags retourne le flag du champs spécifié. Actuellement, il peut valoir soit "not NULL ", "primary key", ou une combinaison des deux ou "" (chaîne vide).

6.62.23 msql_fieldlen() : Retourne la longueur d'un champs.

`int` **msql_fieldlen** (`resource query_identifier` , `int i`)
[lien librairie]

`msql_fieldlen` retourne la longueur du champs `i` .

6.62.24 msql_fieldname() : Lit le nom d'un champs.

`string` **msql_fieldname** (`resource query_identifier` , `int field`)
[lien librairie]

`msql_fieldname` retourne le nom du champs à l'index `field` . `query_identifier` est un identifiant de résultat, et `field` est un index de champs. `msql_fieldname($result, 2);` retournera le nom du deuxième champs, dans le résultat associé à `query_identifier` .

6.62.25 msql_fieldtable() : Retourne le nom d'une table à partir d'un nom de champs.

`int` **msql_fieldtable** (`resource query_identifier` , `int field`)
[lien librairie]

`msql_fieldtable` retourne le nom de la table d'où est le champs `field` a été extrait.

6.62.26 msql_fieldtype() : Retourne le type de champs.

`string` **msql_fieldtype** (`resource query_identifier` , `int i`)
[lien librairie]

`msql_fieldtype` est similaire à `msql_fieldname` . Les arguments sont identiques, mais c'est le type du champs qui est retourné. Cela produira un résultat tel que "int", "string" ou "real".

6.62.27 msql_free_result() : Libère le résultat de la mémoire.

`int` **msql_free_result** (`resource query_identifier`)
[lien librairie]

`msql_free_result` libère de la mémoire le résultat associé à l'identifiant de résultat `query_identifier` . Lorsque PHP a terminé une requête, cette mémoire est libérée, ce qui fait que vous n'aurez pas besoin de cette fonction. Vous pouvez toujours l'utiliser pour vous assurer que vous n'utilisez pas trop de mémoire durant un script.

6.62.28 msql_freeresult() : Libère le résultat de la mémoire.

Voir `msql_free_result`

[lien librairie]

6.62.29 msql_list_dbs() : Liste les bases de données mSQL sur un serveur.

`int` **msql_list_dbs** (`void`)
[lien librairie]

`msql_list_dbs` retournea un pointeur de résultat, qui contiendra les noms des bases de données disponibles sur la connexion mSQL courante. Utilisez `msql_dbname` pour passer en revue toutes les lignes.

6.62.30 msql_list_fields() : Liste les champs d'une table.

`int` **msql_list_fields** (`string database` , `string tablename`)
[lien librairie]

`msql_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 `msql_fieldflags` , `msql_fieldlen` , `msql_fieldname` et `msql_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 `$phperrmsg` , et à moins que cette fonction n'ai été appelée avec @ (`@msql_list_fields()`), alors cette erreur sera affichée.

Voir aussi `msql_error` .

6.62.31 msql_list_tables() : Liste les tables mSQL sur une base de données

`int` **msql_list_tables** (`string database`)
[lien librairie]

`msql_list_tables` prend un nom de base de données, et fourni un résultat, un peu comme la fonction `msql` . La fonction `msql_tablename` devrait être utilisée de préférence pour extraire les nom de table d'un pointeur de résultat.

6.62.32 msql_listdbs() : Liste les bases de données mSQL sur un serveur.

Voir `msql_list_dbs` .

[lien librairie]

6.62.33 msql_listfields() : Liste les champs d'une table.

Voir `msql_list_fields` .

[lien librairie]

6.62.34 msql_listtables() : Liste les tables mSQL sur une base de données.

Voir `msql_list_tables` .

[lien librairie]

6.62.35 msql_num_fields() : Retourne le nombre de champs dans un résultat.

`int` **msql_num_fields** (`resource query_identifier`)
[lien librairie]

`msql_num_fields` retourne le nombre de champs du résultat associé à l'identifiant `query_identifier` .

Voir aussi `msql` , `msql_query` , `msql_fetch_field` et `msql_num_rows` .

6.62.36 msql_num_rows() : Retourne le nombre de lignes dans un résultat.

`int` **msql_num_rows** (`resource query_identifier`)
[lien librairie]

`msql_num_rows` retourne le nombre de lignes du résultat associé à l'identifiant `query_identifier` .

Voir aussi `msql` , `msql_query` et `msql_fetch_row` .

6.62.37 msql_numfields() : Retourne le nombre de champs dans un résultat.

`int` **msql_numfields** (`resource query_identifier`)
[lien librairie]

Identique à `msql_num_fields` .

6.62.38 msql_numrows() : Retourne le nombre de lignes dans un résultat.

`int` **msql_numrows** (`void`)
[lien librairie]

Identique à `msql_num_rows` .

6.62.39 msql_pconnect() : Ouvre une connexion persistante à un serveur mSQL.

`int` **msql_pconnect** (*`string  hostname`* , *`string  hostname:port`* , *`string  username`* , *`string  password`*)
[lien librairie]

Retourne un identifiant de connexion persistante à un serveur mSQL en cas de succès, et FALSE sinon.

`msql_pconnect` se comporte presque comme `msql_connect` mais avec deux différences majeures.

D'abord, lors de la connexion, `msql_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. (`msql_close` ne fermera pas un lien ouvert par `msql_pconnect`).

C'est pourquoi une telle connexion est considérée comme 'persistante'.

6.62.40 msql_query() : Envoie une requête mSQL

`resource` **msql_query** (`string  query` , `int  link_identifier`)
[lien librairie]

`msql_query` envoie une requête à la base de données active, sur le serveur associé à l'identifiant de connexion `link_identifier` . Si `link_identifier` 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 `msql_connect` appelé sans argument.

`msql_query` retourne un identifiant positif mSQL en cas de succès, et FALSE sinon.

Voir aussi `msql` , `msql_select_db` , et `msql_connect` .

6.62.41 msql_regcase() : Prépare une chaîne pour une recherche par expression régulière insensible à la casse.

Voir `sql_regcase` .

[lien librairie]

6.62.42 msql_result() : Retourne les données de résultat.

`int` **msql_result** (`resource  query_identifier` , `int  i` , `mixed  field`)
[lien librairie]

msql_result retourne la valeur de la cellule, à la ligne i et l'offset spécifié, field dans le résultat mSQL query_identifier .

msql_result retourne le contenu d'une cellule depuis un résultat mSQL query_identifier . L'argument de champs field peut être aussi bien un offset, qu'un nom de champs, ou encore le nom de la table point le nom du fichier (nom_table.nom_champs). 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 résutats 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 msql_result . De plus, sachez qu'accéder à un champs avec son indice numérique est beaucoup plus rapide qu'en utilisant les autres méthodes.

Alternatives recommandées : msql_fetch_row , msql_fetch_array et msql_fetch_object .

6.62.43 msql_select_db() : Sélectionne une base de données mSQL.

int **msql_select_db** (string database_name , resource link_identifier)
[lien librairie]

msql_select_db retourne TRUE en cas de succès, et FALSE en cas d'erreur.

msql_select_db choisi la base de données courante sur le serveur associé à l'identifiant de connexion link_identifier . Si link_identifier 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 msql_connect appelée sans argument.

Les prochains appels à msql_query seront fait dans la base de données active.

Voir aussi msql_connect , msql_pconnect et msql_query .

6.62.44 msql_selectdb() : Sélectionne une base de données mSQL.

Voir aussi msql_select_db .

[lien librairie]

6.62.45 msql_tablename() : Retourne le nom d'une table à partir d'un nom de champs.

string **msql_tablename** (int query_identifier , int field)
[lien librairie]

msql_tablename prend un pointeur de résultat (retourné par la fonction msql_list_tables), ainsi qu'un index, et retourne le nom d'une table. La fonction msql_numrows peut servir à déterminer le nombre de table dans le pointeur de résultat.

Exemple msql_tablename

```php
<?php
msql_connect ("localhost");
$result = msql_list_tables ("limousin");
$i = 0;
while ($i < msql_numrows ($result)) {
    $tb_names[$i] = msql_tablename ($result, $i);
    echo $tb_names[$i] . "<br>";
    $i++;
}
?>
```

6.62.46 msql() : Exécute une requête mSQL.

`resource` **msql** (`string database` , `string query` , `resource link_identifier`)
[lien librairie]

`msql` retourne un identifiant positif de résultat de requête, ou FALSE en cas d'erreur.

`msql` sélectionne la base de données `database` , et y exécute la requête `query` . Si l'identifiant de connexion `link_identifier` 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 `msql_connect` , sans argument.

6.63 MySQL

6.63.1 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/ .

6.63.2 Pré−requis

Afin de pouvoir les utiliser, vous devez compiler PHP avec le support MySQL.

6.63.3 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 librairies 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 librairies clientes installées par MySQL et évitera les conflits.

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.

6.63.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
mysql.allow_persistent	"On"	PHP_INI_SYSTEM
mysql.max_persistent	"−1"	PHP_INI_SYSTEM
mysql.max_links	"−1"	PHP_INI_SYSTEM
mysql.default_port	NULL	PHP_INI_ALL
mysql.default_socket	NULL	PHP_INI_ALL
mysql.default_host	NULL	PHP_INI_ALL
mysql.default_user	NULL	PHP_INI_ALL
mysql.default_password	NULL	PHP_INI_ALL
mysql.connect_timeout	"0"	PHP_INI_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

mysql.allow_persistent *boolean*
: 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 conecte 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 par 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 par 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 par 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.

6.63.5 Types de ressources

There are two resource types used in the MySQL module. The first one is the link identifier for a database connection, the second a resource which holds the result of a query.

6.63.6 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
MYSQL_CLIENT_COMPRESS	Utilise le protocole avec compression
MYSQL_CLIENT_IGNORE_SPACE	Autorise les espaces après les noms de fonction
MYSQL_CLIENT_INTERACTIVE	Autorise interactive_timeout secondes d'inactivité sur la connexion (au lieu de wait_timeout).

La fonction `mysql_fetch_array` utilise une constante pour spécifier les différents types de formats de réponse. Les constantes suivante 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.

6.63.7 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

```
<?php
    /* Connexion et sélection de la base */
    $link = mysql_connect("mysql_host", "mysql_user", "mysql_password")
        or die("Could not connect");
    print "Connected successfully";
    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);
?>
```


- 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

6.63.9 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*)
[lien librairie]

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.

Note
mysql_change_user a été introduite en PHP 3.0.13 et requiert MySQL 3.23.3 ou plus récent.

6.63.10 mysql_client_encoding() : Retourne le nom du jeu de caractères utilisé par le client mysql

int **mysql_client_encoding** (*resource link_identifier*)
[lien librairie]

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_identifier .

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);
?>
```

Voir aussi mysql_real_escape_string .

6.63.11 mysql_close() : Ferme la connexion MySQL.

boolean **mysql_close** (*resource link_identifier*)
[lien librairie]

mysql_close retourne TRUE en cas de succès et FALSE sinon.

mysql_close ferme la connexion au serveur MySQL associée à l'identifiant link_identifier . Si cet identifiant n'est pas spécifié, cette commande s'applique à la dernière connexion ouverte.

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
```

```
  $link = mysql_connect("kraemer", "marliesle", "secret") {
    or die("Impossible de se connecter");
  }
  print("Connexion réussie");
  mysql_close($link);
?>
```

Voir aussi `mysql_connect` et `mysql_pconnect` .

6.63.12 mysql_connect() : Ouvre une connexion à un serveur MySQL.

`int` **mysql_connect** (*`string server`* , *`string username`* , *`string password`*)
[lien librairie]

`mysql_connect` retourne un identifiant positif de connexion 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 lien sera fermé automatiquement dès que l'exécution du script sera terminée, à moins d'être fermé explicitement avec `mysql_close` .

Exemple MySQL connect

```
<?php
  $link = mysql_connect("kraemer", "marliesle", "secret") {
    or die("Connexion impossible");
  }
  print("Connexion réussie");
  mysql_close($link);
?>
```

Voir aussi `mysql_pconnect` et `mysql_close` .

6.63.13 mysql_create_db() : Crée une base de données MySQL.

`int` **mysql_create_db** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`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.

Exemple de création de base MySQL

```
<?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.

Voir aussi `mysql_drop_db` .

6.63.14 mysql_data_seek() : Déplace le pointeur interne de résultat.

`int` **mysql_data_seek** (`resource result_identifier` , `int row_number`)
[lien librairie]

`mysql_data_seek` retourne TRUE en cas de succès et FALSE sinon.

`mysql_data_seek` déplace le pointeur interne de résultat, dans le résultat associé à l'identifiant de résultat `result_identifier` . Il le fait pointer à la ligne `row_number` . Le prochain appel à `mysql_fetch_row` retournera cette ligne.

`row_number` commence à 0.

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->last_name$row->first_name}<br>\n";
  }
  mysql_free_result ($result);
?>
```

6.63.15 mysql_db_name() : Lit les noms des bases de données

`int` **mysql_db_name** (`resource result_identifier` , `int row` , *`mixed field`*)
[lien librairie]

`mysql_db_name` prend comme premier argument le pointeur de résultat `result_identifier` , 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.

6.63.16 mysql_db_query() : Envoie une requête MySQL à un serveur MySQL.

`resource` **mysql_db_query** (`string database` , `string query` , *`resource link_identifier`*)
[lien librairie]

`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` .

6.63.17 mysql_drop_db() : Efface une base de données MySQL.

`boolean` **mysql_drop_db** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`mysql_drop_db` retourne TRUE en cas de succès et FALSE sinon.

`mysql_drop_db` essaie d'effacer une base de données entière sur le serveur associé à l'identifiant de connexion `link_identifier` .

Voir aussi `mysql_create_db` .

Pour des raisons de compatibilité ascendante, mysql_drop_db() est toujours utilisable.

6.63.18 mysql_errno() : Retourne le numéro de message d'erreur de la dernière opération MySQL.

`int` **mysql_errno** (*`resource link_identifier`*)
[lien librairie]

`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_identifier` . 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.

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

Voir aussi `mysql_error` .

6.63.19 mysql_error() : Retourne le texte associé avec l'erreur générée lors de la dernière requête.

`string` **mysql_error** (*`resource link_identifier`*)
[lien librairie]

`mysql_error` retourne le dernier message d'erreur MySQL sur la connexion courante, ou sur la connexion spécifiée avec `link_identifier` .

Les erreurs générées par mySQL ne se transforment plus en alerte. A la place, elles sont accessibles via ces fonctions :

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

Voir aussi `mysql_errno` .

6.63.20 mysql_escape_string() : Protège une chaîne pour la passer à mysql_query.

`string` **mysql_escape_string** (`string unescaped_string`)
[lien librairie]

`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 _ .

6.63.21 mysql_fetch_array() : Retourne une ligne de résultat sous la forme d'un tableau associatif.

`array` **mysql_fetch_array** (`resource result_identifier` , *`int result_type`*)
[lien librairie]

`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.

```
select t1.f1 as foo t2.f1 as bar from t1, t2
```

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.

Voir aussi `mysql_fetch_row` .

mysql fetch array

```
<?php
  mysql_connect($host,$user,$password);
  $result = mysql_db_query("database","select * from table");
  while($row = mysql_fetch_array($result)) {
    echo $row["user_id"];
    echo $row["fullname"];
  }
  mysql_free_result($result);
?>
```

6.63.22 mysql_fetch_assoc() : Lit une ligne de résultats dans un tableau associatif

`array` **mysql_fetch_assoc** (`resource result_identifier`)
[lien librairie]

`mysql_fetch_assoc` retourne un tableau associatif qui contient la ligne lue, ou bien FALSE , si il ne reste plus de lignes.

`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` .

`mysql_fetch_assoc`

```
<?php
  mysql_connect($host, $user, $password);
  $result = mysql_db_query("database","select * from table");
  while ($row = mysql_fetch_assoc ($result)) {
    echo $row["user_id"];
    echo $row["fullname"];
  }
  mysql_free_result ($result);
?>
```

6.63.23 mysql_fetch_field() : Retourne les données enregistrées dans une colonne sous forme d'objet.

`object` **mysql_fetch_field** (`resource result_identifier` , *`int field_offset`*)
[lien librairie]

`mysql_fetch_field` retourne un objet contenant les données, lu dans le résultat `result_identifier` .

mysql_fetch_field sert à obtenir des informations à propos des champs, dans certaines requêtes. Si l'offset du champs n'est pas spécifié, le champs suivant le dernier champs 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.

Voir aussi mysql_field_seek .

6.63.24 mysql_fetch_lengths() : Retourne la taille de chaque colonne d'une ligne de résultat.

array **mysql_fetch_lengths** (resource result_identifier)
[lien librairie]

mysql_fetch_lengths retourne un tableau avec la taille de chaque colonne de la dernière ligne retournée par mysql_fetch_row , 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.

Voir aussi mysql_fetch_row .

6.63.25 mysql_fetch_object() : Retourne les lignes résultats sous la forme d'un objet.

object **mysql_fetch_object** (resource result_identifier , *int result_type*)
[lien librairie]

mysql_fetch_object retourne un objet dont les propriétés correspondent à une ligne d'un résultat, 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).

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)

mysql fetch object

```
<?php
  mysql_connect($host,$user,$password);
  $result = mysql_db_query("database","select * from table");
  while($row = mysql_fetch_object($result)) {
    echo $row->user_id;
    echo $row->fullname;
  }
  mysql_free_result($result);
?>
```

Voir aussi `mysql_fetch_array` et `mysql_fetch_row` .

6.63.26 mysql_fetch_row() : Retourne une ligne de résultat sous la forme d'un tableau.

`array` **mysql_fetch_row** (`resource result_identifier`)
[lien librairie]

`mysql_fetch_row` retourne un tableau énuméré qui correspond à la ligne demandée, ou FALSE s'il ne reste plus de ligne.

`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_array` , `mysql_fetch_object` , `mysql_data_seek` , `mysql_fetch_lengths` et `mysql_result` .

6.63.27 mysql_field_flags() : Retourne le sémaphore associé à la colonne spécifiée dans le résultat courant.

`string` **mysql_field_flags** (`resource result_identifier` , `int field_offset`)
[lien librairie]

`mysql_field_flags` retourne le sémaphore associé au champs 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.

6.63.28 mysql_field_len() : Retourne la longueur du champs spécifié.

`int` **mysql_field_len** (`resource result_identifier` , `int field_offset`)
[lien librairie]

`mysql_field_len` retourne la taille du champs spécifié par son offset `field_offset` .

Pour des raisons de compatibilité ascendante, mysql_fieldlen() peut encore être utilisé.

6.63.29 mysql_field_name() : Retourne le nom d'une colonne

`string` **mysql_field_name** (`resource result_identifier` , `int field_index`)
[lien librairie]

`mysql_field_name` retourne le nom d'une colonne. Les arguments de la fonction sont un identifiant de résultat `result_identifier` et l'index du champs, ie. `mysql_field_name($result,2);` .

`mysql_field_name` retournera le nom du deuxième champs dans le résultat associé à $result.

Pour des raisons de compatibilité ascendante, mysql_fieldname() peut encore être utilisé.

6.63.30 mysql_field_seek() : Déplace le pointeur de résultat

`int` **mysql_field_seek** (`resource result_identifier` , `int field_offset`)
[lien librairie]

Place le pointeur de résultat sur le champs spécifié. Lors du prochain appel à `mysql_fetch_field` qui n'aura pas d'argument d'index de champs, le champs désormais pointé sera retourné.

Voir aussi `mysql_fetch_field` .

6.63.31 mysql_field_table() : Retourne le nom de la table où se trouve une colonne

`string` **mysql_field_table** (`resource result_identifier` , `int field_offset`)
[lien librairie]

`mysql_field_table` retourne le nom de la table où se trouve une colonne. Pour des raisons de compatibilité ascendante, mysql_fieldtable() peut encore être utilisé.

6.63.32 mysql_field_type() : Retourne le type de la colonne spécifiée dans le résultat courant.

`string` **mysql_field_type** (`resource result_identifier` , `int field_offset`)
[lien librairie]

mysql_field_type est similaire à la fonction mysql_field_name . Les arguments sont identiques, mais c'est le type du champs qui est retourné. Il vaudra "int", "real", "string", "blob", ou d'autres, comme détaillé dans la documentation MySQL.

Types mysql field

```
<?php
  mysql_connect("localhost:3306");
  mysql_select_db("wisconsin");
  $result = mysql_query("SELECT * FROM onek");
  $fields = mysql_num_fields($result);
  $rows   = mysql_num_rows($result);
  $i = 0;
  $table = mysql_field_table($result, $i);
  echo "Your '".$table."' table has ".$fields." fields et ".$rows." records <br>";
  echo "The table has the following fields <br>";
  while ($i < $fields) {
    $type  = mysql_field_type($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."<br>";
    $i++;
  }
  mysql_close();
?>
```

Pour des raisons de compatibilité ascendante, mysql_fieldtype() peut encore être utilisé.

6.63.33 mysql_free_result() : Efface le résultat de la mémoire.

`int` **mysql_free_result** (`resource result_identifier`)
[lien librairie]

`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.

Pour des raisons de compatibilité ascendante, mysql_freeresult() peut encore être utilisé.

6.63.34 mysql_get_client_info() : Lit les informations sur le client MySQL

`string` **mysql_get_client_info** (`void`)
[lien librairie]

`mysql_get_client_info` retourne une chaîne qui représente le numéro de version du client utilisé par PHP.

`mysql_get_client_info` a été ajoutée en PHP 4.0.5.

6.63.35 mysql_get_host_info() : Lit les informations sur l'hôte MySQL

`string` **mysql_get_host_info** (*`resource link_identifier`*)

[lien librairie]

mysql_get_host_info retourne une chaîne qui représente le type de connexion utilisé avec la connexion link_identifier , y compris le nom du serveur hôte. Si link_identifier est omis, la dernière connexion ouverte est utilisée.

mysql_get_host_info a été ajoutée en PHP 4.0.5.

6.63.36 mysql_get_proto_info() : Lit les informations sur le protocole MySQL

string **mysql_get_proto_info** (*resource link_identifier*)
[lien librairie]

mysql_get_proto_info retourne une chaîne qui représente la version du protocole utilisé par la connexion link_identifier . Si link_identifier est omis, la dernière connexion ouverte est utilisée.

mysql_get_proto_info a été ajoutée en PHP 4.0.5.

6.63.37 mysql_get_server_info() : Lit les informations sur le serveur MySQL

string **mysql_get_server_info** (*resource link_identifier*)
[lien librairie]

mysql_get_server_info retourne une chaîne qui représente la version du serveur, dont la connexion est Si link_identifier est omis, la dernière connexion ouverte est utilisée.

mysql_get_server_info a été ajoutée en PHP 4.0.5.

6.63.38 mysql_info() : Lit des informations a propos de la dernière requête

string **mysql_info** (*resource link_identifier*)
[lien librairie]

mysql_info retourne des informations détaillées sur la dernière requête exécutée sur la dernière connexion ouverte (si omise), ou sur la connexion link_identifier . detailed information about

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 la que pour illustrer. Leur valeur s'adaptera 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 spécifiées dans la commande.

Voir aussi `mysql_affected_rows` .

6.63.39 mysql_insert_id() : Retourne l'identifiant généré par la dernière requête INSERT.

`int` **mysql_insert_id** (*`resource link_identifier`*)
[lien librairie]

`mysql_insert_id` retourne le dernier identifiant généré par un champs de type AUTO_INCREMENT, sur la connexion MySQL courante, ou bien sûr la connexion spécifiée par `link_identifier` . `mysql_insert_id` ne prend aucun argument. Elle retourne le dernier identifiant généré par la dernière fonction INSERT effectuée.

6.63.40 mysql_list_dbs() : Liste les bases de données disponibles sur le serveur MySQL.

`resource` **mysql_list_dbs** (*`resource link_identifier`*)
[lien librairie]

`mysql_list_dbs` retournera un identifiant de résultat, qui contiendra les noms des bases de données disponibles sur la connexion MySQL courante, ou bien sûr la connexion spécifiée par `link_identifier` . Utilisez la fonction `mysql_tablename` pour lire toutes les bases de données.

Pour des raisons de compatibilité ascendante, mysql_listdbs() est encore disponible.

6.63.41 mysql_list_fields() : Liste les champs du résultat MySQL.

`resource` **mysql_list_fields** (`string database_name` , `string table_name` , *`resource link_identifier`*)
[lien librairie]

`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 `$phperrmsg` et, à moins que la fonction n'ait été appelée sous la forme `@mysql()` , cette erreur sera aussi affichée.

Pour des raisons de compatibilité ascendante, mysql_listfields() est encore disponible.

6.63.42 mysql_list_processes() : Liste les processus MySQL

`resource` **mysql_list_processes** (*`resource link_identifier`*)
[lien librairie]

`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);
?>
```

Voir aussi `mysql_thread_id` .

6.63.43 mysql_list_tables() : Liste les tables d'une base de données.

`resource` **mysql_list_tables** (`string database` , *`resource link_identifier`*)
[lien librairie]

`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_identifier` . La fonction `mysql_tablename` est le meilleur moyen d'extraire les noms des tables depuis l'identifiant de résultat.

Pour des raisons de compatibilité ascendante, mysql_listtables() est encore disponible.

6.63.44 mysql_num_fields() : Retourne le nombre de champs d'un résultat.

`int` **mysql_num_fields** (`resource result_identifier`)
[lien librairie]

`mysql_num_fields` retourne le nombre de champs du résultat `result_identifier` .

Voir aussi `mysql_db_query` , `mysql_query` , `mysql_fetch_field` et `mysql_num_rows` .

Pour des raisons de compatibilité ascendante mysql_numfields() est encore disponible.

6.63.45 mysql_num_rows() : Retourne le nombre de lignes d'un résultat.

`int` **mysql_num_rows** (`resource result_identifier`)
[lien librairie]

`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` par crubel@trilizio.org

```
<?php
  $conn = mysql_connect("adresse de l'hôte", "utilisateur", "mot de passe");
  mysql_select_db("base",$conn); // nécessaire si vous avez plusieurs bases
  $Resultfornummembers = mysql_query("SELECT * FROM Accounts",$conn);
  $NumMembers = mysql_num_rows($Resultfornummembers);
  echo "$NumMembers Membres";
?>
```

Voir aussi `mysql_db_query` , `mysql_query` et `mysql_fetch_row` .

Pour des raisons de compatibilité ascendante mysql_numrows() est encore disponible.

6.63.46 mysql_pconnect() : Ouvre une connexion persistante à un serveur MySQL.

`resource` **mysql_pconnect** (*`string server`* , *`string username`* , *`string password`*)
[lien librairie]

`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 utilisér 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'.

6.63.47 mysql_ping() : Pingue le serveur, et se reconnecte au besoin

`bool` **mysql_ping** (*`resource link_identifier`*)
[lien librairie]

`mysql_ping` vérifie sir 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` .

6.63.48 mysql_query() : Envoie une requête SQL à un serveur MySQL.

`resource` **mysql_query** (`string query` , *`resource link_identifier`*)
[lien librairie]

`mysql_query` envoie une requête SQL à la base de données actuellement active sur le serveur MysQL. Si `link_identifier` 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).

`mysql_query` retourne TRUE ou FALSE , pour indiquer le succès ou l'échec de la requête. En cas de retour TRUE , la requête était valide et a pu être exécuté sur le serveur. Cela n'indique pas le nombre de lignes affectées, ou retournées. Il est parfaitement possible qu'une requête valide n'affecte aucune ligne ou ne retourne aucune ligne.

L'exemple suivant est syntaxiquement invalide, ce qui conduit `mysql_query` à l'échec et retourne FALSE :

`mysql_query`

```
<?php
  $result = mysql_query ("SELECT * WHERE 1=1")
    or die ("Requête invalide");
?>
```

L'exemple suivant est sémantiquement invalide si `my_col` n'est pas une colonne de la table `my_tbl` , ce qui conduit `mysql_query` à l'échec et retourne FALSE :

`mysql_query`

```
<?php
  $result = mysql_query ("SELECT my_col FROM my_tbl")
```

```
    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_result . Lorsque vous avez terminé avec le résultat, libérez la mémoire avec mysql_free_result .

Voir aussi mysql_affected_rows , mysql_unbuffered_query , mysql_db_query , mysql_free_result , mysql_result , mysql_select_db et mysql_connect .

6.63.49 mysql_real_escape_string() : Protège les caractères spéciaux d'une commande SQL.

string **mysql_real_escape_string** (string unescaped_string , *resource link_identifier*)
[lien librairie]

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);
?>
```

Voir aussi mysql_escape_string et mysql_character_set_name .

6.63.50 mysql_result() : Retourne un champs d'un résultat.

mixed **mysql_result** (resource result_identifier , int row , *mixed field*)
[lien librairie]

mysql_result retourne le contenu d'un champs dans le résultat MySQL result_identifier . L'argument field peut−être un offset de champs, ou le nom d'un champs, ou le nom de la table + point + le nom du champs ("table.champs"). Si la colonne a été aliasée, 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.

Alternatives à haut rendement, RECOMMANDEES : `mysql_fetch_row` , `mysql_fetch_array` et `mysql_fetch_object` .

6.63.51 mysql_select_db() : Sélectionne une base de données MySQL.

`int` **mysql_select_db** (`string database_name` , *`resource link_identifier`*)
[lien librairie]

`mysql_select_db` retourne TRUE en cas de succès, FALSE sinon.

`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.

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.

6.63.52 mysql_stat() : Retourne le statut courant du serveur

`string` **mysql_stat** (*`resource link_identifier`*)
[lien librairie]

`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);
?>
```

6.63.53 mysql_tablename() : Lit le nom de la table qui contient le champs spécifié.

`string` **mysql_tablename** (`resource result_identifier` , `int i`)
[lien librairie]

`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_num_rows` peut être utilisée pour déterminer le nombre de tables dans le pointeur de résultat.

Exemple `mysql_tablename`

```
<?php
  mysql_connect("localhost:3306");
  $result = mysql_list_tables ("wisconsin");
  $i = 0;
  while ($i < mysql_num_rows($result)) {
    $tb_names[$i] = mysql_tablename ($result, $i);
    echo $tb_names[$i] . "<br>";
    $i++;
  }
?>
```

6.63.54 mysql_thread_id() : Retourne l'identifiant du thread courant

`int` **mysql_thread_id** (*`resource link_identifier`*)
[lien librairie]

`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);
  }
?>
```

Voir aussi `mysql_ping` et `mysql_list_processes` .

6.63.55 mysql_unbuffered_query() : Exécute une requête SQL sans mobiliser les résultats

`resource` **mysql_unbuffered_query** (`string query` , *`resource link_identifier`* , *`int result_mode`*)

[lien librairie]

`mysql_unbuffered_query` envoie la requête SQL `query` au serveur MySQL identifié par `link_identifier` , 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 génère 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 `mysql_unbuffered_query` est tempéré par une limitation : `mysql_num_rows` ne fonctionne pas sur une ressource retournée par `mysql_unbuffered_query` . Vous devez aussi lire tous les résultats d'une première requête exécutée avec `mysql_unbuffered_query` , avant de pouvoir en exécuter une autre.

Voir aussi `mysql_query` .

6.64 Sessions Mohawk

6.64.1 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://www.mohawksoft.com/phoenix/ .

6.64.2 Pré−requis

6.64.3 Installation

6.64.4 Configuration à l'exécution

6.64.5 Types de ressources

6.64.6 Constantes Prédefinies

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- 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

6.64.8 msession_count() : Compte le nombre de sessions

`int` **msession_count** (`void`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.9 msession_create() : Crée une session

`bool` **msession_create** (`string session`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.10 msession_destroy() : Détruit la session

`bool` **msession_destroy** (`string name`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.11 msession_disconnect() : Ferme la connexion au serveur msession

`void` **msession_disconnect** (`void`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.12 msession_find() : Trouve une valeur

`array` **msession_find** (`string name` , `string value`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.13 msession_get_array() : Lit un tablaeu de ... (??)

`array` **msession_get_array** (`string session`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.14 msession_get() : Lit une valeur dans une session

`string` **msession_get** (`string session` , `string name` , `string value`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.15 msession_getdata() : Lit les données ... (??)

`string` **msession_getdata** (`string session`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.16 msession_inc() : Incrémente une valeur dans une session

`string` **msession_inc** (`string session` , `string name`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.17 msession_list() : Liste ... (??)

`array` **msession_list** (`void`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.18 msession_listvar() : Liste les sessions contenant une variable

`array` **msession_listvar** (`string name`)
[lien librairie]

`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.

6.64.19 msession_lock() : Verrouille une session

`int` **msession_lock** (`string name`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.20 msession_plugin() : Appele une fonction d'échappement dans les plugins de personnalité msession

`string` **msession_plugin** (`string session` , `string val` , *`string param`*)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.21 msession_randstr() : Lit une chaîne aléatoire

`string` **msession_randstr** (`int param`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.22 msession_set_array() : Modifie un tableau de ... (??)

`bool` **msession_set_array** (`string session` , `array tuples`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.23 msession_set() : Modifie une valeur dans la session

`bool` **msession_set** (`string session` , `string name` , `string value`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.24 msession_setdata() : Modifie les données (??)

`bool` **msession_setdata** (`string session` , `string value`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.25 msession_timeout() : Lit et/ou modifie la durée de vie de la session

`int` **msession_timeout** `(string session,`*`int param`*`)`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.26 msession_uniq() : Lit un identifiant unique

`string` **msession_uniq** `(int param)`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.64.27 msession_unlock() : Déverrouille une session

`int` **msession_unlock** `(string session,int key)`
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.65 Fonctions muscat

- muscat_close
- muscat_get
- muscat_give
- muscat_setup_net
- muscat_setup

6.65.2 muscat_get() : Lit la réponse de l'API muscat

`string` **muscat_get** (`resource muscat_handle`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, 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.

6.65.3 muscat_give() : Envoie une chaîne à l'API de base muscat

`int` **muscat_give** (`resource muscat_handle` , `string string`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.65.4 muscat_setup_net() : Crée une nouvelle session muscat

`resource` **muscat_setup_net** (`string muscat_host` , `int port`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, 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` .

6.65.5 muscat_setup() : Crée une nouvelle session

`resource` **muscat_setup** (`int size` , *`string muscat_dir`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, 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.

6.66 Réseau

6.66.1 Introduction

6.66.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.66.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.66.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
define_syslog_variables	"0"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, voyez `ini_set` .

Voici une courte explication sur les 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 fontionnalité est recommandé pour les performances. Durant l'exécution, vous pouvez définir ces variables en appelant `define_syslog_variables` .

6.66.5 Types de ressources

Cette extension ne définit aucune ressource.

6.66.6 Constantes Prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

Constante	Description
LOG_CONS	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.
LOG_NDELAY	Ouvre la connexion à l'historique immédiatement
LOG_ODELAY	(par défaut) retarde l'ouverture de la connexion jusqu'à ce que le premier message soit archivé
LOG_PERROR	Affiche aussi le message sur la sortie standard
LOG_PID	Ajoute le PID dans chaque message

Constante	Description
LOG_EMERG	Le système est inutilisable
LOG_ALERT	Une mesure doit être prise immédiatement
LOG_CRIT	Conditions critiques
LOG_ERR	Erreurs
LOG_WARNING	Alertes
LOG_NOTICE	Notes (normales mais significantives)
LOG_INFO	Message d'informations
LOG_DEBUG	Message de débogage

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- checkdnsrr
- closelog
- debugger_off
- debugger_on
- define_syslog_variables
- fsockopen
- gethostbyaddr
- gethostbyname
- gethostbynamel
- getmxrr
- getprotobyname
- getprotobynumber
- getservbyname
- getservbyport
- ip2long
- long2ip
- openlog
- pfsockopen
- socket_get_status
- socket_set_blocking
- socket_set_timeout
- syslog

6.66.8 closelog() : Ferme la connexion à l'historique système.

`int` **closelog** (`void`)
[lien librairie]

`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` .

6.66.9 debugger_off() : Inactive le debugger interne de PHP.

`int` **debugger_off** (`void`)
[lien librairie]

`debugger_off` inactive le debuggeur interne de PHP. Le debuggeur est toujours en cours de développement.

6.66.10 debugger_on() : Active le debugger interne de PHP.

`int` **debugger_on** (`string address`)
[lien librairie]

`debugger_on` active le debugger interne de PHP, et le connecte à l'adresse `address` . Le debuggeur est toujours en cours de développement.

6.66.11 define_syslog_variables() : Initialise toutes les constantes liées au syslog

`void` **define_syslog_variables** (`void`)
[lien librairie]

`define_syslog_variables` initialise toutes les constantes utilisées par les fonctions de syslog.

Voir aussi `openlog` , `syslog` et `closelog` .

6.66.12 fsockopen() : Ouvre une socket de connexion Internet ou Unix.

`int` **fsockopen** (*`string udp://hostname`* , `int port` , *`int errno`* , *`string errstr`* , *`float timeout`*)
[lien librairie]

`fsockopen` crée un flot 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 `hostname` sur le port `port` . Pour les connexions UDP, vous devez explicitement spécifier le protocole : `udp://hostname` . Via un domaine Unix, `hostname` 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.

`fsockopen` 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, `fsockopen` retourne FALSE , et si les options `errno` et `errstr` ont été fournies, elles contiennent désormais les raisons de l'échec. Si l'erreur retournée est 0 et que la fonction retourne FALSE , c'est une indication d'erreur. C'est probablement du 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 `fsockopen`

```
<?php
$fp = fsockopen("www.php.net", 80, &$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
$fp = fsockopen("udp://127.0.0.1", 13, &$errno, &$errstr);
if (!$fp) {
    echo "ERREUR: $errno − $errstr<br>\n";
} else {
    fwrite($fp,"\n");
    echo fread($fp, 26);
    fclose($fp);
}
?>
```

Note

Le paramètre `timeout` a été introduit en PHP 3.0.9 et le support UDP en PHP 4.

Voir aussi `pfsockopen` , `socket_set_blocking` , `socket_set_timeout` , `fgets` , `fgetss` , `fputs` , `fclose` et `feof` .

6.66.13 gethostbyaddr() : Retourne le nom d'hôte correspondant à une IP.

`string` **gethostbyaddr** (`string ip_address`)
[lien librairie]

`gethostbyaddr` retourne le nom d'hôte correspondant à l'IP `ip_address` . Si une erreur survient, retourne `ip_address` .

Voir aussi `gethostbyname` .

6.66.14 gethostbyname() : Retourne l'adresse IP correspondant à un hôte.

`string` **gethostbyname** (`string hostname`)
[lien librairie]

`gethostbyname` retourne l'adresse IP correspondant à l'hôte `hostname` .

Voir aussi `gethostbyaddr` .

6.66.15 gethostbynamel() : Retourne la liste d'IP correspondante à un hôte.

`array` **gethostbynamel** (`string hostname`)
[lien librairie]

`gethostbynamel` retourne la liste d'IP correspondant à l'hôte `hostname` .

Voir aussi `gethostbyname` , `gethostbyaddr` , `checkdnsrr` , `getmxrr` et la page 8 du manuel.

6.66.16 getmxrr() : Retourne les enregistrements MX d'un hôte.

`int` **getmxrr** (`string hostname` , `array mxhosts` , *`array weight`*)
[lien librairie]

`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.

Voir aussi `checkdnsrr` , `gethostbyname` , `gethostbynamel` , `gethostbyaddr` , et la page 8 du manuel.

6.66.17 getprotobyname() : Retourne le numéro de protocole associé à un nom de protocole

`int` **getprotobyname** (`string name`)
[lien librairie]

`getprotobyname` retourne le numéro de protocole associé avec le nom de protocole `name` , comme dans `/etc/protocols` .

Voir aussi `getprotobynumber` .

6.66.18 getprotobynumber() : Retourne le nom de protocole associé à un numéro de protocole

`string` **getprotobynumber** (`int number`)
[lien librairie]

`getprotobynumber` retourne le nom de protocole associé avec le numéro de protocole `name` , comme dans `/etc/protocols` .

Voir aussi `getprotobyname` .

6.66.19 getservbyname() : Retourne le numéro de port associé à un service Internet et un protocole.

`int` **getservbyname** (`string service` , `string protocol`)
[lien librairie]

`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` .

6.66.20 getservbyport() : Retourne le service Internet qui correspond au port et protocole.

`string` **getservbyport** (`int port` , `string protocol`)
[lien librairie]

`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` .

6.66.21 ip2long() : Convertit une chaîne contenant une adresse (IPv4) IP numérique en adresse littérale.

`int` **ip2long** (`string ip_address`)
[lien librairie]

`ip2long` génère une adresse IPv4 à partir de son équivalent numérique.

Exemple `ip2long`

```
<?php
$ip = gethostbyname("www.php.net");
$out = "Les URLS suivantes sont équivalentes :<br>\n";
$out .= "http://www.php.net/, http://".$ip."/, et http://".ip2long($ip)."/<br>\n";
echo $out;
?>
```

Ce second exemple montre comment afficher une adresse convertie à l'aide de la fonction `printf` :

Affichage d'adresse IP

```
<?php
```

```
    $ip = gethostbyname("www.php.net");
    printf ("%u\n", ip2long ($ip));
    echo $out;
?>
```

Voir aussi `long2ip`

6.66.22 long2ip() : Convertit une adresse IP (IPv4) en adresse IP numérique

`string` **long2ip** (`int proper_address`)
[lien librairie]

`long2ip` génère une adresse IP (format aaa.bbb.ccc.ddd) à partir de sa représentation littérale.

Voir aussi `ip2long`

6.66.23 openlog() : Ouvre la connexion à l'historique système.

`int` **openlog** (`string ident` , `int option` , `int facility`)
[lien librairie]

`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.

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	réservé pour utilisation ultérieure
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` .

6.66.24 pfsockopen() : Ouvre une socket de connexion Internet ou Unix persistante.

`int` **pfsockopen** (`string hostname` , `int port` , *`int errno`* , *`string errstr`* , *`int timeout`*)
[lien librairie]

`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` .

6.66.25 socket_get_status() : Retourne les informations sur une socket

`array` **socket_get_status** (`resource socket_get_status`)
[lien librairie]

`socket_get_status` retourne les informations sur la socket `socket_get_status` , et fournit la réponse sous la forme d'un tableau à quatre entrées:

- `timed_out` (boolean) − La socket a expirée en attendant des données
- `blocked` (boolean) − La socket a été bloquée
- `eof` (boolean) − Indique un événement fin de fichier (EOF)
- `unread_bytes` (int) − Nombre d'octets restant dans les buffers de la socket.

Voir aussi `accept_connect` , `bind` , `connect` , `listen` et `strerror` .

6.66.26 socket_set_blocking() : Active/désactive le mode bloquant d'une socket.

`int` **socket_set_blocking** (`int socket_descriptor` , `int mode`)
[lien librairie]

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 immediatement toujours `TRUE` tandis qu'en mode bloquant, elle va attendre que des données arrivent pour répondre `TRUE` .

6.66.27 socket_set_timeout() : Fixe la durée de vie de la socket

`boolean` **socket_set_timeout** (`int socket_descriptor` , `int seconds` , `int microseconds`)
[lien librairie]

`socket_set_timeout` fixe la durée de vie de la socket `socket_descriptor` , exprimée comme la somme de `seconds` secondes et `microseconds` micro−secondes.

Exemple `socket_set_timeout`

```
<?php
$fp = fsockopen("http://www.php.net", 80);
if(!$fp) {
    echo "Unable to open\n";
} else {
    fputs($fp,"GET / HTTP/1.0\n\n");
    $start = time();
    socket_set_timeout($fp, 2);
    $res = fread($fp, 2000);
    var_dump(socket_get_status($fp));
    fclose($fp);
    print $res;
}
?>
```

Cette fonction s'appelait set_socket_timeout() mais elle est désormais obsolète.

Voir aussi `fsockopen` et `fopen` .

6.66.28 syslog() : Génère un message dans l'historique système.

`int` **syslog** (`int priority` , `string message`)
[lien librairie]

`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 débuggage

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("myScriptLog", LOG_PID | LOG_PERROR, LOG_LOCAL0);
// quelques lignes de code
if (authorized_client()) {
    // faire quelquechose
} else {
    // client non autorisé!
    // notation de la tentative
    $access = date("Y/m/d H:i:s");
    syslog(LOG_WARNING,"Client non autorisé: $access $REMOTE_ADDR ($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.

6.67 Ncurses terminal screen control functions

6.67.1 Introduction

ncurses (new curses) is a free software emulation of curses in System V Rel 4.0 (and above). It uses terminfo format, supports pads, colors, multiple highlights, form characters and function key mapping.

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 utiliser ce module à vos risques et périls.

Ncurses is available for the following platforms:

- AIX
- BeOS
- Cygwin
- Digital Unix (aka OSF1)
- FreeBSD
- GNU/Linux
- HPUX
- IRIX
- OS/2
- SCO OpenServer
- Solaris
- SunOS

6.67.2 Pré−requis

You need the ncurses libraries and headerfiles. Download the latest version from the ftp://ftp.gnu.org/pub/gnu/ncurses/ or from an other GNU−Mirror.

6.67.3 Installation

To get these functions to work, you have to compile the CGI version of PHP with `−−with−ncurses` .

6.67.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Name	Default	Changeable
ncurses.value	"42"	PHP_INI_ALL
ncurses.string	"foobar"	PHP_INI_ALL

For further details and definition of the PHP_INI_* constants see `ini_set` .

6.67.5 Types de ressources

Cette extension ne définit aucune ressource.

6.67.6 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.

6.67.6.1 Error codes

On error ncurses functions return NCURSES_ERR.

6.67.6.2 Colors

constant	meaning
NCURSES_COLOR_BLACK	no color (black)
NCURSES_COLOR_WHITE	white
NCURSES_COLOR_RED	red − supported when terminal is in color mode
NCURSES_COLOR_GREEN	green − supported when terminal is in color mod
NCURSES_COLOR_YELLOW	yellow − supported when terminal is in color mod
NCURSES_COLOR_BLUE	blue − supported when terminal is in color mod
NCURSES_COLOR_CYAN	cyan − supported when terminal is in color mod
NCURSES_COLOR_MAGENTA	magenta − supported when terminal is in color mod

6.67.6.3 Keys

constant	meaning
NCURSES_KEY_F0 − NCURSES_KEY_F64	function keys F1 − F64
NCURSES_KEY_DOWN	down arrow
NCURSES_KEY_UP	up arrow
NCURSES_KEY_LEFT	left arrow
NCURSES_KEY_RIGHT	right arrow
NCURSES_KEY_HOME	home key (upward+left arrow)
NCURSES_KEY_BACKSPACE	backspace
NCURSES_KEY_DL	delete line
NCURSES_KEY_IL	insert line
NCURSES_KEY_DC	delete character
NCURSES_KEY_IC	insert char or enter insert mode
NCURSES_KEY_EIC	exit insert char mode
NCURSES_KEY_CLEAR	clear screen
NCURSES_KEY_EOS	clear to end of screen
NCURSES_KEY_EOL	clear to end of line
NCURSES_KEY_SF	scroll one line forward

NCURSES_KEY_SR	scroll one line backward
NCURSES_KEY_NPAGE	next page
NCURSES_KEY_PPAGE	previous page
NCURSES_KEY_STAB	set tab
NCURSES_KEY_CTAB	clear tab
NCURSES_KEY_CATAB	clear all tabs
NCURSES_KEY_SRESET	soft (partial) reset
NCURSES_KEY_RESET	reset or hard reset
NCURSES_KEY_PRINT	print
NCURSES_KEY_LL	lower left
NCURSES_KEY_A1	upper left of keypad
NCURSES_KEY_A3	upper right of keypad
NCURSES_KEY_B2	center of keypad
NCURSES_KEY_C1	lower left of keypad
NCURSES_KEY_C3	lower right of keypad
NCURSES_KEY_BTAB	back tab
NCURSES_KEY_BEG	beginning
NCURSES_KEY_CANCEL	cancel
NCURSES_KEY_CLOSE	close
NCURSES_KEY_COMMAND	cmd (command)
NCURSES_KEY_COPY	copy
NCURSES_KEY_CREATE	create
NCURSES_KEY_END	end
NCURSES_KEY_EXIT	exit
NCURSES_KEY_FIND	find
NCURSES_KEY_HELP	help
NCURSES_KEY_MARK	mark
NCURSES_KEY_MESSAGE	message
NCURSES_KEY_MOVE	move
NCURSES_KEY_NEXT	next
NCURSES_KEY_OPEN	open
NCURSES_KEY_OPTIONS	options
NCURSES_KEY_PREVIOUS	previous
NCURSES_KEY_REDO	redo
NCURSES_KEY_REFERENCE	ref (reference)
NCURSES_KEY_REFRESH	refresh
NCURSES_KEY_REPLACE	replace
NCURSES_KEY_RESTART	restart
NCURSES_KEY_RESUME	resume
NCURSES_KEY_SAVE	save
NCURSES_KEY_SBEG	shiftet beg (beginning)
NCURSES_KEY_SCANCEL	shifted cancel
NCURSES_KEY_SCOMMAND	shifted command
NCURSES_KEY_SCOPY	shifted copy
NCURSES_KEY_SCREATE	shifted create

NCURSES_KEY_SDC	shifted delete char
NCURSES_KEY_SDL	shifted delete line
NCURSES_KEY_SELECT	select
NCURSES_KEY_SEND	shifted end
NCURSES_KEY_SEOL	shifted end of line
NCURSES_KEY_SEXIT	shifted exit
NCURSES_KEY_SFIND	shifted find
NCURSES_KEY_SHELP	shifted help
NCURSES_KEY_SHOME	shifted home
NCURSES_KEY_SIC	shifted input
NCURSES_KEY_SLEFT	shifted left arrow
NCURSES_KEY_SMESSAGE	shifted message
NCURSES_KEY_SMOVE	shifted move
NCURSES_KEY_SNEXT	shifted next
NCURSES_KEY_SOPTIONS	shifted options
NCURSES_KEY_SPREVIOUS	shifted previous
NCURSES_KEY_SPRINT	shifted print
NCURSES_KEY_SREDO	shifted redo
NCURSES_KEY_SREPLACE	shifted replace
NCURSES_KEY_SRIGHT	shifted right arrow
NCURSES_KEY_SRSUME	shifted resume
NCURSES_KEY_SSAVE	shifted save
NCURSES_KEY_SSUSPEND	shifted suspend
NCURSES_KEY_UNDO	undo
NCURSES_KEY_MOUSE	mouse event has occured
NCURSES_KEY_MAX	maximum key value

Mouse

Constant	meaning
NCURSES_BUTTON1_RELEASED − NCURSES_BUTTON4_RELEASED	button (1−4) released
NCURSES_BUTTON1_PRESSED − NCURSES_BUTTON4_PRESSED	button (1−4) pressed
NCURSES_BUTTON1_CLICKED − NCURSES_BUTTON4_CLICKED	button (1−4) clicked
NCURSES_BUTTON1_DOUBLE_CLICKED − NCURSES_BUTTON4_DOUBLE_CLICKED	button (1−4) double clicked
NCURSES_BUTTON1_TRIPLE_CLICKED − NCURSES_BUTTON4_TRIPLE_CLICKED	button (1−4) triple clicked
NCURSES_BUTTON_CTRL	ctrl pressed during click
NCURSES_BUTTON_SHIFT	shift pressed during click
NCURSES_BUTTON_ALT	alt pressed during click
NCURSES_ALL_MOUSE_EVENTS	report all mouse events
NCURSES_REPORT_MOUSE_POSITION	report mouse position

- 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_can_change_color
- ncurses_cbreak
- ncurses_clear
- ncurses_clrtobot
- ncurses_clrtoeol
- ncurses_color_set
- ncurses_curs_set
- ncurses_def_prog_mode
- ncurses_def_shell_mode
- ncurses_define_key
- 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_getmouse
- ncurses_halfdelay
- ncurses_has_colors
- ncurses_has_ic
- ncurses_has_il
- ncurses_has_key
- 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_killchar
- ncurses_longname
- ncurses_mouseinterval
- ncurses_mousemask
- 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_newwin
- ncurses_nl
- ncurses_nocbreak
- ncurses_noecho
- ncurses_nonl
- ncurses_noqiflush
- ncurses_noraw
- ncurses_putp
- ncurses_qiflush
- ncurses_raw
- ncurses_refresh
- ncurses_resetty
- ncurses_savetty
- ncurses_scr_dump
- ncurses_scr_init
- ncurses_scr_restore
- ncurses_scr_set
- ncurses_scrl
- 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_touch
- ncurses_standend
- ncurses_standout
- ncurses_start_color
- ncurses_termattrs
- ncurses_termname
- ncurses_timeout
- ncurses_typeahead
- ncurses_ungetch
- ncurses_ungetmouse
- ncurses_use_default_colors
- ncurses_use_env
- ncurses_use_extended_names
- ncurses_vidattr
- ncurses_vline
- ncurses_wrefresh

6.67.8 ncurses_addchnstr() : Add attributed string with specified length at current position

`int` **ncurses_addchnstr** `( string s , int n )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.9 ncurses_addchstr() : Add attributed string at current position

`int` **ncurses_addchstr** `( string s )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.10 ncurses_addnstr() : Add string with specified length at current position

`int` **ncurses_addnstr** `( string s , int n )`
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.11 ncurses_addstr() : Output text at current position

`int` **ncurses_addstr** (`string text`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.12 ncurses_assume_default_colors() : Define default colors for color 0

`int` **ncurses_assume_default_colors** (`int fg` , `int bg`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.13 ncurses_attroff() : Turn off the given attributes

`int` **ncurses_attroff** (`int attributes`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.14 ncurses_attron() : Turn on the given attributes

`int` **ncurses_attron** (`int attributes`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.15 ncurses_attrset() : Set given attributes

`int` **ncurses_attrset** (`int attributes`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.16 ncurses_baudrate() : Returns baudrate of terminal

`int` **ncurses_baudrate** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.17 ncurses_beep() : Let the terminal beep

`int` **ncurses_beep** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` sends an audlible alert (bell) and if its not possible flashes the screen. Returns `FALSE` on success, otherwise `TRUE` .

See also: `ncurses_flash`

6.67.18 ncurses_bkgd() : Set background property for terminal screen

`int` **ncurses_bkgd** (`int attrchar`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.19 ncurses_bkgdset() : Control screen background

`void` **ncurses_bkgdset** (`int attrchar`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.20 ncurses_border() : Draw a border around the screen using attributed characters

`int` **ncurses_border** (`int left` , `int right` , `int top` , `int bottom` , `int tl_corner` , `int tr_corner` , `int bl_corner` , `int br_corner`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.21 ncurses_can_change_color() : Check if we can change terminals colors

`bool` **ncurses_can_change_color** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

The function ncurses_can_change_color returns TRUE or FALSE , depending on whether the terminal has color capabilities and whether the programmer can change the colors.

6.67.22 ncurses_cbreak() : Switch of input buffering

bool **ncurses_cbreak** (void)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 disables line buffering and character processing (interrupt and flow control characters are unaffected), making characters typed by the user immediately available to the program.

ncurses_cbreak returns TRUE or NCURSES_ERR if any error occured.

See also: ncurses_nocbreak

6.67.23 ncurses_clear() : Clear screen

bool **ncurses_clear** (void)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 clears the screen completely without setting blanks. Returns FALSE on success, otherwise TRUE .

Note: ncurses_clear clears the screen without setting blanks, which have the current background rendition. To clear screen with blanks, use ncurses_erase .

See also: ncurses_erase

6.67.24 ncurses_clrtobot() : Clear screen from current position to bottom

bool **ncurses_clrtobot** (void)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_clrtobot` erases all lines from cursor to end of screen and creates blanks. Blanks created by `ncurses_clrtobot` have the current background rendition. Returns `TRUE` if any error occured, otherwise `FALSE` .

See also: `ncurses_clear` , `ncurses_clrtoeol`

6.67.25 ncurses_clrtoeol() : Clear screen from current position to end of line

`bool` **ncurses_clrtoeol** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` erases the current line from cursor position to the end. Blanks created by `ncurses_clrtoeol` have the current background rendition. Returns `TRUE` if any error occured, otherwise `FALSE` .

See also: `ncurses_clear` , `ncurses_clrtobot`

6.67.26 ncurses_color_set() : Set fore− and background color

`int` **ncurses_color_set** (`int pair`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.27 ncurses_curs_set() : Set cursor state

`int` **ncurses_curs_set** (`int visibility`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.28 ncurses_def_prog_mode() : Saves terminals (program) mode

`bool` **ncurses_def_prog_mode** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` saves the current terminal modes for program (in curses) for use by `ncurses_reset_prog_mode` . Returns FALSE on success, otherwise TRUE .

See also: `ncurses_reset_prog_mode`

6.67.29 ncurses_def_shell_mode() : Saves terminals (shell) mode

`bool` **ncurses_def_shell_mode** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` saves the current terminal modes for shell (not in curses) for use by `ncurses_reset_shell_mode` . Returns FALSE on success, otherwise TRUE .

See also: `ncurses_reset_shell_mode`

6.67.30 ncurses_define_key() : Define a keycode

`int` **ncurses_define_key** (`string definition` , `int keycode`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.31 ncurses_delay_output() : Delay output on terminal using padding characters

`int` **ncurses_delay_output** (`int milliseconds`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.32 ncurses_delch() : Delete character at current position, move rest of line left

`bool` **ncurses_delch** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` deletes the character under the cursor. All characters to the right of the cursor on the same line are moved to the left one position and the last character on the line is filled with a blank. The cursor position does not change. Returns FALSE on success, otherwise TRUE .

See also: `ncurses_deleteln`

6.67.33 ncurses_deleteln() : Delete line at current position, move rest of screen up

`bool` **ncurses_deleteln** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` deletes the current line under cursorposition. All lines below the current line are moved up one line. The bottom line of window is cleared. Cursor position does not change. Returns FALSE on success, otherwise TRUE .

See also: `ncurses_delch`

6.67.34 ncurses_delwin() : Delete a ncurses window

`int` **ncurses_delwin** (`resource window`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.35 ncurses_doupdate() : Write all prepared refreshes to terminal

`bool` **ncurses_doupdate** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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()` compares the virtual screen to the physical screen and updates the physical screen. This way is more effective than using multiple refresh calls. Returns FALSE on success, TRUE if any error occured.

6.67.36 ncurses_echo() : Activate keyboard input echo

`bool` **ncurses_echo** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` enables echo mode. All characters typed by user are echoed by `ncurses_getch` . Returns FALSE on success, TRUE if any error occured.

To disable echo mode use `ncurses_noecho` .

6.67.37 ncurses_echochar() : Single character output including refresh

`int` **ncurses_echochar** (`int character`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.38 ncurses_end() : Stop using ncurses, clean up the screen

`int` **ncurses_end** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.39 ncurses_erase() : Erase terminal screen

`bool` **ncurses_erase** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` fills the terminal screen with blanks. Created blanks have the current background rendition, set by `ncurses_bkgd` . Returns FALSE on success, TRUE if any error occured.

See also: `ncurses_bkgd` , `ncurses_clear`

6.67.40 ncurses_erasechar() : Returns current erase character

`string` **ncurses_erasechar** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` returns the current erase char character.

See also: `ncurses_killchar`

6.67.41 ncurses_filter() :

`int` **ncurses_filter** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.42 ncurses_flash() : Flash terminal screen (visual bell)

`bool` **ncurses_flash** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` flashes the screen, and if its not possible, sends an audible alert (bell). Returns `FALSE` on success, otherwise `TRUE` .

See also: `ncurses_beep`

6.67.43 ncurses_flushinp() : Flush keyboard input buffer

`bool` **ncurses_flushinp** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

The `ncurses_flushinp` throws away any typeahead that has been typed and has not yet been read by your program. Returns `FALSE` on success, otherwise `TRUE` .

6.67.44 ncurses_getch() : Read a character from keyboard

`int` **ncurses_getch** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.45 ncurses_getmouse() : Reads mouse event

`bool` **ncurses_getmouse** (`array mevent`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 reads mouse event out of queue. Function ncurses_getmouse will return ; FALSE if a mouse event is actually visible in the given window, otherwise it will return TRUE . Event options will be delivered in parameter mevent , which has to be an array, passed by reference (see example below). On success an associative array with following keys will be delivered:

- "id" : Id to distinguish multiple devices
- "x" : screen relative x−position in character cells
- "y" : screen relative y−position in character cells
- "z" : currently not supported
- "mmask" : Mouse action

ncurses_getmouse example

```
switch (ncurses_getch){
  case NCURSES_KEY_MOUSE:
    if (!ncurses_getmouse(&$mevent)){
      if ($mevent["mmask"] & NCURSES_MOUSE_BUTTON1_PRESSED){
        $mouse_x = $mevent["x"]; // Save mouse position
        $mouse_y = $mevent["y"];
      }
    }
  break;

  default:
    ....
}
```

See also: ncurses_ungetmouse

6.67.46 ncurses_halfdelay() : Put terminal into halfdelay mode

int **ncurses_halfdelay** (int tenth)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.47 ncurses_has_colors() : Check if terminal has colors

bool **ncurses_has_colors** (void)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` returns TRUE or FALSE depending on whether the terminal has color capacitites.

See also: `ncurses_can_change_color`

6.67.48 ncurses_has_ic() : Check for insert− and delete−capabilities

`bool` **ncurses_has_ic** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` checks terminals insert− and delete capabilitites. It returns TRUE when terminal has insert/delete−capabilities, otherwise FALSE .

See also: `ncurses_has_il`

6.67.49 ncurses_has_il() : Check for line insert− and delete−capabilities

`bool` **ncurses_has_il** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` checks terminals insert− and delete−line−capabilities. It returns TRUE when terminal has insert/delete−line capabilities, otherwise FALSE

See also: `ncurses_has_ic`

6.67.50 ncurses_has_key() : Check for presence of a function key on terminal keyboard

`int` **ncurses_has_key** (`int keycode`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.51 ncurses_hline() : Draw a horizontal line at current position using an attributed character and max. n characters long

`int` **ncurses_hline** (`int charattr` , `int n`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.52 ncurses_inch() : Get character and attribute at current position

`string` **ncurses_inch** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` returns the character from the current position.

6.67.53 ncurses_init_color() : Set new RGB value for color

`int` **ncurses_init_color** (`int color` , `int r` , `int g` , `int b`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.54 ncurses_init_pair() : Allocate a color pair

`int` **ncurses_init_pair** (`int pair` , `int fg` , `int bg`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.55 ncurses_init() : Initialize ncurses

`int` **ncurses_init** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.56 ncurses_insch() : Insert character moving rest of line including character at current position

`int` **ncurses_insch** (`int character`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.57 ncurses_insdelln() : Insert lines before current line scrolling down (negative numbers delete and scroll up)

`int` **ncurses_insdelln** (`int count`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.58 ncurses_insertln() : Insert a line, move rest of screen down

`bool` **ncurses_insertln** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 inserts a new line above the current line. The bottom line will be lost.

6.67.59 ncurses_insstr() : Insert string at current position, moving rest of line right

int **ncurses_insstr** (string text)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.60 ncurses_instr() : Reads string from terminal screen

int **ncurses_instr** (string buffer)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 returns the number of charaters read from the current character position until end of line. buffer contains the characters. Atrributes are stripped from the characters.

6.67.61 ncurses_isendwin() : Ncurses is in endwin mode, normal screen output may be performed

bool **ncurses_isendwin** (void)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 returns TRUE , if ncurses_endwin has been called without any subsequent calls to ncurses_wrefresh , otherwise FALSE .

See also: ncurses_endwin ncurses_wrefresh()

6.67.62 ncurses_keyok() : Enable or disable a keycode

int **ncurses_keyok** (int keycode , bool enable)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.63 ncurses_killchar() : Returns current line kill character

`bool` **ncurses_killchar** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` returns the current line kill character.

See also: `ncurses_erasechar`

6.67.64 ncurses_longname() : Returns terminals description

`string` **ncurses_longname** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` returns a verbose description of the terminal. The descritpion is truncated to 128 characters. On Error `ncurses_longname` returns NULL.

See also: `ncurses_termname`

6.67.65 ncurses_mouseinterval() : Set timeout for mouse button clicks

`int` **ncurses_mouseinterval** (`int milliseconds`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.66 ncurses_mousemask() : Sets mouse options

`int` **ncurses_mousemask** (`int newmask` , `int oldmask`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Function `ncurses_mousemask` will set mouse events to be reported. By default no mouse events will be reported. The function `ncurses_mousemask` will return a mask to indicated which of the in parameter `newmask` specified mouse events can be reported. On complete failure, it returns 0. In parameter `oldmask` , which is passed by reference `ncurses_mousemask` returns the previous value of mouse event mask. Mouse events are represented bei NCURSES_KEY_MOUSE in the `ncurses_wgetch` input stream. To read the event data and pop the event of of queue, call `ncurses_getmouse` .

As a side effect, setting a zero mousemask in `newmask` turns off the mouse pointer. Setting a non zero value turns mouse pointer on.

mouse mask options can be set with the following predefined constants:

- 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

See also: `ncurses_getmouse` , `ncurses_ungetmouse` `ncurese_getch`

`ncurses_mousemask` example

```
$newmask = NCURSES_BUTTON1_CLICKED + NCURSES_BUTTON1_RELEASED;
$mask = ncurses_mousemask($newmask, &$oldmask);
if ($mask & $newmask){
  printf ("All specified mouse options will be supported\n");
}
```

6.67.67 ncurses_move() : Move output position

`int` **ncurses_move** `( int y , int x )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.68 ncurses_mvaddch() : Move current position and add character

`int` **ncurses_mvaddch** `( int y , int x , int c )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.69 ncurses_mvaddchnstr() : Move position and add attrributed string with specified length

`int` **ncurses_mvaddchnstr** `( int y , int x , string s , int n )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.70 ncurses_mvaddchstr() : Move position and add attributed string

`int` **ncurses_mvaddchstr** `( int y , int x , string s )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.71 ncurses_mvaddnstr() : Move position and add string with specified length

`int` **ncurses_mvaddnstr** `( int y , int x , string s , int n )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.72 ncurses_mvaddstr() : Move position and add string

`int` **ncurses_mvaddstr** `( int y , int x , string s )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.73 ncurses_mvcur() : Move cursor immediately

`int` **ncurses_mvcur** `( int old_y , int old_x , int new_y , int new_x )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.74 ncurses_mvdelch() : Move position and delete character, shift rest of line left

`int` **ncurses_mvdelch** (`int y` , `int x`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.75 ncurses_mvgetch() : Move position and get character at new position

`int` **ncurses_mvgetch** (`int y` , `int x`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.76 ncurses_mvhline() : Set new position and draw a horizontal line using an attributed character and max. n characters long

`int` **ncurses_mvhline** (`int y` , `int x` , `int attrchar` , `int n`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.77 ncurses_mvinch() : Move position and get attributed character at new position

`int` **ncurses_mvinch** (`int y` , `int x`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.78 ncurses_mvvline() : Set new position and draw a vertical line using an attributed character and max. n characters long

`int` **ncurses_mvvline** (`int y` , `int x` , `int attrchar` , `int n`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.79 ncurses_mvwaddstr() : Add string at new position in window

`int` **ncurses_mvwaddstr** (`resource window` , `int y` , `int x` , `string text`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.80 ncurses_napms() : Sleep

`int` **ncurses_napms** (`int milliseconds`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.81 ncurses_newwin() : Create a new window

`int` **ncurses_newwin** (`int rows` , `int cols` , `int y` , `int x`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.82 ncurses_nl() : Translate newline and carriage return / line feed

`bool` **ncurses_nl** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.83 ncurses_nocbreak() : Switch terminal to cooked mode

`bool` **ncurses_nocbreak** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` routine returns terminal to normal (cooked) mode. Initially the terminal may or may not in cbreak mode as the mode is inherited. Therefore a program should call `ncurses_cbreak` and `ncurses_nocbreak` explicitly. Returns TRUE if any error occured, otherwise FALSE .

See also: `ncurses_cbreak`

6.67.84 ncurses_noecho() : Switch off keyboard input echo

`bool` **ncurses_noecho** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` prevents echoing of user typed characters. Returns TRUE if any error occured, otherwise FALSE .

See also: `ncurses_echo` , `ncurses_getch`

6.67.85 ncurses_nonl() : Do not translate newline and carriage return / line feed

`bool` **ncurses_nonl** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.86 ncurses_noqiflush() : Do not flush on signal characters

`int` **ncurses_noqiflush** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.87 ncurses_noraw() : Switch terminal out of raw mode

`bool` **ncurses_noraw** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` switches the terminal out of raw mode. Raw mode is similar to cbreak mode, in that characters typed are immediately passed through to the user program. The differences that are that in raw mode, the interrupt, quit, suspend and flow control characters are all passed through uninterpreted, instead of generating a signal. Returns `TRUE` if any error occured, otherwise `FALSE` .

See also: `ncurses_raw` , `ncurses_cbreak` , `ncurses_nocbreak`

6.67.88 ncurses_putp() :

`int` **ncurses_putp** (`string text`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.89 ncurses_qiflush() : Flush on signal characters

`int` **ncurses_qiflush** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.90 ncurses_raw() : Switch terminal into raw mode

`bool` **ncurses_raw** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` places the terminal in raw mode. Raw mode is similar to cbreak mode, in that characters typed are immediately passed through to the user program. The differences that are that in raw mode, the interrupt, quit, suspend and flow control characters are all passed through uninterpreted, instead of generating a signal. Returns TRUE if any error occured, otherwise FALSE .

See also: `ncurses_noraw` , `ncurses_cbreak` , `ncurses_nocbreak`

6.67.91 ncurses_refresh() : Refresh screen

`int` **ncurses_refresh** (`int ch`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.92 ncurses_resetty() : Restores saved terminal state

`bool` **ncurses_resetty** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Function `ncurses_resetty` restores the terminal state, which was previously saved by calling `ncurses_savetty` . This function always returns FALSE .

See also: `ncurses_savetty`

6.67.93 ncurses_savetty() : Saves terminal state

`bool` **ncurses_savetty** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Function `ncurses_savetty` saves the current terminal state. The saved terminal state can be restored with function `ncurses_resetty` . `ncurses_savetty` always returns FALSE .

See also: `ncurses_resetty`

6.67.94 ncurses_scr_dump() : Dump screen content to file

`int` **ncurses_scr_dump** (`string filename`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.95 ncurses_scr_init() : Initialize screen from file dump

`int` **ncurses_scr_init** (`string filename`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.96 ncurses_scr_restore() : Restore screen from file dump

`int` **ncurses_scr_restore** (`string filename`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.97 ncurses_scr_set() : Inherit screen from file dump

`int` **ncurses_scr_set** (`string filename`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.98 ncurses_scrl() : Scroll window content up or down without changing current position

`int` **ncurses_scrl** (`int count`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.99 ncurses_slk_attr() : Returns current soft label key attribute

`bool` **ncurses_slk_attr** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` returns the current soft label key attribute. On error returns `TRUE` , otherwise `FALSE` .

6.67.100 ncurses_slk_attroff() :

`int` **ncurses_slk_attroff** (`int intarg`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.101 ncurses_slk_attron() :

`int` **ncurses_slk_attron** (`int intarg`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.102 ncurses_slk_attrset() :

`int` **ncurses_slk_attrset** (`int intarg`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.103 ncurses_slk_clear() : Clears soft labels from screen

`bool` **ncurses_slk_clear** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

The function `ncurses_slk_clear` clears soft label keys from screen. Returns `TRUE` on error, otherwise `FALSE` .

6.67.104 ncurses_slk_color() : Sets color for soft label keys

`int` **ncurses_slk_color** (`int intarg`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.105 ncurses_slk_init() : Initializes soft label key functions

`bool` **ncurses_slk_init** (`int format`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Funtion `ncurses_slk_init` must be called before `ncurses_initscr` or `ncurses_newterm` is called. If `ncurses_initscr` eventually uses a line from stdscr to emulate the soft labels, then `format` determines how the labels are arranged of the screen. Setting `format` to 0 indicates a 3−2−3 arrangement of the labels, 1 indicates a 4−4 arrangement and 2 indicates the PC like 4−4−4 mode, but in addition an index line will be created.

6.67.106 ncurses_slk_noutrefresh() : Copies soft label keys to virtual screen

`bool` **ncurses_slk_noutrefresh** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.107 ncurses_slk_refresh() : Copies soft label keys to screen

`bool` **ncurses_slk_refresh** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` copies soft label keys from virtual screen to physical screen. Returns `TRUE` on error, otherwise `FALSE` .

6.67.108 ncurses_slk_restore() : Restores soft label keys

`bool` **ncurses_slk_restore** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

The function `ncurses_slk_restore` restores the soft label keys after `ncurses_slk_clear` has been performed.

6.67.109 ncurses_slk_touch() : Fources output when ncurses_slk_noutrefresh is performed

`bool` **ncurses_slk_touch** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

The `ncurses_slk_touch` function forces all the soft labels to be output the next time a `ncurses_slk_noutrefresh` is performed.

6.67.110 ncurses_standend() : Stop using 'standout' attribute

`int` **ncurses_standend** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.111 ncurses_standout() : Start using 'standout' attribute

`int` **ncurses_standout** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.112 ncurses_start_color() : Start using colors

`int` **ncurses_start_color** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.113 ncurses_termattrs() : Returns a logical OR of all attribute flags supported by terminal

`bool` **ncurses_termattrs** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.114 ncurses_termname() : Returns terminals (short)−name

`string` **ncurses_termname** (`void`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` returns terminals shortname. The shortname is truncated to 14 characters. On error `ncurses_termname` returns NULL.

See also: `ncurses_longname`

6.67.115 ncurses_timeout() : Set timeout for special key sequences

`void` **ncurses_timeout** (`int millisec`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.116 ncurses_typeahead() : Specify different filedescriptor for typeahead checking

`int` **ncurses_typeahead** (`int fd`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.117 ncurses_ungetch() : Put a character back into the input stream

`int` **ncurses_ungetch** (`int keycode`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.118 ncurses_ungetmouse() : Pushes mouse event to queue

`bool` **ncurses_ungetmouse** (`array mevent`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` pushes a KEY_MOUSE event onto the unput queue and associates with this event the given state sata and screen−relative character cell coordinates, specified in `mevent` . Event options will be specified in associative array `mevent` :

- "id" : Id to distinguish multiple devices
- "x" : screen relative x−position in character cells
- "y" : screen relative y−position in character cells
- "z" : currently not supported
- "mmask" : Mouse action

`ncurses_ungetmouse` returns FALSE on success, otherwise `TRUE` .

See also: `ncurses_getmouse`

6.67.119 ncurses_use_default_colors() : Assign terminal default colors to color id −1

`bool` **ncurses_use_default_colors** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.120 ncurses_use_env() : Control use of environment information about terminal size

`void` **ncurses_use_env** (`bool flag`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.121 ncurses_use_extended_names() : Control use of extended names in terminfo descriptions

`int` **ncurses_use_extended_names** `( bool flag )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.122 ncurses_vidattr() :

`int` **ncurses_vidattr** `( int intarg )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.123 ncurses_vline() : Draw a vertical line at current position using an attributed character and max. n characters long

`int` **ncurses_vline** `( int charattr , int n )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.67.124 ncurses_wrefresh() : Refresh window on terminal screen

`int` **ncurses_wrefresh** `( resource window )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

undocumented

6.68 Lotus Notes

6.68.1 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 utiliser 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

6.68.3 notes_copy_db() : Crée une note à partir du formulaire form form_name

`string` **notes_copy_db** (`string from_database_name` , `string to_database_name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.4 notes_create_db() : Crée une base Lotus Notes

`bool` **notes_create_db** (`string database_name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.5 notes_create_note() : Crée une note à partir du formulaire form_name

`string` **notes_create_note** (`string database_name` , `string form_name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.6 notes_drop_db() : Supprime une base Lotus Notes

`bool` **notes_drop_db** (`string database_name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.7 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`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.8 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`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.9 notes_list_msgs() : Retourne les notes rangées dans database_name

`bool` **notes_list_msgs** (`string db`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.10 notes_mark_read() : Marque la note note_id comme lu par l'utilisateur user_name

`string` **notes_mark_read** (`string database_name` , `string user_name` , `string note_id`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.11 notes_mark_unread() : Marque la note note_id comme non plus pour l'utilisateur user_name

`string` **notes_mark_unread** (`string database_name` , `string user_name` , `string note_id`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.12 notes_nav_create() : Crée un nom de navigateur dans database_name

`bool` **notes_nav_create** (`string database_name` , `string name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.13 notes_search() : Cherche les notes par mots clés dans database_name

`string` **notes_search** (`string database_name` , `string keywords`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.14 notes_unread() : Retourne l'identifiant de la note non lue pour l'utilisateur courant user_name

`string` **notes_unread** (`string database_name` , `string user_name`)
[lien librairie]

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.68.15 notes_version() : Lit la version de Lotus Notes

`string` **notes_version** (`string database_name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.69 ODBC unifié

6.69.1 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 .

Reportez−vous à Installation sous Unix pour plus de détails sur les configurations de ces serveurs.

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 à OpenLink Software . Plus d'informations sur iODBC, ainsi qu'un HOWTO (en anglais), est disponible à www.iodbc.org .

6.69.2 Pré−requis

Pour accéder à l'une des bases de données supportées, il faut que les librairies équivalentes soient installées.

6.69.3 Installation

Lisez le chapitre Options d'installation de bases de données pour plus d'informations sur la configuration de ces bases.

6.69.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
odbc.default_db *	NULL	PHP_INI_ALL
odbc.default_user *	NULL	PHP_INI_ALL
odbc.default_pw *	NULL	PHP_INI_ALL
odbc.allow_persistent	"1"	PHP_INI_SYSTEM
odbc.check_persistent	"1"	PHP_INI_SYSTEM
odbc.max_persistent	"−1"	PHP_INI_SYSTEM
odbc.max_links	"−1"	PHP_INI_SYSTEM
odbc.defaultlrl	"4096"	PHP_INI_ALL
odbc.defaultbinmode	"1"	PHP_INI_ALL

Note
Les entrées marquées avec une étoile ne sont pas encore implémentées.

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les 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 spécifiée dans les fonctions `odbc_connect` ou `odbc_pconnect` .

`odbc.default_pw` *`chaîne de caractères`*
: Mot de passe à utiliser si aucun n'est spécifiée 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 connecxon 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 connexion ODBC par processus, y compris les connexion 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.

6.69.5 Types de ressources

Cette extension ne définit aucune ressource.

6.69.6 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 Prédefinies
- odbc_autocommit
- odbc_binmode
- odbc_close_all
- odbc_close
- odbc_columnprivileges
- odbc_columns
- odbc_commit

- odbc_connect
- odbc_cursor
- odbc_do
- odbc_error
- odbc_errormsg
- odbc_exec
- odbc_execute
- odbc_fetch_into
- 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_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

6.69.8 odbc_binmode() : Modifie la gestion des colonnes de données binaires.

`int` **odbc_binmode** (`resource result_id` , `int mode`)
[lien librairie]

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 `4096` est 4096 et les valeurs par défaut de odbc_binmode est `ODBC_BINMODE_RETURN` . La gestion des colonnes binaires est aussi modifiée par `odbc_longreadlen` .

6.69.9 odbc_close_all() : Ferme toutes les connexions ODBC

`void` **odbc_close_all** (`void`)
[lien librairie]

`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.

6.69.10 odbc_close() : Ferme une connexion ODBC.

`void` **odbc_close** (`resource connection_id`)
[lien librairie]

`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.

6.69.11 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`*)
[lien librairie]

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).

6.69.12 odbc_columns() : Liste les colonnes d'une table

int **odbc_columns** (resource connection_id , *string qualifier* , *string owner* , *string table_name* , *string column_name*)
[lien librairie]

odbc_columns liste toutes les colonnes de la source dedonnées. 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.

6.69.13 odbc_commit() : Valide une transaction ODBC

`int` **odbc_commit** (`resource connection_id`)
[lien librairie]

`odbc_commit` retourne TRUE en cas de succès, FALSE en cas d'erreur. Toutes les connexions en cours sur `connection_id` sont validées.

6.69.14 odbc_connect() : Connexion à une source

`resource` **odbc_connect** (`string dsn` , `string user` , `string password` , *`int cursor_type`*)
[lien librairie]

`odbc_connect` retourne un identifiant de connexion ODBC ou 0 (FALSE) en cas d'erreur.

L'identifiant de connexion retournée 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 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 type 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` .

6.69.15 odbc_cursor() : Lecture du pointeur de fiche courante (cursorname).

`string` **odbc_cursor** (`resource result_id`)
[lien librairie]

`odbc_cursor` lit le pointeur de fiche courante (cursorname) pour le résultat `result_id` .

6.69.16 odbc_do() : Synonyme de odbc_exec

`string` **odbc_do** (`resource connection_id` , `string query`)
[lien librairie]

odbc_do exécute la requête query avec la connexion connection_id .

6.69.17 odbc_error() : Lit le dernier code d'erreur

string **odbc_error** (*resource connection_id*)
[lien librairie]

odbc_error retourne un état ODVBC 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 .

6.69.18 odbc_errormsg() : Lit le dernier message d'erreur

string **odbc_errormsg** (*int connection_id*)
[lien librairie]

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 .

6.69.19 odbc_exec() : Prépare et exécute une requête SQL.

int **odbc_exec** (resource connection_id , string query_string)
[lien librairie]

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 éxecutions multiples de requêtes SQL.

6.69.20 odbc_execute() : Exécute une requête SQL préparée.

int **odbc_execute** (resource result_id , *array parameters_array*)
[lien librairie]

odbc_execute exécute une requête SQL préparée par odbc_prepare . odbc_execute retourne TRUE en cas d'exécution réussie, et FALSE sinon. Le tableau de paramètres parameters_array ne sert que si vous avez besoin de paramètrer votre requête.

6.69.21 odbc_fetch_into() : Lit une ligne de résultat, et la place dans un tableau.

`int` **odbc_fetch_into** (`resource result_id` , *`int rownumber`* , `array result_array`)
[lien librairie]

`odbc_fetch_into` retourne le nombre de colonnes dans le résultat, ou FALSE en cas d'erreur. `result_array` doit avoir été passé par référence, mais il peut être de n'importe quel type, étant donné qu'il sera converti en tableau. Le tableau contiendra les valeurs des colonnes, ces dernières étant numérotées à partir de 0.

Exemple avec `odbc_fetch_into` (avant PHP 4.0.6)

```
<?php
  $rc = odbc_fetch_into($res_id, $my_array);
?>
```

ou

```
<?php
  $rc = odbc_fetch_into($res_id, $row, $my_array);
  $rc = odbc_fetch_into($res_id, 1, $my_array);
?>
```

Jusqu'en PHP 4.0.5, le paramètre `result_array` n'a plus besoin d'être passé par référence.

Depuis PHP 4.0.6, le paramètre `rownumber` ne peut pas être passé comme une constante, mais comme une variable.

Exemple avec `odbc_fetch_into` (après PHP 4.0.6)

```
<?php
  $rc = odbc_fetch_into($res_id, $my_array);
?>
```

ou

```
<?php
  $row = 1;
  $rc = odbc_fetch_into($res_id, $row, $my_array);
?>
```

Evolution ultérieure : en PHP 4.1, `odbc_fetch_into` aura le format suivant : `int` **odbc_fetch_into** (`int result_id` , `array result_array` , *`int rownumber`*) Notez que le paramètre `rownumber` sera optionnel, tandis que `result_array` ne l'est pas.

6.69.22 odbc_fetch_row() : Lit une ligne de résultat.

`int` **odbc_fetch_row** (`resource result_id` , *`int row_number`*)
[lien librairie]

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.

6.69.23 odbc_field_len() : Lit la longueur d'un champs.

int **odbc_field_len** (resource result_id , int field_number)
[lien librairie]

odbc_field_len retourne la longueur du champs référence par le nombre field_number , dans la connexion ODBC result_id . Les numéros de champs commencent à 1.

6.69.24 odbc_field_name() : Lit le nom de la colonne.

string **odbc_field_name** (resource result_id , int field_number)
[lien librairie]

odbc_field_name lit le nom de la colonne dont l'index est field_number . La numérotation des champs commence à 1. FALSE est retourné en cas d'erreur.

6.69.25 odbc_field_num() : Numéro de colonne

int **odbc_field_num** (resource result_id , string field_name)
[lien librairie]

odbc_field_num retourne le numéro de la colonne nommée field_name . Ce numéro correspond à l'index du champs dans le résultat ODBC. La numérotation commence à 1. FALSE est retourné en cas d'erreur.

6.69.26 odbc_field_precision() : Alias de odbc_field_len

string **odbc_field_precision** (resource result_id , int field_number)
[lien librairie]

odbc_field_precision retourne la précision du champs référéncé 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.

6.69.27 odbc_field_scale() : Lit l'échelle d'un champs

string **odbc_field_scale** (resource result_id , int field_number)
[lien librairie]

odbc_field_precision retourne l'échelle du champs référéncé par son numéro de champs field_number dans le résultat ODBC result_id .

6.69.28 odbc_field_type() : Type de données d'un champs.

string **odbc_field_type** (resource result_id , int field_number)
[lien librairie]

odbc_field_type retourne le type de données SQL d'un champs, identifié par son index. La numérotation des champs commence à 1.

6.69.29 odbc_foreignkeys() : Liste les clés étrangères

int **odbc_foreignkeys** (resource connection_id , string pk_qualifier , string pk_owner , string pk_table , string fk_qualifier , string fk_owner , string fk_table)
[lien librairie]

odbc_foreignkeys liste les clés étrangèes 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.

6.69.30 odbc_free_result() : Libère les ressources associées à un résultat

int **odbc_free_result** (resource result_id)
[lien librairie]

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 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.

6.69.31 odbc_gettypeinfo() : Liste les types de données supportés par une source

int **odbc_gettypeinfo** (resource connection_id , *int data_type*)
[lien librairie]

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.

6.69.32 odbc_longreadlen() : Gestion des colonnes de type LONG.

`int` **odbc_longreadlen** (`resource result_id` , `int length`)
[lien librairie]

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 chaîne vide.

Note
La gestion des types LONGVARBINARY est aussi affectée par `odbc_binmode` .

6.69.33 odbc_num_fields() : Nombre de colonnes dans un résultat

`int` **odbc_num_fields** (`resource result_id`)
[lien librairie]

`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` .

6.69.34 odbc_num_rows() : Nombre de ligne dans un résultat.

`int` **odbc_num_rows** (`odbc_prepare result_id`)
[lien librairie]

`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, ce PEUT le nombre de lignes disponibles, mais ce n'est pas certain.

Note: `odbc_num_rows` après un SELECT retournera −1 avec de nombreux pilotes.

6.69.35 odbc_pconnect() : Ouvre une connexion persistante à une source de données.

`resource` **odbc_pconnect** (`string dsn` , `string user` , `string password` , *`int cursor_type`*)
[lien librairie]

`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.

6.69.36 odbc_prepare() : Prépare une commande pour l'exécution

`resource` **odbc_prepare** (`resource connection_id` , `string query_string`)
[lien librairie]

`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` .

6.69.37 odbc_primarykeys() : Liste les colonnes utilisées dans une clé primaire

`int` **odbc_primarykeys** (`resource connection_id` , `string qualifier` , `string owner` , `string table`)
[lien librairie]

`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

6.69.38 odbc_procedurecolumns() : Liste les paramètres des procédures

`int` **odbc_procedurecolumns** (`resource connection_id` , *`string qualifier`* , *`string owner`* , *`string proc`* , *`string column`*)
[lien librairie]

`odbc_procedurecolumns` list les paramètres d'entrée et de sortie, ainsi que les colonnes utilisées dans les procédures désignées par les paramètres. `odbc_procedurecolumns` etourne 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).

6.69.39 odbc_procedures() : Liste les procédure stockées

`int` **odbc_procedures** (`resource connection_id` , *`string qualifier`* , *`string owner`* , *`string name`*)
[lien librairie]

`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).

6.69.40 odbc_result_all() : Affiche le résultat sous la forme d'une table HTML.

`int` **odbc_result_all** (`odbc_prepare result_id` , *`string format`*)
[lien librairie]

`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.

6.69.41 odbc_result() : Lit les données de résultat.

`string` **odbc_result** (`odbc_prepare result_id` , `mixed field`)
[lien librairie]

`odbc_result` retourne le contenu d'un champs.

`field` peut être aussi bien un entier, contenant le numéro de colonne du champs, dans le résultat, ou bien une chaîne de caractère, qui représente le nom du champs. Par exemple:

```
<?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 champs de la ligne courante, du résultat `result_id` . Le deuxième appel à `odbc_result` retourne la valeur du troisième champs 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 champs passé ne correspond à aucun champs 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` .

6.69.42 odbc_rollback() : Annule une transaction

`int` **odbc_rollback** (`odbc_prepare connection_id`)
[lien librairie]

`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.

6.69.43 odbc_setoption() : Modifie les paramètres ODBC.

`int` **odbc_setoption** (`resource id` , `int function` , `int option` , `int param`)
[lien librairie]

`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 problème 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 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($conn,
<TT>TRUE</TT>
);
odbc_setoption($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);
?>
```

6.69.44 odbc_specialcolumns() : Retourne l'ensemble optimal de colonnes, qui permettent de définir uniquement une ligne dans une table

`int` **odbc_specialcolumns** (`resource connection_id` , `int type` , `string qualifier` , `string owner` , `string table` , `int scope` , `int nullable`)
[lien librairie]

Lorsque le `type` est SQL_BEST_ROWID, `odbc_specialcolumns` retourne la ou les colonnes qui permettent de repérer uniquement chaque ligne d'une table.

Lorsque le type `type` est SQL_ROWVER, `odbc_specialcolumns` retourne l'ensemble optimal de colonne 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.

6.69.45 odbc_statistics() : Calcule des statistiques sur une table

`int` **odbc_statistics** (`resource connection_id` , `string qualifier` , `string owner` , `string table_name` , `int unique` , `int accuracy`)
[lien librairie]

`odbc_statistics` effectue quelques statistiques sur une tables 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.

6.69.46 odbc_tableprivileges() : Liste les tables et leurs privilèges

`int` **odbc_tableprivileges** (`resource connection_id` , *`string qualifier`* , *`string owner`* , *`string name`*)
[lien librairie]

`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).

6.69.47 odbc_tables() : Liste les tables d'une source.

`int` **odbc_tables** (`odbc_setoption connection_id` , *`string qualifier`* , *`string owner`* , *`string name`* , *`string types`*)
[lien librairie]

`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 qualifiers 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 representent 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.

6.70 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 utiliser ce module à vos risques et périls.

6.70.1.1 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 strategie habile pour mettre en place des objets complexes, et des hierarchies d'objets. Ce système peut fonctionne 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 classe

```
<?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 = $_dt−>now();
       // autre code ...
   }

   // autres méthodes ...
}

$rep = new Report();
$dt = new DateTime();
$dtp = new DateTimePlus("l, F j, Y (h:i:s a, T)");

// Générating du rapport avec une simple date
$rep−>setDateTime(&$dt);
echo $rep−>generateReport();

// plus loin dans le code

// generation du rapport avec une date designée
$rep−>setDateTime(&$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 aggréger 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 aggrégée 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.

6.70.1.2 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 instantion quelques objets issue de ces classes, et nous réalisons des aggrégations et désaggrégations, 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 aggrégations

echo "\nAggrégeons \$fs avec la classe WDDXStorage\n";
aggregate($fs, "WDDXStorage");
```

```
echo "L'objet \$fs \n";
echo object_info($fs)."\n";

echo "\nAggrégeons 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 aggrégations 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'aggrégation 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.

```
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 aggrégeons alors `$fs` avec la classe `WDDXStorage` , et nous affichons les informations. Nous pouvons aussi vior que même si l'objet `$fs` est toujours du type `FileStorage` , il a maintenant la propriété `$version` , et la méthode `store` , qui sont définies dans `WDDXStorage` . Une chose importante à noter est que les éléments privés n'ont pas été aggrégés, même si ils sont présents dans l'objet `$ws` . Un autre absent est le constructeur de `WDDXStorage` , qu'il n'est pas logique d'aggréger.

```
Aggrégeons \$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'aggrégation est cumulatif, ce qui fait que lorsque nous aggrégeons `$fs` avec la classe `DBStorage` , nous générons un objet qui peut utiliser n'importe laquelle des méthodes de stockage de ces classes.

```
Aggrégeons 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 aggrégés les méthodes et propriétés dynamiquement, nous pouvons aussi les désaggrégéer. Si vous désaggrégeons la classe `WDDXStorage` de l'objet `$fs` , nous allons obtenir :

```
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 mentionnéé 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.

- aggregate_methods_by_list
- aggregate_methods_by_regexp
- aggregate_methods
- aggregate_properties_by_list
- aggregate_properties_by_regexp
- aggregate_properties
- aggregate
- aggregation_info
- deaggregate

6.70.3 aggregate_methods_by_regexp() : Aggrège sélectivement les méthodes d'une classe grâce à une expression régulière

`void` **aggregate_methods_by_regexp** (`object object` , `string class_name` , `string regexp` , *`boolean exclude`*)
[lien librairie]

`aggregate_methods_by_regexp` aggrège les méthodes de la classe `class_name` avec l'objet `object` , en filtrant les méthodes aggrégées grâce à l'expression régulière `regexp` , 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_regexp` et `deaggregate`

6.70.4 aggregate_methods() : Aggrège dynamiquement les méthodes d'une classe à un objet

`void` **aggregate_methods** (`object object` , `string class_name`)
[lien librairie]

`aggregate_methods` aggège 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 commes privées, et donc ignorées.

Voir aussi `aggregate` , `aggregate_info` , `aggregate_methods_by_list` , `aggregate_methods_by_regexp` , `aggregate_properties` , `aggregate_properties_by_list` , `aggregate_properties_by_regexp` et `deaggregate`

6.70.5 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` , *`boolean exclude`*)
[lien librairie]

`aggregate_properties_by_list` aggrège les propriétés de la classe `class_name` avec l'objet `object` , en filtrant les propriétés aggrégées grâce à la liste `methods_list` , appliquée au nom de la propriété. Le paramètre optionnel `exclude` sert à décider si l'expression régulière sélectionne les propriétés (`exclude` vaut FALSE sa valeur par défaut), ou bien si elle sert à identifier les propriétés qui doivent être exclues (`exclude` vaut TRUE).

Les propriétés dont le nom commence par un caractère souligné ("_"), qui sont considérées comme privées, sont toujours exclues.

Voir aussi aggregate , aggregate_methods , aggregate_methods_by_list , aggregate_methods_by_regexp , aggregate_properties , aggregate_properties_by_regexp , aggregate_info , deaggregate

6.70.6 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 , *boolean exclude*)
[lien librairie]

aggregate_properties_by_regexp aggrège les propriétés de la classe class_name avec l'objet object , en filtrant les propriétés aggrégées 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

6.70.7 aggregate_properties() : Aggrège dynamiquement les propriétés d'une classe à un objet

void **aggregate_properties** (object object , string class_name)
[lien librairie]

aggregate_properties aggège 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 commes 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

6.70.8 aggregate() : Aggrège dynamiquemetn des classes et objets

void **aggregate** (object object , string class_name)
[lien librairie]

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

6.70.9 aggregation_info() : Retourne un tableau associatif décrivant les classes aggrégées

`array` **aggregation_info** (`object object`)
[lien librairie]

`aggregation_info` retourne les informations d'aggrégation pour l'objet `object` , sous la forme d'un tableau associatif. Les clés de ce tableau sont les noms des classes aggrégées.

Par exemple, le code ci−dessous

Exemple avec `aggregation_info`

```
<?php

class Slicer {
    var $vegetable;

    function Slicer($vegetable) {
        $this->vegetable = $vegetable;
    }

    function slice_it($num_cuts) {
        echo "Tranchage\n";
        for ($i=0; $i < $num_cuts; $i++) {
            // un peu de tranchage some slicing
        }
    }
}

class Dicer {
    var $vegetable;
    var $rotation_angle = 90;   // degrés

    function Dicer($vegetable) {
        $this->vegetable = $vegetable;
    }

    function dice_it($num_cuts) {
        echo "Coupe suivant une direction\n";
        for ($i=0; $i < $num_cuts; $i++) {
            // tranchage
        }
        $this->rotate($this->rotation_angle);
        echo "Coupe suivant une autre direction\n";
        for ($i=0; $i < $num_cuts; $i++) {
            // tranchage encoer
        }
    }

    function rotate($deg) {
        echo "Rotation de {$this->vegetable} {$deg} degrés\n";
    }

    function _secret_super_dicing($num_cuts) {
        // secret industriel, a ne pas montrer.
    }
}

$obj = new Slicer('onion');
aggregate($obj, 'Dicer');
print_r(aggregate_info($obj));
```

```
?>
```

va produire le résultat suivant

```
Array
(
    [dicer] => Array
        (
            [methods] => Array
                (
                    [0] => dice_it
                    [1] => rotate
                )

            [properties] => Array
                (
                    [0] => rotation_angle
                )

        )

)
```

Comme vous pouvez le voir, toutes les méthodes et propriétés de `Dicer` ont été aggrégées dans notre nouvel objet, a l'excéption de la classe constructuer, et de la méthode **_secret_super_dicing**

Voir aussi `aggregate` , `aggregate_methods` , `aggregate_methods_by_list` , `aggregate_methods_by_regexp` , `aggregate_properties` , `aggregate_properties_by_list` , `aggregate_properties_by_regexp` et `deaggregate`

6.70.10 deaggregate() : Desaggrège un objet

`void` **object_aggregation** (`object object` , *`string class_name`*)
[lien librairie]

`object_aggregation` supprime les méthodes et propriétés de la classe `class_name` , qui ont été aggrégées dans l'objet `object` . Si le paramètre optionnel `class_name` est passé, seules les méthodes et propriétés de cette classe sont désaggrégées. Sinon, toutes les méthodes et propriétés aggrégées 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`

6.71 Oracle 8

6.71.1 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 librairies 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.

6.71.2 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

```
# ldd /www/apache/bin/httpd
    libpthread.so.0 => /lib/libpthread.so.0 (0x4001c000)
    libm.so.6 => /lib/libm.so.6 (0x4002f000)
    libcrypt.so.1 => /lib/libcrypt.so.1 (0x4004c000)
    libdl.so.2 => /lib/libdl.so.2 (0x4007a000)
    libc.so.6 => /lib/libc.so.6 (0x4007e000)
    /lib/ld−linux.so.2 => /lib/ld−linux.so.2 (0x40000000)
```

Si la libpthread n'est pas listée, vous devez réinstaller Apache :

Reinstaller Apache avec pthread

```
# cd /usr/src/apache_1.3.xx
# make clean
# LIBS=−lpthread ./config.status
# make
```

```
# make install
```

Notez bien que sur certains systèmes comme UnixWare il faut utiliser libthread au lieu de libpthread. PHP et Apache doivent alors être configurés avec EXTRA_LIBS=−lthread.

6.71.3 Installation

Vous devez compiler PHP avec l'option `−−with−oci8[=DIR]` , où DIR vaut par défaut la variable d'environnement ORACLE_HOME.

6.71.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.71.5 Types de ressources

6.71.6 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.

`OCI_DEFAULT` (nombre ˆ virgule flottante)
`OCI_DESCRIBE_ONLY` (nombre ˆ virgule flottante)
`OCI_COMMIT_ON_SUCCESS` (nombre ˆ virgule flottante)
`OCI_EXACT_FETCH` (nombre ˆ virgule flottante)
`SQLT_BFILEE` (nombre ˆ virgule flottante)
`SQLT_CFILEE` (nombre ˆ virgule flottante)
`SQLT_CLOB` (nombre ˆ virgule flottante)
`SQLT_BLOB` (nombre ˆ virgule flottante)
`SQLT_RDD` (nombre ˆ virgule flottante)
`OCI_B_SQLT_NTY` (nombre ˆ virgule flottante)
`OCI_SYSDATE` (nombre ˆ virgule flottante)
`OCI_B_BFILE` (nombre ˆ virgule flottante)
`OCI_B_CFILEE` (nombre ˆ virgule flottante)
`OCI_B_CLOB` (nombre ˆ virgule flottante)
`OCI_B_BLOB` (nombre ˆ virgule flottante)
`OCI_B_ROWID` (nombre ˆ virgule flottante)
`OCI_B_CURSOR` (nombre ˆ virgule flottante)
`OCI_B_BIN` (nombre ˆ virgule flottante)
`OCI_FETCHSTATEMENT_BY_COLUMN` (nombre ˆ virgule flottante)
`OCI_FETCHSTATEMENT_BY_ROW` (nombre ˆ virgule flottante)
`OCI_ASSOC` (nombre ˆ virgule flottante)
`OCI_NUM` (nombre ˆ virgule flottante)
`OCI_BOTH` (nombre ˆ virgule flottante)
`OCI_RETURN_NULLS` (nombre ˆ virgule flottante)
`OCI_RETURN_LOBS` (nombre ˆ virgule flottante)
`OCI_DTYPE_FILE` (nombre ˆ virgule flottante)
`OCI_DTYPE_LOB` (nombre ˆ virgule flottante)

OCI_DTYPE_ROWID (nombre ˆ virgule flottante)
OCI_D_FILE (nombre ˆ virgule flottante)
OCI_D_LOB (nombre ˆ virgule flottante)
OCI_D_ROWID (nombre ˆ virgule flottante)

6.71.7 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
$sth = 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
- 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
- ociResult
- ociRollback
- ociRowCount
- OCISaveLob
- OCISaveLobFile
- ociServerVersion
- ocisetprefetch
- ociStatementType
- OCIWriteLobToFile

6.71.9 OCICancel() : Libère les ressouces

`bool` **ocicancel** (`resource stmt`)
[lien librairie]

`ocicancel` détruit les ressources liées au dernier résultat `stmt` . Si vous ne souhaitez plus lire d'informations dans ce résultat, utilisez cette fonction.

6.71.10 OCICollAppend() : Non documenté

`bool` **OCICollAppend** (`object collection` , `object object`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.71.11 OCICollAssign() : Bientôt documenté....

`bool` **ocicollassign** (`object collection` , `object object`)
[lien librairie]

Bientôt documenté....

6.71.12 OCICollAssignElem() : Bientôt documenté....

`bool` **ocicollassignelem** (`object collection` , `string ndx` , `string val`)
[lien librairie]

Bientôt documenté....

6.71.13 OCICollGetElem() : Bientôt documenté....

`string` **ocicollgetelem** (`object collection` , `string ndx`)
[lien librairie]

Bientôt documenté....

6.71.14 OCICollMax() : Bientôt documenté....

`bool` **ocicollmax** (`object collection`)
[lien librairie]

Bientôt documenté....

6.71.15 OCICollSize() : Bientôt documenté....

`bool` **ocicollsize** (`object collection`)
[lien librairie]

Bientôt documenté....

6.71.16 OCICollTrim() : Bientôt documenté....

`bool` **ocicolltrim** (`object collection` , `int num`)
[lien librairie]

Bientôt documenté....

6.71.17 ociColumnIsNULL() : Teste si la valeur d'une colonne est NULL.

`bool` **ocicolumnisnull** (`resource stmt` , `mixed column`)
[lien librairie]

`ocicolumnisnull` retourne TRUE si la colonne `col` 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 `col` .

6.71.18 ociColumnName() : Retourne le nom d'une colonne.

`string` **ocicolumnname** (`resource stmt` , `int col`)
[lien librairie]

`ocicolumnname` retourne le nom de la colonne numéro `col` (en commençant à 1).

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>Name</TH>";
    print "<TH>Type</TH>";
    print "<TH>Length</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>";
    }
```

```
    ociFreeStatement($stmt);
    ociLogoff($conn);
    print "</PRE>";
    print "</HTML>\n";
?>
```

Voir aussi `ocinumcols` , `ocicolumntype` et `ocicolumnsize` .

6.71.19 OCIColumnPrecision() : Bientôt documenté....

`int` **ocicolumnprecision** (`resource stmt` , `int col`)
[lien librairie]

Bientôt documenté....

6.71.20 OCIColumnScale() : Bientôt documenté....

`int` **ocicolumnscale** (`resource stmt` , `int col`)
[lien librairie]

Bientôt documenté....

6.71.21 ociColumnSize() : Retourne la taille de la colonne.

`int` **ocicolumnsize** (`resource stmt` , `mixed column`)
[lien librairie]

`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 `col` .

ociColumnSize

```
<?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>Name</TH>";
    print "<TH>Type</TH>";
    print "<TH>Length</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>";
```

```
    ociFreeStatement($stmt);
    ociLogoff($conn);
    print "</PRE>";
    print "</HTML>\n";
?>
```

Voir aussi `ocinumcols` , `ocicolumnname` et `ocicolumnsize` .

6.71.22 ociColumnType() : Retourne le type de données d'une colonne.

`mixed` **ocicolumntype** (`resource stmt` , `int col`)
[lien librairie]

`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 `ocicolumntype`

```
<?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>Name</TH>";
    print "<TH>Type</TH>";
    print "<TH>Length</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>";
    }
    ociFreeStatement($stmt);
    ociLogoff($conn);
    print "</PRE>";
    print "</HTML>\n";
?>
```

Voir aussi `ocinumcols` , `ocicolumnname` et `ocicolumnsize` .

6.71.23 OCIColumnTypeRaw() : Bientôt documenté....

`mixed` **ocicolumntyperaw** (`resource stmt` , `int col`)
[lien librairie]

Bientôt documenté....

6.71.24 ociCommit() : Valide les transactions en cours.

`int` **ocicommit** (`resource connection`)
[lien librairie]

`ocicommit` valide toutes les transactions en cours sur la connexion Oracle `connection` .

6.71.25 ociDefineByName() : Utilise une variable PHP pour la phase de définition, dans une commande SELECT.

`bool` **ocidefinebyname** (`resource stmt` , `string Column-Name` , `mixed variable` , *`int type`*)
[lien librairie]

`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'attends 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` .

ociDefineByName

```
<?php
/* Exemple ociDefineByPos par thies@thieso.net (980219) */
$conn = ociLogon("scott","tiger");
$stmt = ociparse($conn,"select empno, ename from emp");
/* La définition DOIT être faîte AVANT ociexecute! */
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);
?>
```

6.71.26 ociError() : Retourne la dernière erreur de stmt|conn|global.

`array` **ocierror** (*`int stmt|conn`*)
[lien librairie]

`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 .

6.71.27 ociexecute() : Exécute une commande.

`int` **ociexecute** (`resource statement` , *`int mode`*)
[lien librairie]

`ociexecute` éxécute 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.

6.71.28 ociFetch() : Modifie la prochaine ligne dans le résultat.

`bool` **ocifetch** (`resource statement`)
[lien librairie]

`ocifetch` place la prochaine ligne (d'une commande SELECT) dans le pointeur interne de résultat.

6.71.29 ociFetchInto() : Retourne la ligne suivante dans un tableau.

`int` **ocifetchinto** (`resource stmt` , `array &result` , *`int mode`*)
[lien librairie]

`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.

6.71.30 ociFetchStatement() : Retourne toutes les lignes d'un résultat.

`int` **ocifetchstatement** (`resource stmt` , `array &variable`)
[lien librairie]

`ocifetchstatement` retourne toutes les lignes d'un résultat dans le tableau variable. `ocifetchstatement` retourne le nombre de lignes retournées.

ociFetchStatement

```
<?php
/* exemple ociFetchStatement par mbritton@verinet.com (990624) */
$conn = ociLogon("scott","tiger");
$stmt = ociparse($conn,"select * from emp");
ociexecute($stmt);
$nrows = ociFetchStatement($stmt,$results);
if ( $nrows > 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 < $nrows; $i++ ) {
      reset($results);
      print "<TR>\n";
      while ( $column = each($results) ) {
         $data = $column['value'];
         print "<TD>$data[$i]</TD>\n";
      }
      print "</TR>\n";
   }
   print "</TABLE>\n";
} else {
   echo "Rien n'a été trouvé<br>\n";
}
print "$nrows Records Selected<br>\n";
ociFreeStatement($stmt);
ociLogoff($conn);
?>
```

6.71.31 OCIFreeCollection() : Bientôt documenté....

`bool` **ocifreecollection** (`object  lob`)
[lien librairie]

Bientôt documenté....

6.71.32 ociFreeCursor() : Libère toutes les ressources occupées par un pointeur.

`boolean` **ocifreecursor** (`resource  stmt`)
[lien librairie]

`ocifreecursor` retourne TRUE en cas de succès, et FALSE en cas d'échec.

6.71.33 ociFreeDesc() : Supprime un descripteur de LOB

`bool` **ocifreedesc** (`object  lob`)
[lien librairie]

`ocifreedesc` retourne TRUE en cas de succès, et FALSE en cas d'échec.

6.71.34 ociFreeStatement() : Libère toutes les ressources occupées par une commande.

`bool` **ocifreestatement** (`resource  stmt`)
[lien librairie]

ocifreestatement retourne TRUE en cas de succès, et FALSE en cas d'échec.

6.71.35 ociinternaldebug() : Active ou désactive l'affichage des données de debuggage.

void **ociinternaldebug** (int onoff)
[lien librairie]

ociinternaldebug active ou désactive l'affichage des informations de debuggage. Pour les afficher, mettez onoff à 1, ou sinon mettez onoff à 0 pour les cacher.

6.71.36 OCILoadLob() : Bientôt documenté....

string **ociloadlob** (object lob)
[lien librairie]

Bientôt documenté....

6.71.37 ociLogOff() : Déconnexion d'un serveur Oracle.

bool **ocilogoff** (resource connection)
[lien librairie]

ocilogoff ferme la connexion Oracle.

6.71.38 ociLogon() : Etablit une connexion à un serveur Oracle.

resource **ocilogon** (string username , string password , *string db*)
[lien librairie]

ocilogon retourne un identifiant de connexion, nécessaire à la plupart des fonctions oci. Si l'option ORACLE_SID n'est pas précisée, PHP utilisera la variable d'environnement ORACLE_SID pour déterminer le serveur de connexion.

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.

Cet exemple montre comment les connexions sont partagées :

ociLogon

```
<?php
print "<HTML><PRE>";
$db = "";
$c1 = ocilogon("scott","tiger",$db);
$c2 = ocilogon("scott","tiger",$db);
function create_table($conn)
{ $stmt = ociparse($conn,"create table scott.hallo (test
varchar2(64))");
  ociexecute($stmt);
```

```
  echo $conn." created table\n\n";
}
function drop_table($conn)
{ $stmt = ociparse($conn,"drop table scott.hallo");
  ociexecute($stmt);
  echo $conn." dropped table\n\n";
}
function insert_data($conn)
{ $stmt = ociparse($conn,"insert into scott.hallo values('$conn' || ' ' || to_char(sysdate,'DD-MON-Y
  ociexecute($stmt,oci_DEFAULT);
  echo $conn." inserted hallo\n\n";
}
function delete_data($conn)
{ $stmt = ociparse($conn,"delete from scott.hallo");
  ociexecute($stmt,oci_DEFAULT);
  echo $conn." deleted hallo\n\n";
}
function commit($conn)
{ ocicommit($conn);
  echo $conn." commited\n\n";
}
function rollback($conn)
{ ocirollback($conn);
  echo $conn." rollback\n\n";
}
function select_data($conn)
{ $stmt = ociparse($conn,"select * from scott.hallo");
  ociexecute($stmt,oci_DEFAULT);
  echo $conn."----selecting\n\n";
  while (ocifetch($stmt))
    echo $conn." <".ociresult($stmt,"TEST").">\n\n";
  echo $conn."----done\n\n";
}
create_table($c1);
insert_data($c1);   // Insertion d'une ligne avec c1
insert_data($c2);   // Insertion d'une ligne avec c2
select_data($c1);   // Les résultats des deux insertions sont retournés
select_data($c2);
rollback($c1);      // Annulation avec c1
select_data($c1);   // Les résultats des deux insertions sont annulés
select_data($c2);
insert_data($c2);   // Insertion d'une ligne avec c2
commit($c2);        // Validation avec using c2
select_data($c1);   // Le résultat de c2 est retourné
delete_data($c1);   // Effacement de toutes les lignes avec c1
select_data($c1);   // Aucune ligne n'est retournée
select_data($c2);   // Aucune ligne n'est retournée
commit($c1);        // Validation avec c1
select_data($c1);   // Aucune ligne n'est retournée
select_data($c2);   // Aucune ligne n'est retournée
drop_table($c1);
print "</PRE></HTML>";
?>
```

Voir aussi `ociplogon` et `ocinlogon` .

6.71.39 OCINewCollection() : Bientôt documenté....

`bool` **ocinewcollection** (`int conn` , `string tdo` , *`string shema`*)
[lien librairie]

Bientôt documenté....

6.71.40 ociNewCursor() : Retourne un nouveau pointeur à utiliser pour lier les pointeurs de références

`int` **ocinewcursor** `( resource conn )`
[lien librairie]

`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 stoquée info.output retourne un pointeur
// de curseur 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($data);
}
ociFreeStatement($curs);
ociFreeCursor($stmt);
ociLogoff($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);
?>
```

6.71.41 ociNewDescriptor() : Initialise un nouveau pointeur vide de LOB/FILE

`resource` **ocinewdescriptor** (`resource connection` , *`int type`*)
[lien librairie]

`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.

ocinewdescriptor

```
<?php
    /* Ce script est fait pour être appelé dans un formulaire HTML
     * Il attends les variables $user, $password, $table, $where, et $commitsize
     * Le script efface alors les lignes sélectionnées avec ROWID et valide
     * l'effacement après chaque groupe de $commitsize lignes.
     * (Utilisez avec prudence, car il n'y a pas d'annulation possible).
     */
    $conn = ociLogon($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 "$nrows effacées...\n";
    ociFreeStatement($stmt);
    ociLogoff($conn);
?>
```

Exemple avec `ocinewdescriptor`

```
<?php
    /* Appel d'une procédure PL/SQL stockée qui prend un clobs
     * en entrée (PHP 4 >= 4.0.6).
     * Exemple de signateure 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);
?>
```

6.71.42 ociNLogon() : Se connecte à un serveur Oracle avec une nouvelle connexion.

`resource` **ocinlogon** (`string username` , `string password` , *`string db`*)
[lien librairie]

`ocinlogon` crée une nouvelle connexion à un serveur Oracle et s'authentifie. Si l'option ORACLE_SID n'est pas spécifié, PHP utilisera la variable d'environnement ORACLE_SID pour déterminer le serveur de connexion.

`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 `ociplogon` . Si vous avez de multiples connexions ouvertes avec `ocinlogon` , les validations et annulations ne s'appliquent qu'à la connexion spécifée.

L'exemple ci−dessous montre l'utilisation des connexions séparées.

Exemple avec `ocinlogon`

```
<?php
print "<HTML><PRE>";
$db = "";
$c1 = ocilogon("scott","tiger",$db);
$c2 = ocinlogon("scott","tiger",$db);
function create_table($conn)
{ $stmt = ociparse($conn,"create table scott.hallo (test
varchar2(64))");
  ociexecute($stmt);
  echo $conn." created table\n\n";
}
function drop_table($conn)
{ $stmt = ociparse($conn,"drop table scott.hallo");
  ociexecute($stmt);
  echo $conn." dropped table\n\n";
}
function insert_data($conn)
{ $stmt = ociparse($conn,"insert into scott.hallo values('$conn' || ' ' || to_char(sysdate,'DD−MON−Y
  ociexecute($stmt,oci_DEFAULT);
  echo $conn." inserted hallo\n\n";
}
function delete_data($conn)
{ $stmt = ociparse($conn,"delete from scott.hallo");
  ociexecute($stmt,oci_DEFAULT);
  echo $conn." deleted hallo\n\n";
}
function commit($conn)
{ ocicommit($conn);
  echo $conn." commited\n\n";
}
function rollback($conn)
{ ocirollback($conn);
  echo $conn." rollback\n\n";
```

```
}
function select_data($conn)
{ $stmt = ociparse($conn,"select * from scott.hallo");
  ociexecute($stmt,oci_DEFAULT);
  echo $conn."----selecting\n\n";
  while (ocifetch($stmt))
    echo $conn." <".ociresult($stmt,"TEST").">\n\n";
  echo $conn."----done\n\n";
}
create_table($c1);
insert_data($c1);
select_data($c1);
select_data($c2);
rollback($c1);
select_data($c1);
select_data($c2);
insert_data($c2);
commit($c2);
select_data($c1);
delete_data($c1);
select_data($c1);
select_data($c2);
commit($c1);
select_data($c1);
select_data($c2);
drop_table($c1);
print "</PRE></HTML>";
?>
```

Voir aussi `ocilogon` et `ociplogon` .

6.71.43 ociNumCols() : Retourne le nombre de colonnes dans un résultat

`int` **ocinumcols** (`resource stmt`)
[lien librairie]

`ocinumcols` retourne le nombre de colonnes dans un résultat.

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";
?>
```

6.71.44 ociparse() : Analyse une requête.

`int` **ociparse** (`resource conn` , `string query`)
[lien librairie]

`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.

6.71.45 ociPLogon() : Connection persistante à un serveur Oracle.

`resource` **ociplogon** (`string username` , `string password` , *`string db`*)
[lien librairie]

`ociplogon` crée une connexion persistante à un serveur Oracle 8 et s'authentifie. Si l'option ORACLE_SID n'est pas spécifiée, PHP utilisera la variable d'environnement ORACLE_SID pour déterminer le serveur de connexion.

Voir aussi `ocilogon` et `ocinlogon` .

6.71.46 ociResult() : Retourne la valeur d'une colonne

`mixed` **ociresult** (`resource statement` , `mixed column`)
[lien librairie]

`ociresult` retourne les données de la colonne `column` dans la ligne courante (voir `ocifetch`). `ocifetch` retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILEs).

6.71.47 ociRollback() : Annule les transactions en cours.

`int` **ocirollback** (`resource connection`)
[lien librairie]

`ocirollback` annule les transactions en cours sur la connexion Oracle `connection` .

6.71.48 ociRowCount() : Retourne le nombre de lignes affectées.

`int` **ocirowcount** (`resource statement`)
[lien librairie]

`ocirowcount` retourne le nombre de lignes affectées par une commande de modification. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées.

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>";
?>
```

6.71.49 OCISaveLob() : Bientôt documenté....

`bool` **ocisavelob** `( object lob )`
[lien librairie]

Bientôt documenté....

6.71.50 OCISaveLobFile() : Bientôt documenté....

`bool` **ocisavelobfile** `( object lob )`
[lien librairie]

Bientôt documenté....

6.71.51 ociServerVersion() : Retourne une chaîne contenant les informations de version du serveur.

`string` **ociserverversion** `( resource conn )`
[lien librairie]

`ociserverversion` retourne une chaîne contenant les informations de version du serveur

Exemple avec `ociserverversion`

```
<?php
   $conn = ociLogon("scott","tiger");
   print "Version du serveur : " . ociServerVersion($conn);
   ociLogOff($conn);
?>
```

6.71.52 ocisetprefetch() : Indique le nombre de lignes qui doivent être pré−lues

`int` **ocisetprefetch** (`resource stmt` , `int rows`)
[lien librairie]

`ocisetprefetch` indique le nombre des premières lignes qui doivent être pré−lues. La valeur par défaut est de 1.

6.71.53 ociStatementType() : Retourne le type de commande OCI.

`string` **ocistatementtype** (`resource stmt`)
[lien librairie]

`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"

Exemples

```
<?php
    print "<HTML><PRE>";
    $conn = ociLogon("scott","tiger");
    $sql  = "delete from emp where deptno = 10";
    $stmt = ociparse($conn,$sql);
    if ( ociStatementType($stmt) == "DELETE" ) {
        die "Vous n'etes pas autorisé à effacer dans cette table.<BR>";
    }
    ociLogoff($conn);
    print "</PRE></HTML>";
?>
```

6.71.54 OCIWriteLobToFile() : Bientôt documenté....

`bool` **ociwritelobtofile** (`object lob` , *`string filename`* , *`int start`* , *`int lenght`*)
[lien librairie]

Bientôt documenté....

6.72 OpenSSL

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 utiliser ce module à vos risques et périls.

6.72.1 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.

6.72.2 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.

6.72.3 Installation

Pour utiliser le support OpenSSL de PHP, vous devez aussi compiler PHP avec l'option de configuratoin `−−with−openssl[=DIR]` .

6.72.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.72.5 Types de ressources

6.72.6 Paramètres clés/certificats

Un bon nombre de fonctions OpenSSL demandent une clé et un certificat comme paramètres. PHP 4.0.5 et plus récent utilisait des clés ou certificats sous forme de ressource, retournée par l'une des fonctions openssl_get_xxx(). Les versions ultérieures utilisent l'une des méthodes suivantes :

- Certificats
    1. Une ressource X.509 retournée par `openssl_x509_read`
    2. Une chaîne au format `file://path/to/cert.pem` ; Le fichier ainsi repéré doit contenir un certificat, encodé au format PEM
    3. Une chaîne contenant le contenu d'un certificat, encodé au format PEM.
- Clés publiques/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 privé 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.

6.72.7 Vérification de certificats

Lorsque vous appelez une fonction qui va vérifier une signature ou un certificat, le paramètre **cainfo** doit être un tableau contenant les noms d'un dossier et d'un fichier 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
- Paramètres clés/certificats
- Vérification de certificats
- 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_seal
- openssl_sign
- openssl_verify
- openssl_x509_checkpurpose
- openssl_x509_free
- openssl_x509_parse
- openssl_x509_read

6.72.9 openssl_free_key() : Libère les ressources

`void` **openssl_free_key** (`resource key_identifier`)
[lien librairie]

`openssl_free_key` libère les ressources associées à `key_identifier` .

6.72.10 openssl_get_privatekey() : Prépare une clé privée au format PEM

`resource` **openssl_get_privatekey** (`mixed key` , *`string passphrase`*)
[lien librairie]

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).

6.72.11 openssl_get_publickey() : Extrait une clé publique d'un certificat

resource **openssl_get_publickey** (mixed certificate)
[lien librairie]

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.

6.72.12 openssl_open() : Ouvre des données scellées

boolean **openssl_open** (string sealed_data , string open_data , string env_key , mixed priv_key_id)
[lien librairie]

openssl_open TRUE en cas de succès, et FALSE sinon. En cas de succès, les données déchiffrées sont placées dans open_data .

openssl_open ouvre (déchiffre) les données sealed_data en utilisant la clé prviée priv_key_id et la clé d'enveloppe env_key et remplis 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.

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);
// déchiffrage 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 .

6.72.13 openssl_pkcs7_decrypt() : Déchiffre un message S/MIME

boolean **openssl_pkcs7_decrypt** (string infilename , string outfilename , mixed recipcert , mixed recipkey)
[lien librairie]

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'evoluer 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.

6.72.14 openssl_pkcs7_encrypt() : Chiffre un message S/MIME

boolean **openssl_pkcs7_encrypt** (string infilename , string outfilename , mixed recipcerts , array headers , *long flags*)
[lien librairie]

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'evoluer 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" => "Eyes only")))
{
    // message chiffré : envoyez le!
    exec(ini_get("sendmail_path") . " < enc.txt");
}
?>
```

Note
Ces constantes ont été ajoutées en PHP 4.0.6.

6.72.15 openssl_pkcs7_sign() : Signe un message S/MIME

`boolean` **openssl_pkcs7_sign** (`string infilename` , `string outfilename` , `mixed signcert` , `mixed privkey` , `array headers` , *`long flags`* , *`string extracertsfilename`*)
[lien librairie]

`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` spécifi 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'evoluer 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", // keyed syntax
          "From: HQ <ceo@sales.com>", // indexed syntax
          "Subject" => "Eyes only"))
{
    // message signed - send it!
    exec(ini_get("sendmail_path") . " < signed.txt");
}
?>
```

Note
Ces constantes ont été ajoutées en PHP 4.0.6.

6.72.16 openssl_pkcs7_verify() : Vérifie la signature d'un message S/MIME

`boolean` **openssl_pkcs7_verify** (`string filename` , `int flags` , *`string outfilename`* , *`array cainfo`* , *`string extracerts`*)
[lien librairie]

`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'evoluer d'ici à la prochaine version de PHP.

Note
Ces constantes ont été ajoutées en PHP 4.0.6.

6.72.17 openssl_seal() : Scelle des données

`int` **openssl_seal** (`string data` , `string sealed_data` , `array env_keys` , `array pub_key_ids`)
[lien librairie]

`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 secrètre 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 signifique que vous pouvez envoyez des données scellées à plusieurs destinataires (en supposant que chacun ait recu 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 possessuers de $pk1 et $pk2 peuvent déchiffrer
// le message $sealed avec les clés $ekeys[0] et $ekeys[1] (respectivement).
openssl_seal($data, $sealed, $ekeys, array($pk1,$pk2));
// libère les clés de la mémoire
openssl_free_key($pk1);
openssl_free_key($pk2);
?>
```

Voir aussi `openssl_open` .

6.72.18 openssl_sign() : Signe les données

`boolean` **openssl_sign** (`string data` , `string signature` , `mixed priv_key_id`)
[lien librairie]

`openssl_sign` retourne TRUE en cas de succès, et FALSE sinon. En cas de succès, la signature est placée dans `signature` .

`openssl_sign` calcule la signature des données `data` en utilisant l'algorithme SHA1 (hashing) suivi du chiffrage avec la clé privée `priv_key_id` . Notez que les données elles-mêmes ne sont pas chiffrées.

Exemple avec `openssl_sign`

```php
<?php
// On suppose que $data contient les données à signer
// lecture de la clé publique pour chaque destinataire
$fp = fopen("/src/openssl-0.9.6/demos/sign/key.pem", "r");
$priv_key = fread($fp, 8192);
fclose($fp);
$pkeyid = openssl_get_privatekey($priv_key);
// calcule de la signature
openssl_sign($data, $signature, $pkeyid);
// libère les clés de la mémoire
openssl_free_key($pkeyid);
?>
```

Voir aussi `openssl_verify` .

6.72.19 openssl_verify() : Vérifie une signature

`int` **openssl_verify** (`string data` , `string signature` , `resource pub_key_id`)
[lien librairie]

`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 avecd 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
<?php
// On suppose que $data et $signature contiennent les données à signer et
// la signature
// lecture de la clé publique depuis le certificat
$fp = fopen("/src/openssl-0.9.6/demos/sign/cert.pem", "r");
$cert = fread($fp, 8192);
fclose($fp);
$pubkeyid = openssl_get_publickey($cert);
// indique si la signature est correcte
$ok = openssl_verify($data, $signature, $pubkeyid);
if ($ok == 1)
    echo "Signature valide";
elseif ($ok == 0)
    echo "Signature erronée";
else
    echo "Erreur de vérification de la signature";
// libère les clés de la mémoire
openssl_free_key($pubkeyid);
?>
```

Voir aussi `openssl_sign` .

6.72.20 openssl_x509_checkpurpose() : Vérifie l'usage d'un certificat

`boolean` **openssl_x509_checkpurpose** (`mixed x509cert` , `int purpose` , `array cainfo` , *`string untrustedfile`*)

[lien librairie]

`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'evoluer d'ici à la prochaine version de PHP.

Constante	Description
X509_PURPOSE_SSL_CLIENT	Est ce que le certificat peut être utilisé avec le client d'une connexion SSL?
X509_PURPOSE_SSL_SERVER	Est ce que le certificat peut être utilisé avec le serveur d'une connexion SSL?
X509_PURPOSE_NS_SSL_SERVER	Est ce que le certificat peut être utilisé avec un serveur Netscape d'une connexion SSL?
X509_PURPOSE_SMIME_SIGN	Est ce que le certificat peut être utilisé pour signer des courrier à la norme S/MIME?
X509_PURPOSE_SMIME_ENCRYPT	Est−ce que le certificat peut être utilisé pour chiffrer un courrier au format S/MIME?
X509_PURPOSE_CRL_SIGN	Est−ce que le certificat peut être utilisé pour chiffrer une liste de revocation de certificat? (CRL)?
X509_PURPOSE_ANY	Est−ce que le certificat peut être utilisé pour n'importe lequel de ces cas?

Ces options ne sont pas des champs de bits : vous ne pouvez en passer qu'une seule à la fois.

Note
Ces constantes ont été ajoutées en PHP 4.0.6.

6.72.21 openssl_x509_free() : Libère les ressources prises par un certificat

`void` **openssl_x509_free** (`resource x509cert`)
[lien librairie]

`openssl_x509_free` libère les ressources prises par le certificat `x509cert`

Note
Ces constantes ont été ajoutées en PHP 4.0.6.

6.72.22 openssl_x509_parse() : Analyse un certificat X509.

`array` **openssl_x509_parse** (`mixed x509cert` , *`boolean shortnames`*)
[lien librairie]

`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.

6.72.23 openssl_x509_read() : Analyse un certificat X.509 et retourne une ressource

`resource` **openssl_x509_read** (`mixed x509certdata`)
[lien librairie]

`openssl_x509_read` analyse le certificat `x509certdata` et retourne un identifiant de ressource.

Note
Ces constantes ont été ajoutées en PHP 4.0.6.

6.73 Oracle

6.73.1 Introduction

Cette extension ajoute le supoprt des bases de données Oracle. Voyez aussi OCI8 .

6.73.2 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` (nombre ˆ virgule flottante)
`ORA_BIND_IN` (nombre ˆ virgule flottante)
`ORA_BIND_OUT` (nombre ˆ virgule flottante)
`ORA_FETCHINTO_ASSOC` (nombre ˆ virgule flottante)
`ORA_FETCHINTO_NULLS` (nombre ˆ virgule flottante)

- Introduction
- 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

6.73.4 Ora_Close() : Ferme un pointeur Oracle.

`int` **ora_close** (`resource cursor`)
[lien librairie]

`ora_close` retourne TRUE si la fermeture a bien eu lieu, et FALSE sinon. Les erreurs sont accessibles avec les fonctions `ora_error` et `ora_errorcode` .

ora_close termine les pointeurs ouverts avec la fonction ora_open .

6.73.5 Ora_ColumnName() : Retourne le nom de la colonne de résultat.

string **ora_columnname** (resource cursor , int column)
[lien librairie]

ora_columnname retourne le nom du champs column du pointeur cursor . Le nom retourné sera en majuscule.

6.73.6 Ora_ColumnSize() : Lit la taille d'une colonne

int **ora_columnsize** (resource cursor , int column)
[lien librairie]

ora_columnsize retourne la taille de la colonne column dans le résultat cursor .

6.73.7 Ora_ColumnType() : Retourne le type de la colonne de résultat.

string **ora_columntype** (resource cursor , int column)
[lien librairie]

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"

6.73.8 Ora_Commit() : Valide une transaction Oracle.

int **ora_commit** (resource conn)
[lien librairie]

ora_commit retourne TRUE si la validation a bien eu lieu, et FALSE sinon. Les erreurs sont accessibles avec les fonctions ora_error et ora_errorcode .

ora_commit valide les transactions Oracle. Une transaction est définie par toutes les requêtes effectuées sur la connexion conn depuis la dernière validation ou annulation (avec auto−validation inactivée) ou depuis l'établissement de la connexion.

6.73.9 Ora_CommitOff() : Inactive la validation automatique.

`int` **ora_commitoff** (`resource conn`)
[lien librairie]

`ora_commitoff` retourne TRUE si la désactivation a bien eu lieu, et FALSE sinon. Les erreurs sont accessibles avec les fonctions `ora_error` et `ora_errorcode` .

`ora_commitoff` inactive la validation automatique après chaque `ora_exec` .

6.73.10 Ora_CommitOn() : Active la validation automatique.

`int` **ora_commiton** (`resource conn`)
[lien librairie]

`ora_commiton` active la validation automatique après chaque `ora_exec` .

`ora_commiton` retourne TRUE si l'activation a bien eu lieu, et FALSE sinon. Les erreurs sont accessibles avec les fonctions `ora_error` et `ora_errorcode` .

6.73.11 Ora_Do() : Analyse, exécute et lit une requête

`int` **ora_do** (`resource conn` , `string query`)
[lien librairie]

`ora_do` est une combinaison de `ora_parse` , `ora_exec` et `ora_fetch` . Elle va analyser la requête, l'éxécuter et lire la première ligne du résultat.

`ora_do` retourne TRUE en cas de succès, et FALSE sinon. Le détails sur les erreurs est accessible avec les fonctions `ora_error` et `ora_errorcode` .

Voir aussi `ora_parse` , `ora_exec` et `ora_fetch` .

6.73.12 Ora_Error() : Retourne le message d'erreur Oracle.

`string` **ora_error** (`resource cursor_or_connection`)
[lien librairie]

`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 avoir des messages d'erreur tels que :
$

```
Array

00001, 00000, "unique constraint (%s.%s) violated" // *Cause: An update
```

```
or insert
statement attempted to insert a duplicate key // For Trusted
ORACLE configured in DBMS MAC mode, you may see // this message
if a duplicate entry exists at a different level. // *Action: Either
remove the unique restriction or do not insert the key.
```

6.73.13 Ora_ErrorCode() : Retourne le code d'erreur Oracle.

`int` **ora_errorcode** (`resource cursor_or_connection`)
[lien librairie]

`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.

6.73.14 Ora_Exec() : Exécute une commande analysée sur un pointeur Oracle.

`boolean` **ora_exec** (`resource cursor`)
[lien librairie]

`ora_exec` retourne TRUE en cas de succès, et FALSE en cas d'erreur. L'erreur générée sera alors accessible avec les fonctions `ora_error` et `ora_errorcode` .

6.73.15 Ora_Fetch_Into() : Lit une ligne dans un tableau

`int` **ora_fetch_into** (`resource cursor` , `array result` , *`int flags`*)
[lien librairie]

`ora_fetch_into` lit la ligne courante du résultat `cursor` dans le tableau `result` .

Exemple `ora_fetch_into`

```php
<?php
array($results);
ora_fetch_into($cursor, &$results);
echo $results[0];
echo $results[1];
?>
```

Notez que vous devez passer le tableau par référence;

Voir aussi `ora_parse` , `ora_exec` , `ora_fetch` et `ora_do` .

6.73.16 Ora_Fetch() : Retourne une ligne de résultat.

`int` **ora_fetch** (`resource cursor`)
[lien librairie]

`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` .

Lit une ligne de données sur le pointeur cursor.

Voir aussi `ora_parse` , `ora_exec` et `ora_do` .

6.73.17 Ora_GetColumn() : Retourne une donnée d'une ligne lue.

`mixed` **ora_getcolumn** (`resource cursor` , `mixed column`)
[lien librairie]

`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".

6.73.18 Ora_Logoff() : Ferme une connexion Oracle.

`int` **ora_logoff** (`resource connection`)
[lien librairie]

`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").

6.73.19 Ora_Logon() : Ouvre une connexion Oracle.

`resource` **ora_logon** (`string user` , `string password`)
[lien librairie]

`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 faîtes avec `SQL*Net` en fournissant le nom `TNS` de la manière suivante :

```
<?php
$conn = ora_logon("user
<B>@TNSNAME</B>
", "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 .

6.73.20 Ora_Numcols() : Retourne le nombre de colonnes

int **ora_numcols** (resource cursor_ind)
[lien librairie]

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 .

6.73.21 Ora_Numrows() : Retourne le nombre de colonnes

int **ora_numrows** (resource cursor_ind)
[lien librairie]

ora_numrows retourne le nombre de colonnes dans le résultat cursor_ind .

6.73.22 Ora_Open() : Ouvre un pointeur Oracle.

resource **ora_open** (resource connection)
[lien librairie]

ora_open ouvre un pointeur Oracle sur la connexion.

ora_open retourne TRUE si la fermeture a bien eu lieu, et FALSE sinon. Les erreurs sont accessibles avec les fonctions ora_error et ora_errorcode .

6.73.23 Ora_Parse() : Analyse une requête SQL.

int **ora_parse** (resource cursor_ind , string sql_statement , int defer)
[lien librairie]

ora_parse analyse une requête SQL ou un bloc PL/SQL et l'associe avec le pointeur cursor_ind. Retourne 0 en cas de succès, et −1 en cas d'erreur.

ora_parse retourne 0 en cas de succès, et −1 en cas d'erreur.

Voir aussi ora_exec , ora_fetch et ora_do .

6.73.24 Ora_pLogon() : Ouvre une connexion persistante à Oracle

`resource` **ora_plogon** (`string user` , `string password`)
[lien librairie]

`ora_plogon` établit une connexion persistante à un serveur Oracle, avec l'utilisateur `user` et le mot de passe `password` .

Voir aussi `ora_logon` .

6.73.25 Ora_Rollback() : Annule une transaction.

`int` **ora_rollback** (`resource connection`)
[lien librairie]

`ora_rollback` annule une transaction Oracle. (Voir `ora_commit` pour la définition d'une transaction).

`ora_rollback` retourne `TRUE` si la fermeture a bien eu lieu, et `FALSE` sinon. Les erreurs sont accessibles avec les fonctions `ora_error` et `ora_errorcode` .

6.74 Ovrimos SQL

Ovrimos SQL Server est une base de données relationnelle client/serveur et transactionnelle, combinée avec des fonctionnalités web, et des transactions rapides.

Ovrimos SQL Server est disponible à www.ovrimos.com . Pour activer le support ovrimos de PHP, il suffit de compiler PHP avec l'option `--with-ovrimos` du script de configuration. Vous devrez aussi installer la librairie sqlcli disponible avec la distribution Ovrimos SQL Server.

Connection au serveur Ovrimos SQL Server et sélection d'une table système

```
<?php
$conn = ovrimos_connect("server.domain.com", "8001", "admin", "password");
if ($conn != 0) {
    echo ("Connection 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);
}
?>
```

Cet exemple effectue une connexion réussie.

- 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

6.74.2 ovrimos_commit() : Valide une transaction

`int` **ovrimos_commit** `( int connection_id )`
[lien librairie]

`ovrimos_commit` sert à exécuter une transaction.

ovrimos_commit exécute la transaction préparée sur la connexion connection_id .

6.74.3 ovrimos_connect() : Connexion à un serveur

int **ovrimos_connect** (string host , string db , string user , string password)
[lien librairie]

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 "Connection 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.

6.74.4 ovrimos_cursor() : Retourne le nom du curseur

int **ovrimos_cursor** (int result_id)
[lien librairie]

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.

6.74.5 ovrimos_exec() : Exécute une requête SQL

int **ovrimos_exec** (int connection_id , string query)
[lien librairie]

ovrimos_exec sert à exécuter une requête SQL.

ovrimos_exec exécute une requête SQL (selection 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.

6.74.6 ovrimos_execute() : Exécute une requête préparée

`boolean` **ovrimos_execute** (`int result_id` , *`array parameters_array`*)
[lien librairie]

`ovrimos_execute` sert à exécuter une requête SQL.

`ovrimos_execute` exécute une requête préparée. `ovrimos_execute` retourne TRUE ou FALSE . Si la requête préparée contient des paramètres (des points d'interrogations dans la requête), un nombre correct de paramètre 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.

6.74.7 ovrimos_fetch_into() : Lit une ligne dans un résultat

`boolean` **ovrimos_fetch_into** (`int result_id` , `array result_array` , *`string how`* , *`int rownumber`*)
[lien librairie]

`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'. `ovrimos_fetch_into` retourne TRUE ou FALSE .

Lit un exemple

```
<?php
$conn=ovrimos_connect("neptune", "8001", "admin", "password");
if ($conn!=0) {
    echo "Connection 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 lis une ligne.

6.74.8 ovrimos_fetch_row() : Lit une ligne dans un résultat

`boolean` **ovrimos_fetch_row** (`int result_id` , *`int how`* , *`int row_number`*)
[lien librairie]

`ovrimos_fetch_row` lit une ligne dans un résultat SQL.

`ovrimos_fetch_row` lit une ligne dans un résultat. Les colonnes doivent être lues par un autre appel. Retourne TRUE en cas de succès et FALSE sinon.

Exemple de lecture de ligne

```
<?php
$conn = ovrimos_connect("remote.host", "8001", "admin", "password");
if ($conn != 0) {
    echo "Connection 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.

6.74.9 ovrimos_field_len() : Retourne la taille d'une colonne

`int` **ovrimos_field_len** (`int result_id` , `int field_number`)
[lien librairie]

`ovrimos_field_len` sert à connaître la taille d'une colonne.

`ovrimos_field_len` retourne la taille de la colonne `field_number` , dans le résultat `field_number` .

6.74.10 ovrimos_field_name() : Retourne le nom d'une colonne

`int` **ovrimos_field_name** (`int result_id` , `int field_number`)
[lien librairie]

ovrimos_field_name sert à obtenir le nom d'une colonne.

ovrimos_field_name retourne le nom d'une colonne à partir de son numéro de colonne field_number , (la première colonne est à 1).

6.74.11 ovrimos_field_num() : Retourne le numéro de colonne

int **ovrimos_field_num** (int result_id , string field_name)
[lien librairie]

ovrimos_field_num sert à connaître le nuémro de colonne, à partir de son nom.

ovrimos_field_num retourne le numéro de la colonne field_name (la numérotation commence à 1), dans result_id .

6.74.12 ovrimos_field_type() : Retourne le type numérique d'une colonne

int **ovrimos_field_type** (int result_id , int field_number)
[lien librairie]

ovrimos_field_type sert à connaître le type numérique d'une colonne.

ovrimos_field_type retourne le type numérique d'une colonne, identifiée par son numéro field_number dans le résultat field_number .

6.74.13 ovrimos_free_result() : Libère les ressources utilisées par un résultat

int **ovrimos_free_result** (int result_id)
[lien librairie]

ovrimos_free_result sert à effacer un résultat.

ovrimos_free_result libère toutes les ressources prises par le résultat result_id . ovrimos_free_result retourne TRUE .

6.74.14 ovrimos_longreadlen() : Indique la taille des données à lire dans une colonne de grande taille

int **ovrimos_longreadlen** (int result_id , int length)
[lien librairie]

ovrimos_longreadlen sert à lire la taille des données qui sera lues lors de l'accès une colonne de grande taille.

ovrimos_longreadlen indique le nombre d'octets qui seront lus dans une colonne de grande taille (long varchar et long varbinary). Par défaut, 0. Indépendemment du fait que ovrimos_longreadlen requiert result_id , actuellement ovrimos_longreadlen affecte ce

paramètre pour tous les résultats.

6.74.15 ovrimos_num_fields() : Retourne le nombre de colonnes

`int` **ovrimos_num_fields** (`int result_id`)
[lien librairie]

`ovrimos_num_fields` indique le nombre de colonnes du résultat `result_id` .

6.74.16 ovrimos_num_rows() : Retourne le nombre de lignes affectées par une modification

`int` **ovrimos_num_rows** (`int result_id`)
[lien librairie]

`ovrimos_num_rows` retourne le nombre de lignes affectées par une modification

6.74.17 ovrimos_prepare() : Prépare une requête SQL

`int` **ovrimos_prepare** (`int connection_id` , `string query`)
[lien librairie]

`ovrimos_prepare` sert à préparer une requête SQL.

`ovrimos_prepare` prépare une requête SQL et retourne un identifiant de résultat result_id (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 "Connection 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.

6.74.18 ovrimos_result_all() : Affiche un résultat sous forme de table HTML

`boolean` **ovrimos_result_all** (`int result_id` , *`string format`*)
[lien librairie]

`ovrimos_result_all` sert à afficher tout le résultat d'une requête.

`ovrimos_result_all` affiche le résultat de la requête représentée par `result_id` .
`ovrimos_result_all` retourne TRUE ou FALSE .

Prépare une requête, l'éxécute, et affiche le résultat

```
<?php
$conn = ovrimos_connect("db_host", "8001", "admin", "password");
if ($conn != 0) {
    echo "Connection 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 "Connection 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 "Connection 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);
}
?>
```

6.74.19 ovrimos_result() : Lit le contenu d'une colonne

`int` **ovrimos_result** (`int result_id` , `mixed field`)
[lien librairie]

`ovrimos_result` sert à lire le contenu de l;a colonne `field` dans le résultat `result_id` .

`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).

6.74.20 ovrimos_rollback() : Annule une transaction

`int` **ovrimos_rollback** (`int connection_id`)
[lien librairie]

`ovrimos_rollback` sert à annuler une transaction.

`ovrimos_rollback` annule la transaction préparée sur la connexion `connection_id` .

6.75 Entrées/sorties

6.75.1 Introduction

Les fonctions d'entrée/sorties 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.

6.75.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.75.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.75.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
output_buffering	"0"	PHP_INI_PERDIR|PHP_INI_SYSTEM
output_handler	NULL	PHP_INI_PERDIR|PHP_INI_SYSTEM
implicit_flush	"0"	PHP_INI_PERDIR|PHP_INI_SYSTEM

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les 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 `On` . Si vous souhaitez limiter la taille du buffer à une certaine taille, vous pouvez alors indiquer un nombre maximum d'octets à la place de `On` . Par exemple, `output_buffering=4096`).

`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 sérieuse implications et généralement, cela n'est conseillé que pour les débogage. Cette valeur est par défaut à `TRUE` lorsque PHP fonctionne en mode `CLI SAPI` .

Voir aussi `ob_implicit_flush` .

6.75.5 Types de ressources

Cette extension ne définit aucune ressource.

6.75.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

6.75.7 Exemples

Output Control example

```
<?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 (&4.2) à 4.3, assurez vous que `implict_flush` est à `OFF` dans votre `php.ini` , sinon la fonction `ob_start` ne masquera pas les affichages engendrés.

6.75.8 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_contents
- ob_get_length
- ob_get_level
- ob_get_status
- ob_gzhandler
- ob_implicit_flush
- ob_start

6.75.10 ob_clean() : Efface le buffer de sortie

`void` **ob_clean** (`void`)
[lien librairie]

`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` .

6.75.11 ob_end_clean() : Détruit les données du buffer de sortie, et éteint la bufferisation de sortie

`void` **ob_end_clean** (`void`)
[lien librairie]

`ob_end_clean` détruit les données du buffer de sortie, et éteint la bufferisation.

Voir aussi `ob_start` et `ob_end_flush` .

6.75.12 ob_end_flush() : Envoie les données du buffer de sortie, et éteint la bufferisation de sortie

`void` **ob_end_flush** (`void`)
[lien librairie]

`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` .

Voir aussi `ob_start` , `ob_get_contents` et `ob_end_clean` .

6.75.13 ob_flush() : Envoie le buffer de sortie

`void` **ob_flush** (`void`)
[lien librairie]

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` .

6.75.14 ob_get_contents() : Retourne le contenu du buffer de sortie

`string` **ob_get_contents** (`void`)
[lien librairie]

`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` .

6.75.15 ob_get_length() : Retourne la longueur du contenu du buffer de sortie

`string` **ob_get_length** (`void`)
[lien librairie]

`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` .

6.75.16 ob_get_level() : Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie

`int` **ob_get_level** (`void`)
[lien librairie]

ob_get_level retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie.

Voir aussi `ob_start` et `ob_get_contents` .

6.75.17 ob_get_status() : Lit le statut du buffer de sortie

`array` **ob_get_status** (*`bool full_status`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.75.18 ob_gzhandler() : Fonction de callback pour la compression automatique des buffers

`string` **ob_gzhandler** (`string buffer`)
[lien librairie]

`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.

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` .

6.75.19 ob_implicit_flush() : Active/désactive l'envoi implicite

`void` **ob_implicit_flush** (*`int flag`*)
[lien librairie]

`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` .

6.75.20 ob_start() : Enclenche la bufferisation de sortie

`void` **ob_start** (*`string  output_callback`*)
[lien librairie]

`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 callback avec fonction utilisateur

```
<?php
function callback($buffer) {
  // remplace toutes les pommes par des oranges
  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` .

6.76 Overload

6.76.1 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 utiliser ce module à vos risques et périls.

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 nécessaire 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.

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 nouvs 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
- overload

6.77 PDF

Vous disposez de fonctions PDF en PHP pour créer des fichiers PDF, pour peu que vous ayez la bibliothèque PDF de Thomas Merz (disponible à : http://www.pdflib.com/pdflib/index.html (site anglais)). Vous aurez aussi besoin des librairies JPEG library , the TIFF library , pour compiler cette librairie. Ces deux librairies posent pas mal de problèmes lors de la configuration. Suivez attentivement les messages d'erreur.

Reportez vous à l'excellente documentation de PDFLib, disponible avec la distribution de PDFLib. C'est une introduction très pratique des possibilités de PDFLib et elle contient la liste la plus complète et les descriptions les plus à jours des fonctions.

Toutes les fonctions de PDFLib se retrouvent dans PHP sous le même nom. De même, les paramètres sont identiques. Vous devez connaître les concepts de base de PDF ou de Postscript pour utiliser efficacement ce module. Toutes les longueurs et coordonnées sont mesurées en points Postscript points. Il y a généralement 72 points PostScript par pouce, mais cela dépend en fait de la résolution d'affichage.

Il y a un autre module PHP pour créer des document PDF, basé sur la bibliothèque FastIO's 's ClibPDF. Les API sont légèrement différentes. Reportez−vous à la section fonctions ClipPDF pour plus de détails.

Le module PDF introduit un nouveau type de variables. C'est `pdfdoc` : c'est un pointeur sur un document PDF et toutes les fonctions l'utilise comme premier paramètre.

6.77.1 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 complétement abandonné. La liste suivante indique quelles sont les fonctions obsolètes en PHP 4.02, et qui devraient être remplacées par de nouvelles versions.

Anciennes fonctions	Nouvelles fonctions
pdf_put_image()	Désormais inutile
pdf_execute_image()	Désormais inutile
pdf_get_annotation()	`pdf_get_bookmark` avec les mêmes paramètres.
pdf_get_font()	`pdf_get_value` avec `"font"` comme second paramètre.
pdf_get_fontsize()	`pdf_get_value` avec `"fontsize"` comme second paramètre.
pdf_get_fontname()	`pdf_get_parameter` avec `"fontname"` comme second paramètre.
pdf_set_info_creator()	`pdf_set_info` avec `"Creator"` comme second paramètre.
pdf_set_info_title()	`pdf_set_info` avec `"Title"` comme second paramètre.
pdf_set_info_subject()	`pdf_set_info` avec `"Subject"` comme second paramètre.
pdf_set_info_author()	`pdf_set_info` avec `"Author"` comme second paramètre.
pdf_set_info_keywords()	`pdf_set_info` avec `"Keywords"` comme second paramètre.
pdf_set_leading()	`pdf_set_value` avec `"leading"` comme second paramètre.
pdf_set_text_rendering()	`pdf_set_value` avec `"textrendering"` comme second paramètre.

pdf_set_text_rise()	pdf_set_value avec "textrise" comme second paramètre.
pdf_set_horiz_scaling()	pdf_set_value avec "horizscaling" comme second paramètre.
pdf_set_text_matrix()	Désormais inutile
pdf_set_char_spacing()	pdf_set_value avec "charspacing" comme second paramètre.
pdf_set_word_spacing()	pdf_set_value avec "wordspacing" comme second paramètre.
pdf_set_transition()	pdf_set_parameter avec "transition" comme second paramètre.
pdf_open()	pdf_new suivi d'un appel à pdf_open_file
pdf_set_font()	pdf_findfont suivi d'un appel à pdf_setfont
pdf_set_duration()	pdf_set_value avec "duration" comme second paramètre.
pdf_open_gif()	pdf_open_image_file avec "gif" comme second paramètre.
pdf_open_jpeg()	pdf_open_image_file avec "jpeg" comme second paramètre.
pdf_open_tiff()	pdf_open_image_file avec "tiff" comme second paramètre.
pdf_open_png()	pdf_open_image_file avec "png" comme second paramètre.
pdf_get_imagewidth()	pdf_get_value avec "imagewidth" comme second paramètre et l'image comme troisième.
pdf_get_imageheight()	pdf_get_value avec "imageheight" comme second paramètre et l'image comme troisième.

6.77.2 Conseils pour installer PDFLib 3.x

Depuis la version 3.0 de PDFLib vous pouvez configurer cette librairie avec l'option `--enable-shared-pdflib` .

6.77.3 Choix de la version de PDFlib

Avec toutes les versions de PHP 4, ultérieure au 9 mars 2000, vous devez utiliser PDFlib 3.0 ou plus récent.

PHP 3, d'un autre coté, ne doit pas être utilisé avec une version plus récente que la 2.01. Depuis la version 1.61 du source `php3/functions/pdf.c` (php 3.19), il est possible d'utiliser la version PDFlib 3.0 ou plus récent.

6.77.4 Installation des anciennes versions de PDFLib

Si vous utilisez PDFLib 2.01 vérifiez comment votre librairie a été installée. Il doit y avoir un fichier (ou un lien) vers `libpdf.so` . La version 2.01 ne fait que créer une librairie avec le nom `libpdf2.01.so` qui ne peut être trouvé lors de la compilation du programme de configuration. Vous devez créer vous même ce lien symbolique de `libpdf.so` vers `libpdf2.01.so` .

La version 2.20 de PDFLib a introduit de nombreuses modifications dans ses API, ainsi que le support des polices chinoises et japonaises. Cela impliquent malheureusement des modifications dans le module PDF de PHP 4 (mais pas de PHP 3). Si vous utilisez PDFLib 2.20, gérer correctement votre mémoire. Jusqu'à la version 3.0, PDFLib peut se révéler très instable. Le paramètre d'encodage `pdf_set_font` est devenu une chaîne. Cela signifique notamment qu'il faut remplacer 4 par 'winansi'.

Si vous utilisez PDFLib 2.30, pdf_set_text_matrix() a disparu. Elle n'est plus supporté. En général, il

est recommandé de consulter les notes de version de la PDFLib pour lister toutes les modifications.

A partir du 9 mars 2000, PHP 4 ne supporte plus que la version 3.0 et plus récente de PDFLib. PHP 3, par contre, ne doit pas être utilisé avec des versions plus récentes que la 2.01.

6.77.5 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

```
<?php
$pdf = pdf_new();
pdf_open_file($pdf, "test.pdf");
pdf_set_info($pdf, "Author", "Uwe Steinmann");
pdf_set_info($pdf, "Title", "Test for PHP wrapper of PDFLib 2.0");
pdf_set_info($pdf, "Creator", "See Author");
pdf_set_info($pdf, "Subject", "Testing");
pdf_begin_page($pdf, 595, 842);
pdf_add_outline($pdf, "Page 1");
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>finished</A>";
?>
```

La distribution PDFLib contient un exemple plus complexe, qui crée des pages plus élaborées, avec une horloge. Cet exemple a été converti en script PHP (vous retrouverez cet exemple dans le module clibpdf). Il utilise les possibilités de création de fichier en mémoire, sans fichier temporaire.

Exemple pdfclock issue de la distribution 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);
    /* indicateurs de 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($pdf);
    /* indicateurs de 5 minutes */
    pdf_setlinewidth($pdf, 3.0);
    for ($alpha = 0; $alpha < 360; $alpha += 30) {
        pdf_rotate($pdf, 30.0);
        pdf_moveto($pdf, $radius, 0.0);
        pdf_lineto($pdf, $radius-$margin, 0.0);
        pdf_stroke($pdf);
    }
    $ltime = getdate();
    /* aiguille des heures */
    pdf_save($pdf);
    pdf_rotate($pdf,-(($ltime['minutes']/60.0)+$ltime['hours']-3.0)*30.0);
    pdf_moveto($pdf, -$radius/10, -$radius/20);
    pdf_lineto($pdf, $radius/2, 0.0);
    pdf_lineto($pdf, -$radius/10, $radius/20);
    pdf_closepath($pdf);
    pdf_fill($pdf);
    pdf_restore($pdf);
    /* aiguille des minutes */
    pdf_save($pdf);
    pdf_rotate($pdf,-(($ltime['seconds']/60.0)+$ltime['minutes']-15.0)*6.0);
    pdf_moveto($pdf, -$radius/10, -$radius/20);
    pdf_lineto($pdf, $radius * 0.8, 0.0);
    pdf_lineto($pdf, -$radius/10, $radius/20);
    pdf_closepath($pdf);
    pdf_fill($pdf);
    pdf_restore($pdf);
    /* aiguille des secondes */
    pdf_setrgbcolor($pdf, 1.0, 0.0, 0.0);
    pdf_setlinewidth($pdf, 2);
    pdf_save($pdf);
    pdf_rotate($pdf, -(($ltime['seconds'] - 15.0) * 6.0));
    pdf_moveto($pdf, -$radius/5, 0.0);
    pdf_lineto($pdf, $radius, 0.0);
    pdf_stroke($pdf);
    pdf_restore($pdf);
    /* petit cercle au centre */
    pdf_circle($pdf, 0, 0, $radius/30);
    pdf_fill($pdf);
    pdf_restore($pdf);
    pdf_end_page($pdf);
    # pour voir une 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);
?>
```

- Confusion entre les vieilles versions de PDFLib
- Conseils pour installer PDFLib 3.x
- Choix de la version de PDFlib
- Installation des anciennes versions de PDFLib
- Exemples
- pdf_add_annotation
- pdf_add_bookmark
- pdf_add_launchlink
- pdf_add_locallink
- pdf_add_note
- pdf_add_outline
- pdf_add_pdflink
- pdf_add_weblink
- pdf_arc
- pdf_attach_file
- pdf_begin_page
- pdf_circle
- pdf_clip
- pdf_close_image
- pdf_close
- pdf_closepath_fill_stroke
- pdf_closepath_stroke
- pdf_closepath
- pdf_concat
- pdf_continue_text
- pdf_curveto
- pdf_delete
- pdf_end_page
- 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_parameter
- pdf_get_value
- pdf_lineto
- 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_png
- pdf_open_tiff
- pdf_open
- pdf_place_image

- 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_leading
- pdf_set_parameter
- pdf_set_text_matrix
- pdf_set_text_pos
- pdf_set_text_rendering
- pdf_set_value
- pdf_set_word_spacing
- pdf_setdash
- pdf_setflat
- pdf_setfont
- pdf_setgray_fill
- pdf_setgray_stroke
- pdf_setgray
- pdf_setlinecap
- pdf_setlinejoin
- pdf_setlinewidth
- 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

6.77.7 pdf_add_bookmark() : Ajoute un signet à la page courante

`int` **pdf_add_bookmark** (`resource pdf_object` , `string text` , *`int parent`* , *`int open`*)
[lien librairie]

`pdf_add_bookmark` ajoute un signet sous le père `parent` , ou un signet général, si `parent` vaut 0. `pdf_add_bookmark` retourne un descripteur de signet, qui peut être utilisé comme père d'un autre signet. Si `open` vaut 1, les signets fils seront ouverts. Ils seront fermés sinon.

6.77.8 pdf_add_launchlink() : Ajoute une annotation éxécutable dans la page courante

`int` **pdf_add_launchlink** (`resource pdf_object` , `double llx` , `double lly` , `double urx` , `double ury` , `string filename`)
[lien librairie]

`pdf_add_launchlink` ajoute une annotation exécutable (le fichier de destination peut etre n'importe quel fichier).

6.77.9 pdf_add_locallink() : Ajoute un lien sur une annotation dans la page courante

`int` **pdf_add_locallink** (`resource pdf_object` , `double llx` , `double lly` , `double urx` , `double ury` , `int page` , `string dest`)
[lien librairie]

Add a link annotation to a target within the current PDF file.

6.77.10 pdf_add_note() : Ajoute une note d'annotation dans la page courante

`int` **pdf_add_note** (`resource pdf_object` , `double llx` , `double lly` , `double urx` , `double ury` , `string contents` , `string title` , `string icon` , `int open`)
[lien librairie]

`pdf_add_note` ajoute une note d'annotation. `icon` peut prendre une des valeurs suivantes : "comment" (commentaire), "insert" (insertion), "note" (note), "paragraph" (paragraphe), "newparagraph" (nouveau paragraphe), "key" (cle), ou "help" (aide).

6.77.11 pdf_add_outline() : Obsolète: Ajoute un signet dans la page courante

Obsolète.

[lien librairie]

Voir aussi `pdf_add_bookmark` .

6.77.12 pdf_add_pdflink() : Ajoute un lien sur un fichier dans la page courante

`int` **pdf_add_pdflink** (`resource pdf_object` , `double llx` , `double lly` , `double urx` , `double ury` , `string filename` , `int page` , `string dest`)
[lien librairie]

`pdf_add_pdflink` ajoute un lien vers un fichier PDF, sous forme d'annotation.

6.77.13 pdf_add_weblink() : Ajoute un lien hypertexte dans la page courante

`int` **pdf_add_weblink** (`resource pdf_object` , `double llx` , `double lly` , `double urx` , `double ury` , `string url`)
[lien librairie]

`pdf_add_weblink` ajoute un lien hypertexte vers une URL sur le web.

6.77.14 pdf_arc() : Dessine un arc.

`void` **pdf_arc** (`resource pdf_object` , `double x-coor` , `double y-coor` , `double radius` , `double start` , `double end`)
[lien librairie]

`pdf_arc` dessine un arc de cercle, de centre (`x-coor` , `y-coor`) et de rayon `radius` , en commencant à l'angle `start` et finissant à l'angle `end` .

Voir aussi `pdf_circle` , `pdf_stroke` .

6.77.15 pdf_attach_file() : Attache un fichier à la page courante

`int` **pdf_attach_file** (`resource pdf_object` , `double llx` , `double lly` , `double urx` , `double ury` , `string filename` , `string description` , `string author` , `string mimetype` , `string icon`)
[lien librairie]

`pdf_attach_file` attache un fichier à la page courante. `icon` peut prendre l'une des valeurs suivantes : "graph" (image), "paperclip" (texte), "pushpin" (??? NdTraducteur : mailez moi!) ou "tag" (étiquette).

6.77.16 pdf_begin_page() : Commence une nouvelle page.

`void` **pdf_begin_page** (`resource pdf_object` , `double width` , `double height`)
[lien librairie]

`pdf_begin_page` commence une nouvelle page avec la taille `height` et la largueur `width` . Afin de créer un nouveau document valide, vous devez appeler cette fonction et `pdf_end_page` au moins une fois.

Voir aussi `pdf_end_page` .

6.77.17 pdf_circle() : Dessine un cercle.

`void` **pdf_circle** (`resource pdf_object` , `double x-coor` , `double y-coor` , `double radius`)
[lien librairie]

`pdf_circle` dessine un cercle de centre (`x-coor` , `y-coor`), et de rayon `radius` .

Voir aussi pdf_arc , pdf_stroke .

6.77.18 pdf_clip() : Aligne sur le chemin courant.

void **pdf_clip** (resource pdf_object)
[lien librairie]

pdf_clip aligne tous les dessins sur le chemin courant.

6.77.19 pdf_close_image() : Ferme une image

void **pdf_close_image** (resource image)
[lien librairie]

pdf_close_image ferme une image qui a été ouverte par pdf_open_gif ou pdf_open_jpeg .

Voir aussi pdf_open_jpeg , pdf_open_gif , pdf_open_tiff et pdf_open_memory_image .

6.77.20 pdf_close() : Ferme un document PDF.

void **pdf_close** (resource pdf_object)
[lien librairie]

pdf_close ferme un document PDF.

Voir aussi pdf_open , fclose .

6.77.21 pdf_closepath_fill_stroke() : Remplis, dessine et ferme le chemin courant.

void **pdf_closepath_fill_stroke** (resource pdf_object)
[lien librairie]

pdf_closepath_fill_stroke clos le chemin, le remplis avec la couleur courante, et dessine le chemin.

Voir aussi pdf_closepath , pdf_stroke , pdf_fill , pdf_setgray_fill , pdf_setgray , pdf_setrgbcolor_fill , pdf_setrgbcolor .

6.77.22 pdf_closepath_stroke() : Ferme le chemin et dessine le long du chemin.

void **pdf_closepath_stroke** (resource pdf_object)
[lien librairie]

pdf_closepath_stroke est une combinaison de pdf_closepath et pdf_stroke . Elle ferme aussi le chemin.

Voir aussi pdf_closepath , pdf_stroke .

6.77.23 pdf_closepath() : Ferme et clos le chemin.

void **pdf_closepath** (resource pdf_object)
[lien librairie]

pdf_closepath ferme et clos le chemin courant. Cela signifie qu'une ligne va être ajoutée entre le point courant et le premier point du chemin. De nombreuses fonctions telles que pdf_moveto , pdf_circle et pdf_rect démarre un nouveau chemin.

6.77.24 pdf_concat() : Concatène une matrice avec CTM

void **pdf_concat** (resource pdf_object , double a , double b , double c , double d , double e , double f)
[lien librairie]

pdf_concat concatène une matrice avec CTM.

6.77.25 pdf_continue_text() : Affiche un texte sur une nouvelle ligne.

void **pdf_continue_text** (resource pdf_object , string text)
[lien librairie]

pdf_continue_text affiche le texte text sur une nouvelle ligne. La distance entre les lignes peut être choisie avec pdf_set_leading .

Voir aussi pdf_show_xy , pdf_set_leading et pdf_set_text_pos .

6.77.26 pdf_curveto() : Dessine une courbe.

void **pdf_curveto** (resource pdf_object , double x1 , double y1 , double x2 , double y2 , double x3 , double y3)
[lien librairie]

pdf_curveto dessine une courbe de Bézier entre le point courant et le point (x3 , y3) en utilisant les points de contrôle (x1 , y1) et (x2 , y2).

Voir aussi pdf_moveto , pdf_lineto et pdf_stroke .

6.77.27 pdf_delete() : Efface un objet PDF

void **pdf_delete** (resource pdf_object)
[lien librairie]

pdf_delete efface un objet PDF et libère les ressources.

6.77.28 pdf_end_page() : Termine une page.

`void` **pdf_end_page** (`resource pdf_object`)
[lien librairie]

`pdf_end_page` termine une page. Une fois qu'une page a été fermée, elle ne peut pas être modifiée.

Voir aussi `pdf_begin_page` .

6.77.29 pdf_endpath() : Ferme le chemin courant

`void` **pdf_endpath** (`resource pdf_object`)
[lien librairie]

`pdf_endpath` ferme le chemin courant mais ne le clôt pas.

Voir aussi `pdf_closepath` .

6.77.30 pdf_fill_stroke() : Remplis et dessine le chemin courant.

`void` **pdf_fill_stroke** (`resource pdf_object`)
[lien librairie]

`pdf_fill_stroke` remplis l'intérieur du chemin courant avec la couleur courante, puis dessine le chemin courant.

Voir aussi `pdf_closepath` , `pdf_stroke` , `pdf_fill` , `pdf_setgray_fill` , `pdf_setgray` , `pdf_setrgbcolor_fill` , `pdf_setrgbcolor` .

6.77.31 pdf_fill() : Remplis le chemin courant.

`void` **pdf_fill** (`resource pdf_object`)
[lien librairie]

`pdf_fill` remplis l'intérieur du chemin courant avec la couleur courante.

Voir aussi `pdf_closepath` , `pdf_stroke` , `pdf_setgray_fill` , `pdf_setgray` , `pdf_setrgbcolor_fill` , `pdf_setrgbcolor` .

6.77.32 pdf_findfont() : Prépare une police pour utilisation ultérieure

`void` **pdf_findfont** (`resource pdf_object` , `string fontname` , `string encoding` , `int embed`)
[lien librairie]

`pdf_findfont` prépare une police pour utilisation ultérieure avec `pdf_setfont` . Les dimensions seront chargées, et lorsque c'est possible, le fichier de police sera vérifié, mais pas utilisé. `encoding` peut prendre l'une des valeurs suivantes : "builtin" (intégrée), "macroman", "winansi",

"host", et nom d'encodage utilisateur, ou encore nom de CMap.

6.77.33 pdf_get_buffer() : Lit un buffer contenant des données PDF

`string` **pdf_get_buffer** (`resource pdf_object`)
[lien librairie]

`pdf_get_buffer` lit le contenu du buffer `pdf_object` . Le résultat doit être utilisé par le client avant d'appeler toute autre fonction PDFLib.

6.77.34 pdf_get_font() : Obsolète : gestion de police

Obsolète.

[lien librairie]

Voir aussi `pdf_get_value` .

6.77.35 pdf_get_fontname() : Obsolète : gestion de nom de police

Obsolète.

[lien librairie]

Voir aussi `pdf_get_parameter` .

6.77.36 pdf_get_fontsize() : Obsolète : gestion de taille de police

Obsolète.

[lien librairie]

Voir aussi `pdf_get_value` .

6.77.37 pdf_get_image_height() : Retourne la hauteur d'une image

`string` **pdf_get_image_height** (`resource pdf_object` , `resource image`)
[lien librairie]

`pdf_get_image_height` retourne la hauteur de l'image `image` en pixels.

Voir aussi `pdf_open_image_file` , `pdf_open_memory_image` et `pdf_get_image_width` .

6.77.38 pdf_get_image_width() : Retourne la largeur d'une image

`string` **pdf_get_image_width** (`resource pdf_object` , `resource image`)
[lien librairie]

`pdf_get_image_width` retourne la largeur de l'image `image` , en pixels.

Voir aussi `pdf_open_image_file` , `pdf_open_memory_image` et `pdf_get_image_height` .

6.77.39 pdf_get_parameter() : Lit la valeur d'un paramètre PDFLib chaîne

`string` **pdf_get_parameter** (`resource pdf_object` , `string key` , *`double modifier`*)
[lien librairie]

`pdf_get_parameter` lit le contenu de certains paramètres PDFLib, au format chaîne de caractères.

6.77.40 pdf_get_value() : Lit la valeur d'un paramètre PDFLib numérique

`double` **pdf_get_value** (`resource pdf_object` , `string key` , *`double modifier`*)
[lien librairie]

`pdf_get_value` lit le contenu de certains paramètres PDFLib, au format numérique.

6.77.41 pdf_lineto() : Dessine une ligne.

`void` **pdf_lineto** (`resource pdf_object` , `double x−coor` , `double y−coor`)
[lien librairie]

`pdf_lineto` dessine une ligne entre le point courant et le point de coordonnées (`x−coor` , `y−coor`).

Voir aussi `pdf_moveto` , `pdf_curveto` et `pdf_stroke` .

6.77.42 pdf_moveto() : Déplace le point courant.

`void` **pdf_moveto** (`resource pdf_object` , `double x−coor` , `double y−coor`)
[lien librairie]

`pdf_moveto` déplace le point courant à la position (`x−coor` , `y−coor`).

6.77.43 pdf_new() : Crée un nouvel objet PDF

`resource` **pdf_new** (`void`)
[lien librairie]

`pdf_new` crée un nouvel objet PDF, avec gestion des erreurs et de la mémoire par défaut.

6.77.44 pdf_open_CCITT() : Ouvre une nouvelle image à partir de données CCITT

`int` **pdf_open_ccitt** (`resource pdf_object` , `string filename` , `int width` , `int height` , `int BitReverse` , `int k` , `int Blackls1`)
[lien librairie]

`pdf_open_ccitt` ouvre une image CCITT.

6.77.45 pdf_open_file() : Ouvre un nouvel objet PDF

`int` **pdf_open_file** (`resource pdf_object` , *`string filename`*)
[lien librairie]

`pdf_open_file` crée un nouvel objet PDF à partir du fichier `filename` . Si `filename` est vide, le fichier PDF sera généré en mémoire. Le résultat devrat être lu avec la fonction `pdf_get_buffer` function.

L'exemple suivant montre comment créer un fichier PDF en mémoire, et l'envoyer correctement au navigateur.

Création d'un fichier PDF en mémoire

```
<?php
$pdf = pdf_new();
pdf_open_file($pdf);
pdf_begin_page($pdf, 595, 842);
pdf_set_font($pdf, "Times-Roman", 30, "host");
pdf_set_value($pdf, "textrendering", 1);
pdf_show_xy($pdf, "Un document PDF créé en memoire!", 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;
?>
```

6.77.46 pdf_open_gif() : Obsolète: Ouvre une image GIF

Obsolète.

[lien librairie]

Voir aussi `pdf_open_image` ,

6.77.47 pdf_open_image_file() : Lit une image depuis un fichier

`int` **pdf_open_image_file** (`resource pdf_object` , `string imagetype` , `string filename` , *`string stringparam`* , *`string intparam`*)

[lien librairie]

`pdf_open_image_file` ouvre une image dans le fichier `filename` . `imagetype` peut prendre une des valeurs suivantes : "jpeg", "tiff", "gif", et "png". `stringparam` peut prendre l'une des valeurs suivantes : "", "mask", "masked", ou "page". `intparam` peut valoir 0, l'id de l'image du masque appliqué, ou la page.

6.77.48 pdf_open_image() : Fonction générique pour les images

`int` **pdf_open_image** (`int PDF-document` , `string imagetype` , `string source` , `string data` , `long length` , `int width` , `int height` , `int components` , `int bpc` , `string params`)
[lien librairie]

`pdf_open_image` ouvre des fichiers de divers formats d'images. `imagetype` peut prendre l'une des valeurs suivantes : "jpeg", "ccitt", "raw". `source` peut prendre l'une des valeurs suivantes : "memory" (mémoire), "fileref" (pointeur de fichier), "url". `length` ne sert que pour le type "raw"; `params` ne sert qur pour le type "ccitt".

6.77.49 pdf_open_jpeg() : Obsolète: Ouvre une image JPEG

Obsolète.

[lien librairie]

Voir aussi `pdf_open_image` .

6.77.50 pdf_open_memory_image() : Ouvre une image créée par les fonctions images PHP.

`resource` **pdf_open_memory_image** (`resource pdf_object` , `resource image`)
[lien librairie]

`pdf_open_memory_image` prend comme argument une image créée avec les fonctions PHP, et la rend disponible pour le document PDF. La fonction retourne un identifiant PDF d'image.

Inclusion d'une image mémoire

```
<?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` , `pdf_open_jpeg` , `pdf_open_gif` et `pdf_place_image` .

6.77.51 pdf_open_png() : Obsolète: Ouvre une image PNG

Obsolète.

[lien librairie]

Voir aussi `pdf_open_image` ,

6.77.52 pdf_open_tiff() : Obsolète: Ouvre une image TIFF

`int` **pdf_open_tiff** (`int PDF-document` , `string filename`)
[lien librairie]

Obsolète.

Voir aussi `pdf_open_image` .

6.77.53 pdf_open() : Obsolète: Ouvre un nouvel objet PDF

`pdf_open` est obsolète. Utilisez `pdf_new` puis `pdf_open_file` .

[lien librairie]

Voir aussi `pdf_new` et `pdf_open_file` .

6.77.54 pdf_place_image() : Place une image dans la page.

`void` **pdf_place_image** (`resource pdf_object` , `resource image` , `double x-coor` , `double y-coor` , `double scale`)
[lien librairie]

`pdf_place_image` place l'image `image` dans la page courante, à la position (`x-coor` , `x-coor`). L'image peut changer d'échelle simultanémement.

6.77.55 pdf_rect() : Dessine un rectangle.

`void` **pdf_rect** (`resource pdf_object` , `double x-coor` , `double y-coor` , `double width` , `double height`)
[lien librairie]

`pdf_rect` dessine un rectangle un rectangle de coin inférieur gauche de coordonnées (`x-coor` , `y-coor`). Sa longueur vaut `width` . Et sa largeur `height` .

Voir aussi `pdf_stroke` .

6.77.56 pdf_restore() : Restaure un environnement sauvé.

`void` **pdf_restore** (`resource pdf_object`)
[lien librairie]

`pdf_restore` restaure un environnement sauvé par `pdf_save` . Cela fonctionne de manière identique à la commande Postscript grestore. Très pratique lorsque vous vous faire des translations ou des rotations sans affecter les autres objets.

Sauver et restaurer un environnement PDF

```
<?php
pdf_save($pdf);
// tout un lot de rotations, translations, transformations...
pdf_restore($pdf)
?>
```

Voir aussi `pdf_save` .

6.77.57 pdf_rotate() : Choisi la rotation.

`void` **pdf_rotate** (`resource pdf_object` , `double angle`)
[lien librairie]

`pdf_rotate` modifie la rotation de `angle` degré.

6.77.58 pdf_save() : Enregistre l'environnement courant.

`void` **pdf_save** (`resource pdf_object`)
[lien librairie]

`pdf_save` enregistre l'environnement courant. Le fonctionnement est identique à la commande postscript gsave. Très pratique si vous voulez faire une translation ou une rotation d'un objet, sans affecter les autres. `pdf_save` sera toujours suivi d'un `pdf_restore` .

Voir aussi `pdf_restore` .

6.77.59 pdf_scale() : Modifie l'échelle.

`void` **pdf_scale** (`resource pdf_object` , `double x-scale` , `double y-scale`)
[lien librairie]

`pdf_scale` modifie l'échelle dans les deux directions. L'exemple suivant multiplie l'échelle par 72. La ligne suivante sera dessinée sur un pouce (2.54 cm) de large.

Mise à l'échelle

```
<?php pdf_scale($pdf, 72.0, 72.0);
pdf_lineto($pdf, 1, 1);
pdf_stroke($pdf);
?>
```

6.77.60 pdf_set_border_color() : Modifie la couleur des liens et annotations

`void` **pdf_set_border_color** (`resource pdf_object` , `double red` , `double green` , `double blue`)
[lien librairie]

`pdf_set_border_color` modifie la couleur des bords de liens et d'annotation. Les trois composants `red` , `green` , `blue` représentent une couleur RGB (rouge, vert, bleu) et leur valeur doivent être comprise entre 0 et 1.

Voir aussi `pdf_set_border_style` et `pdf_set_border_dash` .

6.77.61 pdf_set_border_dash() : Modifie les pointillés des liens et annotations

`void` **pdf_set_border_dash** (`resource pdf_object` , `double black` , `double white`)
[lien librairie]

`pdf_set_border_dash` modifie la longueur des pointillés (si le style de bord d'une annotation est en pointillés). `black` représente la taille des traits noirs, et `white` celle des espaces blancs.

Voir aussi `pdf_set_border_style` et `pdf_set_border_color` .

6.77.62 pdf_set_border_style() : Modifie le bord des liens et annotations

`void` **pdf_set_border_style** (`resource pdf_object` , `string style` , `double width`)
[lien librairie]

`pdf_set_border_style` modifie le style des bords de liens et d'annotation. `style` peut valoir 'solid' (trait plain) ou 'dashed'(pointillé).

Voir aussi `pdf_set_border_color` , `pdf_set_border_dash` .

6.77.63 pdf_set_char_spacing() : Fixe l'espacement des caractères.

`void` **pdf_set_char_spacing** (`resource pdf_object` , `double space`)
[lien librairie]

`pdf_set_char_spacing` modifie l'espacement des caractères.

Voir aussi `pdf_set_word_spacing` et `pdf_set_leading` .

6.77.64 pdf_set_duration() : Choisi la durée de transition entre deux pages.

`void` **pdf_set_duration** (`resource pdf_object` , `double duration`)
[lien librairie]

`pdf_set_duration` choisi la durée de transition, en secondes, entre deux pages.

6.77.65 pdf_set_font() : Sélectionne la police et sa taille.

`void` **pdf_set_font** (`resource pdf_object` , `string font name` , `double size` , `string encoding` , *`int embed`*)
[lien librairie]

`pdf_set_font` sélectionne la police, sa taille et son encodage. Il vous faudra fournir des fichiers Adobe Font Metrics (afm) comme police, dans le dossier de police (par défaut ./fonts). Si vous utilisez PDFLib 0.6, vous devrez fournir des fichiers Adobe Font Métric (afm−files) pour les polices, dans le chemin de police (par défaut, ./fonts). Si vous utilisez php versin 3 ou une version plus ancienne que la version 2.20 de PDFLib, le quatrième paramètre `encoding` peut prendre les valeurs suivantes : 0 = builtin, 1 = pdfdoc, 2 = macroman, 3 = macexpert, 4 = winansi. Un encodage plus grand que 4 et inférieur à 0 sera transformé en 'winansi'. 'winansi' est souvent un bon choix. Si vous utilisez PHP version 4 et une version plus ancienne que la version 2.20 de PDFLib le quatrième paramètre `encoding` est une chaîne : 'builtin', 'pdfdoc', 'macroman', 'macexpert', 'winansi'. Si le dernier paramètre est à 1, la police est intégrée dans le document. Sinon, elle ne le sera pas. Incorporer une police dans un document est un bonne idée si la police n'est pas répandue, ou si vous ne pouvez pas vous assurez que le la personne qui regardera votre document peut accéder à cette police.

Note
`pdf_set_font` doit être appelée après `pdf_begin_page` pour créer un document PDF valide.

Note
Si vous référencez une police dans un fichier `.upr` , assurez−vous que le nom du fichier `.afm` et celui de la police sont bien les mêmes. Sinon, la police sera aggrandie plusieur fois (Merci à Paul Haddon pour cette info).

6.77.66 pdf_set_horiz_scaling() : Fixe l'echelle horizontale du texte.

`void` **pdf_set_horiz_scaling** (`resource pdf_object` , `double scale`)
[lien librairie]

`pdf_set_horiz_scaling` fixe l'échelle horizontale du texte, à `scale` en pourcentage.

6.77.67 pdf_set_info() : Remplis les entêtes du document

`void` **pdf_set_info** (`resource pdf_object` , `string fieldname` , `string value`)
[lien librairie]

`pdf_set_info` modifie un champs d'entête d'un document PDF. Les valeurs possibles pour

fieldname sont : 'Subject' (sujet), 'Title'(titre), 'Creator'(créateur), 'Author'(auteur), 'Keywords'(mots−clé) et un autre nom, défini par l'utilisateur. pdf_set_info peut être appelée avant la création d'une page.

Préparer l'entête d'un document PDF

```
<?php
$fd = fopen("test.pdf", "w");
$pdfdoc = pdf_open($fd);
pdf_set_info($pdfdoc, "Author", "Uwe Steinmann");
pdf_set_info($pdfdoc, "Creator", "Uwe Steinmann");
pdf_set_info($pdfdoc, "Title", "Testing Info Fields");
pdf_set_info($pdfdoc, "Subject", "Test");
pdf_set_info($pdfdoc, "Keywords", "Test, Fields");
pdf_set_info($pdfdoc, "CustomField", "What ever makes sense");
pdf_begin_page($pdfdoc, 595, 842);
pdf_end_page($pdfdoc);
pdf_close($pdfdoc);
?>
```

Note

pdf_set_info remplace pdf_set_info_keywords(), pdf_set_info_title(), pdf_set_info_subject(), pdf_set_info_creator() et pdf_set_info_subject().

6.77.68 pdf_set_leading() : Obsolète : Modifie la distance entre les lignes du texte

Obsolète.

[lien librairie]

Voir aussi pdf_set_value .

6.77.69 pdf_set_parameter() : Modifie certains paramètres.

void **pdf_set_parameter** (resource pdf_object , string name , string value)
[lien librairie]

pdf_set_parameter modifie certaines valeurs de pdglib. value est de type chaîne.

Voir aussi pdf_get_value , pdf_set_value et pdf_get_parameter .

6.77.70 pdf_set_text_matrix() : Obsolète: Modifie la transition des pages

Voir pdf_set_parameter .

[lien librairie]

6.77.71 pdf_set_text_pos() : Fixe la position du texte.

`void` **pdf_set_text_pos** (`resource pdf_object` , `double x-coor` , `double y-coor`)
[lien librairie]

`pdf_set_text_pos` modifie la position du texte qui sera utilisée lors du prochain `pdf_show` .

Voir aussi `pdf_show` et `pdf_show_xy` .

6.77.72 pdf_set_text_rendering() : Détermine le rendu du texte.

`void` **pdf_set_text_rendering** (`resource pdf_object` , `int mode`)
[lien librairie]

`pdf_set_text_rendering` détermine le rendu du texte. Les valeurs possibles pour `mode` sont 0=fill text (texte plein), 1=stroke text (???), 2=fill et stroke text (texte plein et stroke), 3=invisible, 4=texte plein, et ajouté au chemin, 5=stroke text, ajouté au chemin, 6=texte plein et stroke, ajouté au chemin, 7=ajouté au chemin.

6.77.73 pdf_set_value() : Modifie certains paramètre numériques

`void` **pdf_set_value** (`resource pdf_object` , `string name` , `double value`)
[lien librairie]

`pdf_set_value` modifie la valeur (numérique) du paramètre `name` de PDFLib.

Voir aussi `pdf_get_value` , `pdf_get_parameter` et `pdf_set_parameter` .

6.77.74 pdf_set_word_spacing() : Fixe l'espacement des mots.

`void` **pdf_set_word_spacing** (`resource pdf_object` , `double space`)
[lien librairie]

`pdf_set_word_spacing` modifie l'espacement des mots.

Voir aussi `pdf_set_char_spacing` et `pdf_set_leading` .

6.77.75 pdf_setdash() : Modifie les caractères de remplissage.

`void` **pdf_setdash** (`resource pdf_object` , `double white` , `double black`)
[lien librairie]

`pdf_setdash` modifie les caractères de remplissage, et affecte `white` comme caractère invisible, et `black` comme caractère de remplissage. Si les deux sont à zéros, une ligne continue est affichée.

6.77.76 pdf_setflat() : Modifie la platitude (flatness).

`void` **pdf_setflat** (`resource pdf_object` , `double value`)
[lien librairie]

`pdf_setflat` modifie la platitude, et lui affecte la valeur `value` entre 0 et 100.

6.77.77 pdf_setfont() : Modifie la police courante

`void` **pdf_setfont** (`resource pdf_object` , `int font` , `double size`)
[lien librairie]

`pdf_setfont` remplace la police courante par `font` , à la taille `size` . `font` est créé par `pdf_findfont` .

6.77.78 pdf_setgray_fill() : Modifie la couleur grise comme couleur de remplissage.

`void` **pdf_setgray_fill** (`resource pdf_object` , `double gray value`)
[lien librairie]

`pdf_setgray_fill` modifie la couleur grise comme couleur de remplissage.

Voir aussi `pdf_setrgbcolor_fill` .

6.77.79 pdf_setgray_stroke() : Modifie la couleur de dessin à un niveau de gris.

`void` **pdf_setgray_stroke** (`resource pdf_object` , `double gray value`)
[lien librairie]

`pdf_setgray_stroke` modifie la couleur de dessin à un niveau de gris.

Voir aussi `pdf_setrgbcolor_stroke` .

6.77.80 pdf_setgray() : Modifie la couleur grise comme couleur de remplissage et de dessin.

`void` **pdf_setgray** (`resource pdf_object` , `double gray value`)
[lien librairie]

`pdf_setgray` modifie la couleur grise comme couleur de remplissage et de dessin.

Voir aussi `pdf_setrgbcolor_stroke` et `pdf_setrgbcolor_fill` .

6.77.81 pdf_setlinecap() : Modifie le paramètre linecap.

`void` **pdf_setlinecap** (`resource pdf_object` , `int value`)
[lien librairie]

`pdf_setlinecap` affecte au paramètre "linecap" la valeur `value` , entre 0 et 2.

6.77.82 pdf_setlinejoin() : Modifie le paramètre linejoin.

`void` **pdf_setlinejoin** (`resource pdf_object` , `long value`)
[lien librairie]

`pdf_setlinejoin` modifie le paramètre "linejoin", et lui affecte la valeur `value` , entre 0 et 2.

6.77.83 pdf_setlinewidth() : Modifie la largeur de ligne.

`void` **pdf_setlinewidth** (`resource pdf_object` , `double width`)
[lien librairie]

`pdf_setlinewidth` affecte à largeur de ligne la valeur `width` .

6.77.84 pdf_setmiterlimit() : Modifie la "miter limit".

`void` **pdf_setmiterlimit** (`resource pdf_object` , `double value`)
[lien librairie]

`pdf_setmiterlimit` modifie la "miter limit" et lui affecte la valeur `value` , supérieure à 1.

6.77.85 pdf_setpolydash() : Modifie les pointillés compliqués

`void` **pdf_setpolydash** (`resource pdf_object` , `array dasharray`)
[lien librairie]

`pdf_setpolydash` modifie les pointillés complexes, définit par le tableau `dasharray` .

6.77.86 pdf_setrgbcolor_fill() : Modifie la couleur rgb comme couleur de remplissage.

`void` **pdf_setrgbcolor_fill** (`resource pdf_object` , `double red value` , `double green value` , `double blue value`)
[lien librairie]

`pdf_setrgbcolor_fill` choisi la couleur RGB comme couleur de remplissage.

Voir aussi `pdf_setrgbcolor_fill` .

6.77.87 pdf_setrgbcolor_stroke() : Modifie la couleur rgb comme couleur de dessin.

`void` **pdf_setrgbcolor_stroke** (`resource pdf_object` , `double red value` , `double green value` , `double blue value`)
[lien librairie]

`pdf_setrgbcolor_stroke` choisi la couleur `RGB` comme couleur de dessin.

Voir aussi `pdf_setrgbcolor_fill` .

6.77.88 pdf_setrgbcolor() : Modifie la couleur rgb comme couleur de dessin et de remplissage.

`void` **pdf_setrgbcolor** (`resource pdf_object` , `double red value` , `double green value` , `double blue value`)
[lien librairie]

`pdf_setrgbcolor_stroke` modifie la couleur `RGB` comme couleur de remplissage.

Voir aussi `pdf_setrgbcolor_stroke` et `pdf_setrgbcolor_fill` .

6.77.89 pdf_show_boxed() : Affiche un texte dans un rectangle.

`int` **pdf_show_boxed** (`resource pdf_object` , `string text` , `double x-coor` , `double y-coor` , `double width` , `double height` , `string mode` , *`string feature`*)
[lien librairie]

`pdf_show_boxed` affiche le texte `text` dans un rectangle, dont le coin inférieur gauche est aux coordonnées (`x-coor` , `y-coor`). Les dimensions du rectangle sont `height` et `width` . Le paramètre `mode` indique le type de `text` . Si `width` et `height` sont à zéro, le mode `mode` peut être "left" (gauche), "right" (droite) ou "center" (centré). `width` ou `height` sont différents pouvant prendre les valeurs de "justify" (justification) ou "fulljustify" (justification complète).

Si le paramètre `feature` vaut "blind", le texte n'est pas affiché.

`pdf_show_boxed` retourne le nombre de caractères qui n'ont pas pu être traités, car ils ne rentraient pas dans le rectangle.

Voir aussi `pdf_show` et `pdf_show_xy` .

6.77.90 pdf_show_xy() : Affiche un texte à une position donnée.

`void` **pdf_show_xy** (`resource pdf_object` , `string text` , `double x-coor` , `double y-coor`)
[lien librairie]

`pdf_show_xy` affiche le texte `text` à la position donnée par les coordonnées (`x-coor` , `y-coor`).

Voir aussi pdf_show et pdf_show_boxed .

6.77.91 pdf_show() : Affiche un texte à la position courante.

void **pdf_show** (resource pdf_object , string text)
[lien librairie]

pdf_show affiche le texte text avec la position courante, et avec la police courante.

Voir aussi pdf_show_xy , pdf_show_boxed , pdf_set_text_pos et pdf_set_font .

6.77.92 pdf_skew() : Modifie le système de coordonnées.

void **pdf_skew** (resource pdf_object , double alpha , double beta)
[lien librairie]

pdf_skew modifie le système de coordonnées, en faisant une rotation d'angle alpha pour les (x) et d'angle beta pour les (y), en degrés. alpha et beta ne peuvent pas pendre les valeurs de 90 ou 270 degrés.

6.77.93 pdf_stringwidth() : Retourne la largeur du texte avec la police courante.

double **pdf_stringwidth** (resource pdf_object , string text)
[lien librairie]

pdf_stringwidth retourne la largeur du texte text avec la police courante. Il faut qu'une police ait été choisie auparavant.

Voir aussi pdf_set_font .

6.77.94 pdf_stroke() : Dessine le long du chemin.

void **pdf_stroke** (resource pdf_object)
[lien librairie]

pdf_stroke dessine une ligne le long du chemin. Le chemin courant est la somme de toutes les lignes déssinées. Sans cette fonction, la ligne de chemin ne sera pas dessinée.

Voir aussi pdf_closepath , pdf_closepath_stroke .

6.77.95 pdf_translate() : Modifie l'origine du système de coordonnées.

void **pdf_translate** (resource pdf_object , double x−coor , double y−coor)
[lien librairie]

pdf_translate place l'origine du système de coordonnées au point (x−coor , y−coor).
L'exemple suivant trace une ligne de (0, 0) à (200, 200) par rapport aux coordonnées initiales. Il faut

aussi désigner le point courant après `pdf_translate` et avant de commencer à dessiner les objets.

Translation

`<?php pdf_moveto($pdf, 0, 0);`
`pdf_lineto($pdf, 100, 100);`
`pdf_stroke($pdf);`
`pdf_translate($pdf, 100, 100);`
`pdf_moveto($pdf, 0, 0);`
`pdf_lineto($pdf, 100, 100);`
`pdf_stroke($pdf);`
`?>`

6.78 Paiement par Verisign

6.78.1 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.

6.78.2 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.

6.78.3 Installation

Ces fonctions ne sont disponible que si PHP a été compilé avec l'option `--with-pfpro[=DIR]` .

6.78.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
pfpro.defaulthost/PFPRO_VERSION < 3	"test.signio.com"	PHP_INI_ALL
pfpro.defaulthost	"test-payflow.verisign.com"	PHP_INI_ALL
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

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.78.5 Types de ressources

Cette extension ne définit aucune ressource.

6.78.6 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

6.78.8 pfpro_init() : Initialise la librairie Payflow Pro

`void` **pfpro_init** (`void`)
[lien librairie]

`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` .

6.78.9 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`*)
[lien librairie]

`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 argumentest 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 `pfpro_process_raw`

```
<?php
pfpro_init();
$response = pfpro_process("USER=mylogin&PWD[5]=m&ndy&TRXTYPE=S&TENDER=C&AMT=1.50&ACCT=41111111111111
if (!$response) {
  die("Impossible de contacter Verisign.\n");
}
echo "La réponse brute de Verisign est ".$response;
pfpro_cleanup();
?>
```

6.78.10 pfpro_process() : Effectue une transaction avec Payflow Pro

`array` **pfpro_process** (`array parameters` , *`string address`* , *`int port`* , *`int timeout`* , *`string proxy_logon`* , *`int proxy_port`* , *`string proxy_logon`* , *`string proxy_password`*)
[lien librairie]

`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 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();
?>
```

6.78.11 pfpro_version() : Lit le numéro de version de Payflow Pro

`string` **pfpro_version** (`void`)
[lien librairie]

`pfpro_version` lit la version de la librairie Payflow Pro. Au moment de rédaction de la doc, c'était L211.

6.79 Options PHP et informations

6.79.1 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` .

6.79.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.79.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.79.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
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
enable_dl	"1"	PHP_INI_SYSTEM
max_execution_time	"30"	PHP_INI_ALL
magic_quotes_gpc	"1"	PHP_INI_PERDIR|PHP_INI_SYSTEM
magic_quotes_runtime	"0"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

`assert.active` *`Array`*
: Active les évaluations de type `assert` .

`assert.bail` *`Array`*
: Termine le script si une assertion échoue.

`assert.warning` *`Array`*
: *f*met une alerte PHP pour chaque assertion échouée.

`assert.callback` *`string`*
: Fonction définie par le programmeur, à appeler pour chaque assertion échouée.

`assert.quiet_eval` *`Array`*
: 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` Array
: 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 il est toujours impossible d'utiliser la fonction `dl` .

`max_execution_time` integer
: Fixe le temps maximal d'éxécution 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.

`magic_quotes_gpc` Array
: 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.

`magic_quotes_runtime` Array
: 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.

6.79.5 Types de ressources

Cette extension ne définit aucune ressource.

6.79.6 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 varibles d'environnement, qui sont aussi disponibles dans `$_ENV` .
INFO_VARIABLES	32	Toutes les variables prédéfinies : EGPCS (Environnement, GET, POST, Cookie, Server).
INFO_LICENSE	64	La licence PHP. Voir aussi la FAQ de la license.
INFO_ALL	−1	Affiche toutes les valeurs citées ci−dessus. C'est la valeur par défaut.

`ASSERT_ACTIVE` (nombre ˆ virgule flottante)
`ASSERT_CALLBACK` (nombre ˆ virgule flottante)
`ASSERT_BAIL` (nombre ˆ virgule flottante)
`ASSERT_WARNING` (nombre ˆ virgule flottante)
`ASSERT_QUIET_EVAL` (nombre ˆ virgule flottante)

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- assert−options
- assert
- dl
- extension_loaded
- get_cfg_var
- get_current_user
- get_defined_constants
- get_extension_funcs
- 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
- php_ini_scanned_files
- php_logo_guid
- php_sapi_name
- php_uname
- phpcredits
- phpinfo
- phpversion
- putenv
- set_magic_quotes_runtime
- set_time_limit
- version_compare
- zend_logo_guid
- zend_version

6.79.8 assert() : Vérifie si une assertion est fausse

`int` **assert** `( string|boolean assertion )`
[lien librairie]

`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 débuggage. 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.

6.79.9 dl() : Charge une extension PHP à la volée

`int` **dl** `( string library )`
[lien librairie]

`dl` charge l'extension PHP `library` à la volée .

Voir aussi les directives de configuration extension_dir et enable_dl .

6.79.10 extension_loaded() : Détermine si une extension est chargée ou non.

`boolean` **extension_loaded** (`string name`)
[lien librairie]

`extension_loaded` retourne `TRUE` si l'extension `name` a été chargée. Vous pouvez voir les différents noms des extensions, en utilisant la fonction `phpinfo` .

Voir aussi `phpinfo` .

Note
Cette fonction a été ajoutée dans 3.0.10.

6.79.11 get_cfg_var() : Retourne la valeur d'une option de PHP

`string` **get_cfg_var** (`string varname`)
[lien librairie]

`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.

6.79.12 get_current_user() : Retourne le nom du possesseur du script courant.

`string` **get_current_user** (`void`)
[lien librairie]

`get_current_user` retourne le nom du possesseur du script courant.

Voir aussi `getmyuid` , `getmypid` , `getmyinode` et `getlastmod` .

6.79.13 get_defined_constants() : Retourne la liste des constantes et leur valeur

`array` **get_defined_constants** (`void`)
[lien librairie]

`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` .

Par exemple :

```
<?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` .

6.79.14 get_extension_funcs() : Liste les fonctions d'une extension

`array` **get_extension_funcs** (`string module_name`)
[lien librairie]

`get_extension_funcs` retourne le nom des fonctions définies dans le module `module_name` .

Par exemple, les lignes suivantes :

```
<?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` .

6.79.15 get_included_files() : Retourne un tableau avec les noms des fichiers qui sont inclus dans un script

`array` **get_included_files** (`void`)
[lien librairie]

`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 `require_once` , `include_once` et `get_required_files` .

6.79.16 get_loaded_extensions() : Retourne la liste de tous les modules compilés et chargés

`array` **get_loaded_extensions** (`void`)
[lien librairie]

`get_loaded_extensions` retourne un tableau contenant les noms de tous les modules compilés et chargés sur l'interpreteur PHP courant.

Par exemple, la ligne ci−dessous

```
<?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` .

6.79.17 get_magic_quotes_gpc() : Retourne la configuration actuelle de l'option magic_quotes_gpc.

`long` **get_magic_quotes_gpc** (`void`)
[lien librairie]

`get_magic_quotes_gpc` retourne la configuration actuelle de l'option magic_quotes_gpc (0 pour l'option désactivée, 1 pour l'option activée).

Voir aussi `get_magic_quotes_runtime` et `set_magic_quotes_runtime` .

6.79.18 get_magic_quotes_runtime() : Retourne la configuration actuelle de l'option magic_quotes_runtime.

`long` **get_magic_quotes_runtime** (`void`)
[lien librairie]

`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` .

6.79.19 get_required_files() : Retourne un tableau avec les noms des fichiers qui sont requis et inclus dans un script

`array` **get_required_files** (`void`)
[lien librairie]

`get_required_files` retourne un tableau contenant les noms de tous les fichiers qui ont été chargés dans un script avec la fonction `require_once` ou `include_once` . Les index de ces tableaux sont les noms des fichiers utilisés dans les fonctions `require_once` ou `include_once` .

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` Depuis PHP 4.0.4, cette fonction est un alias de `get_included_files`

Voir aussi `require_once` , `include_once` et `get_included_files`

6.79.20 getenv() : Retourne la valeur de la variable d'environnement.

`string` **getenv** (`string varname`)
[lien librairie]

`getenv` retourne la valeur de la variable d'environnement `varname` , ou FALSE en cas d'erreur.

```
<?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 particulièment la page concernant les variables d'environnement. .

Voir aussi `putenv` .

6.79.21 getlastmod() : Retourne la date de dernière modification de la page.

`int` **getlastmod** (`void`)
[lien librairie]

`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` , `get_current_user` , `getmyinode` et `getmypid` .

6.79.22 getmygid() : Retourne le GID du propriétaire du script

`int` **getmygid** (`void`)
[lien librairie]

`getmygid` retourne le GID du propriétaire du script, ou FALSE en cas d'erreur.

Voir aussi `getmyuid` , `getmypid` , `get_current_user` , `getmyinode` et `getlastmod` .

6.79.23 getmyinode() : Retourne l'inode du script.

`int` **getmyinode** (`void`)
[lien librairie]

`getmyinode` retourne l'inode du script, ou FALSE en cas d'erreur.

Voir aussi `getmyuid` , `get_current_user` , `getmypid` et `getlastmod` .

Note
`getmyinode` est inopérante sur les systèmes Windows.

6.79.24 getmypid() : Retourne le numéro de processus courant.

`int` **getmypid** (`void`)
[lien librairie]

`getmypid` retourne le numéro de processus actuel ou FALSE en cas d'erreur.

Il est à noter que si vous utilisez PHP comme module Apache, il n'est pas garanti que deux invocations distinctes de la fonction donnent des résultats différents.

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` , `get_current_user` , `getmyinode` et `getlastmod` .

6.79.25 getmyuid() : Retourne l'UID du propriétaire du script actuel.

`int` **getmyuid** (`void`)
[lien librairie]

`getmyuid` retourne l'UID du propriétaire du script actuel ou FALSE en cas d'erreur.

Voir aussi `getmypid` , `get_current_user` , `getmyinode` et `getlastmod` .

6.79.26 getopt() : Lit des options passés dans la ligne de commande

`string` **getopt** (`string options`)
[lien librairie]

`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 peut si un argument a des espaces auparavant.

`getopt` retourne un tableau avec les options et leur valeur. Si une option n'a pas d'argument, sa valeur prendra FALSE .

Note

`getopt` requiert la directive register_argc_argv activée.

6.79.27 getrusage() : Retourne le niveau d'utilisation des ressources.

`array` **getrusage** (*`int who`*)
[lien librairie]

`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_utime.tv_sec"]; // Temps utilisateur (en secondes)
  echo $dat["ru_utime.tv_usec"];// Temps utilisateur (en microsecondes)
?>
```

Consultez le manuel "man" pour plus de détails.

6.79.28 ini_alter() : Change la valeur d'une option de configuration

`string` **ini_alter** (`string varname` , `string newvalue`)
[lien librairie]

`ini_alter` change la valeur de l'option de configuration `varname` et lui donne la valeur de `newvalue` . `ini_alter` retourne FALSE en cas d'échec, et la valeur précédente en cas de succès.

Note
`ini_alter` est un alias de `ini_set`

Voir aussi `ini_get` , `ini_restore` et `ini_set`

6.79.29 ini_get_all() : Lit toutes les valeurs de configuration

`array` **ini_get_all** (*`string extension`*)
[lien librairie]

`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.

Voir aussi `ini_alter` , `ini_restore` , `ini_get` et `ini_set`

6.79.30 ini_get() : Lit la valeur d'une option de configuration.

`string` **ini_get** (`string varname`)
[lien librairie]

`ini_get` retourne la valeur de l'option de configuration `varname` en cas de succès, et FALSE .

Voir aussi `ini_alter` , `ini_restore` et `ini_set`

6.79.31 ini_restore() : Restaure la valeur de l'option de configuration

`string` **ini_restore** (`string varname`)
[lien librairie]

`ini_restore` restaure la valeur originale de l'option de configuration `varname` .

Voir aussi `ini_alter` , `ini_get` et `ini_set`

6.79.32 ini_set() : Sets the value of a configuration option

`string` **ini_set** (`string varname` , `string newvalue`)
[lien librairie]

`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
hyerwave.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	JAVALIB	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
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_P
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
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_P
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_P
asp_tags	"0"	PHP_INI_SYSTEM|PHP_INI_P
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_S
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_S
magic_quotes_runtime	"0"	PHP_INI_ALL
magic_quotes_sybase	"0"	PHP_INI_ALL
output_buffering	"0"	PHP_INI_PERDIR|PHP_INI_S
output_handler	NULL	PHP_INI_PERDIR|PHP_INI_S
register_argc_argv	"1"	PHP_INI_PERDIR|PHP_INI_S
register_globals	"0"	PHP_INI_PERDIR|PHP_INI_S
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_P
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_P
auto_append_file	NULL	PHP_INI_SYSTEM|PHP_INI_P
auto_prepend_file	NULL	PHP_INI_SYSTEM|PHP_INI_P
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 block vhost de `httpd.conf`
PHP_INI_ALL	7	La valeur peut être modifiée n'importe où

Voir aussi `ini_alter` , `ini_get` et `ini_restore`

6.79.33 php_ini_scanned_files() : Retourne la liste des fichiers .ini analysédans les dossiers de configuration supplémentaires

`string` **php_ini_scanned_files** (`void`)
[lien librairie]

`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 situé 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 appelerez la fonction `php_ini_scanned_files` .

Les fichiers retournés inclus le chemin, comme déclaré dans la directive

`--with-config-file-scan-dir` . De plus, chaque virgule est suivi 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` .

6.79.34 php_logo_guid() : Retourne le logo

`string` **php_logo_guid** (`void`)
[lien librairie]

Note

Cette fonctionnalité a été ajoutée dans PHP 4 Beta 4.

6.79.35 php_sapi_name() : Retourne le type d'interface utilisé entre le serveur web et PHP

`string` **php_sapi_name** (`void`)
[lien librairie]

`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 `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";
?>
```

6.79.36 php_uname() : Retourne les informations sur le système d'exploitation

`string` **php_uname** (`void`)
[lien librairie]

`php_uname` retourne les informations sur le système d'exploitation sur lequel tourne PHP.

Exemple `php_uname`

```
<?php
if (substr(php_uname(), 0, 7) == "Windows") {
    die("Désolé, ce script ne fonctionne pas sous Windows.\n");
}
?>
```

6.79.37 phpcredits() : Imprime les crédits de PHP.

`void` **phpcredits** (`int flag`)
[lien librairie]

`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 :

```
<?php
  phpcredits(CREDITS_GENERAL);
?>
```

Et pour afficher la liste des développeurs et du groupe de documentation dans une page séparée, vous utiliserez

```
<?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 :

```
<html>
  <head>
    <title>Ma page de crédits</title>
  </head>
  <body>
  <?php
    // Un peu de votre code
    phpcredits(CREDITS_ALL + CREDITS_FULLPAGE);
    // 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 4.0.1pl2.

Voir aussi `version_compare` , `phpinfo` , `phpversion` et `php_logo_guid` .

6.79.38 phpinfo() : Affiche de nombreuses informations sur le PHP.

`int` **phpinfo** (*`int what`*)
[lien librairie]

`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.

Les affichages peuvent être personnalisés en passant une ou plusieurs valeurs parmi les suivantes, comme paramètre optionnel `what` :

- INFO_GENERAL
- INFO_CREDITS
- INFO_CONFIGURATION
- INFO_MODULES
- INFO_ENVIRONMENT
- INFO_VARIABLES
- INFO_LICENSE
- INFO_ALL

Voir aussi `phpversion` , `phpcredits` et `php_logo_guid`

6.79.39 phpversion() : Retourne le numéro de la version courante de PHP.

`string` **phpversion** (`void`)
[lien librairie]

`phpversion` retourne le numéro de la version courante de PHP.

Exemple avec `phpversion`

```php
<?php
// affiche le numéro de version courante du PHP.
  echo "PHP Version: ".phpversion();
?>
```

Voir aussi `phpinfo` .

6.79.40 putenv() : Fixe la valeur d'une variable d'environnement.

`void` **putenv** (`string setting`)
[lien librairie]

`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
<?php
putenv("UNIQID=$uniqid");
?>
```

Voir aussi `getenv` .

6.79.41 set_magic_quotes_runtime() : Active/désactive l'option magic_quotes_runtime.

`long` **set_magic_quotes_runtime** (`int new_setting`)
[lien librairie]

`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` .

6.79.42 set_time_limit() : Fixe le temps maximum d'exécution d'un script.

`void` **set_time_limit** (`int seconds`)
[lien librairie]

`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 fichier de configuration . 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 fichier de configuration .

6.79.43 version_compare() : Compare deux versions de PHP

`int` **version_compare** (`string version1` , `string version2` , *`string operator`*)
[lien librairie]

`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
?>
```

6.79.44 zend_logo_guid() : Retourne le logo de Zend

`string` **zend_logo_guid** (`void`)
[lien librairie]

Note

Cette fonctionnalité a été ajoutée en PHP 4 Beta 4.

6.79.45 zend_version() : Lit la version courante du moteur Zend.

`string` **zend_version** (`void`)
[lien librairie]

`zend_version` retourne une chaîne contenant le numéro de version du moteur d'analyse Zend, pour l'exécutable PHP courant.

Exemple `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` .

6.80 POSIX

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.

- posix_ctermid
- 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_times
- posix_ttyname
- posix_uname

6.80.2 posix_getcwd() : Chemin du dossier courant.

`string` **posix_getcwd** (`void`)
[lien librairie]

Encore à faire très rapidement.

6.80.3 posix_getegid() : Retourne l'ID effectif du groupe du processus courant.

`int` **posix_getegid** (`void`)
[lien librairie]

posix_getegid retourne l'ID effectif du groupe du processus courant. Reportez−vous à posix_getgrgid pour transformer cette information en nom de groupe.

6.80.4 posix_geteuid() : Retourne l'UID effectif de l'utilisateur du processus courant.

int **posix_geteuid** (void)
[lien librairie]

posix_geteuid retourne l'UID effectif de l'utilisateur du processus courant. Reportez−vous à posix_getpwuid pour obtenir le nom d'utilisateur.

6.80.5 posix_getgid() : Retourne l'UID du groupe du processus courant.

int **posix_getgid** (void)
[lien librairie]

posix_getgid retourne l'UID du groupe du processus courant. Reportez−vous à posix_getgrgid pour accéder au nom du groupe.

6.80.6 posix_getgrgid() : Retourne des informations sur un groupe.

array **posix_getgrgid** (int gid)
[lien librairie]

Encore à faire.

6.80.7 posix_getgrnam() : Retourne des informations sur un groupe.

array **posix_getgrnam** (string name)
[lien librairie]

Encore à faire.

6.80.8 posix_getgroups() : Retourne les identifiants du groupe du processus courant.

array **posix_getgroups** (void)
[lien librairie]

posix_getgroups retourne un tableau contenant les identifiants du groupe du processus courant. Reportez−vous à posix_getgrgid pour pouvoir utiliser ces id.

6.80.9 posix_getlogin() : Retourne le nom de login.

string **posix_getlogin** (void)

[lien librairie]

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.

6.80.10 posix_getpgid() : Retourne l'id du groupe de processus.

int **posix_getpgid** (int pid)
[lien librairie]

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 .

6.80.11 posix_getpgrp() : Retourne l'identifiant du groupe de processus.

int **posix_getpgrp** (void)
[lien librairie]

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.

6.80.12 posix_getpid() : Retourne l'identifiant du processus courant.

int **posix_getpid** (void)
[lien librairie]

posix_getpid retourne l'identifiant du processus courant.

6.80.13 posix_getppid() : Retourne l'identifiant du processus parent.

int **posix_getppid** (void)
[lien librairie]

posix_getppid retourne l'identifiant du processus parent du processus courant.

6.80.14 posix_getpwnam() : Retourne des informations sur un utilisateur.

array **posix_getpwnam** (string username)
[lien librairie]

posix_getpwnam retourne un tableau associatif qui contient des informations à propos d'un utilisateur, identifié par son nom, passé en paramètre username .

Les éléments du tableau sont :

Elément	Déscription
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 `posix_getgrgid` pour connaître le nom du groupe, et ses membres.
gecos	GECOS est un terme obsolète qui fait référence aux données de finger, sur un système Honeywell. Le champs, cependant, a survécu, et son contenu a été formalisé par POSIX. Le champs 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 sytèmes, 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'éxécution du shell de l'utilisateur.

6.80.15 posix_getpwuid() : Retourne des informations sur un utilisateur.

`array` **posix_getpwuid** (`int uid`)
[lien librairie]

`posix_getpwuid` retourne un tableau associatif contenant des informations sur un utilisateur repéré par son UID, passé dans le paramètre `uid` .

Les éléments du tableau sont :

Elément	Description
name	Le nom contient le nom de l'utilisateur. Généralement, c'est un nom court, de moins de 16 caractères, mais ce n'est pas son nom réel et complet.
passwd	Contient le mot de passe de l'utilisateur, encrypté. Souvent, dans les systèmes utilisant les mots de 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 `posix_getgrgid` pour connaître le nom du groupe, et ses membres.
gecos	GECOS est un terme obsolète qui fait référence aux données de finger, sur un système Honeywell. Le champs, cependant, a survécu, et son contenu a été formalisé par POSIX. Le champs contient une liste, séparée par des virgules, qui contient le nom complet de l'utilisateur, son téléphone professionne, son numéro de bureau, et son numéro de téléphone personnel. Sur la plupart des sytèmes, 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'éxécution du shell de l'utilisateur.

6.80.16 posix_getrlimit() : Retourne les limites système.

`array` **posix_getrlimit** (`void`)
[lien librairie]

Encore à faire rapidement.

6.80.17 posix_getsid() : Retourne le sid du processus.

`int` **posix_getsid** (`int pid`)
[lien librairie]

`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 .

6.80.18 posix_getuid() : Retourne l'ID de l'utilisateur du processus courant.

`int` **posix_getuid** (`void`)
[lien librairie]

`posix_getuid` retourne l'ID numérique de l'utilisateur du processus courant. Reportez−vous à `posix_getpwuid` pour accéder au nom d'utilisateur.

6.80.19 posix_isatty() : Détermine si un pointeur de fichier est un terminal interactif.

`boolean` **posix_isatty** (`int fd`)
[lien librairie]

Encore à faire.

6.80.20 posix_kill() : Envoie un signal à un processus.

`boolean` **posix_kill** (`int pid` , `int sig`)
[lien librairie]

`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.

6.80.21 posix_mkfifo() : Crée un fichier fifo (first in, first out) (un pipe nommé).

`boolean` **posix_mkfifo** (`string pathname` , `int mode`)
[lien librairie]

Encore à faire très rapidement.

6.80.22 posix_setegid() : Modifie le GID réel du processus courant

`bool` **posix_setegid** (`int gid`)
[lien librairie]

`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.

Voir aussi `posix_setuid` .

6.80.23 posix_seteuid() : Set the effective UID of the current process

`bool` **posix_seteuid** (`int uid`)
[lien librairie]

`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.

Voir aussi `posix_setgid` .

6.80.24 posix_setgid() : Fixe le GID effective du processus courant.

`boolean` **posix_setgid** (`int gid`)
[lien librairie]

`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.

6.80.25 posix_setpgid() : Fixe l'identifiant de group de processus.

`int` **posix_setpgid** (`int pid` , `int pgid`)
[lien librairie]

posix_setpgid ajoute le processus pid au groupe d'id pgid . Reportez−vous à POSIX.1 et setsid(2) dans le manuel de votre système POSIX pour plus d'informations sur le contrôle de tâche. posix_setpgid retourne TRUE en cas de succès, et FALSE sinon.

6.80.26 posix_setsid() : Fait du processus courant un chef de session.

int **posix_setsid** (void)
[lien librairie]

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.

6.80.27 posix_setuid() : Fixe l'UID effective du processus courant.

boolean **posix_setuid** (int uid)
[lien librairie]

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 .

6.80.28 posix_times() : Utilisation des ressources.

array **posix_times** (void)
[lien librairie]

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.

6.80.29 posix_ttyname() : Retourne le nom de device du terminal.

string **posix_ttyname** (int fd)
[lien librairie]

Encore à faire.

6.80.30 posix_uname() : Retourne le nom du système.

array **posix_uname** (void)
[lien librairie]

`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.

6.81 PostgreSQL

6.81.1 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.

6.81.2 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 moduel 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/ .

6.81.3 Installation

Afin d'activer le support PostgreSQL, l'option `−−with−pgsql[=DIR]` est nécessaire lore de la compilation de PHP. Si le module shared object est disponible, le module PostgreSQL est chargeable avec la directive du fichier `php.ini` ou via la fonction `dl` .

6.81.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
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

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

`pgsql.allow_persistent` *`boolean`*
: 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.

6.81.5 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
`pg_exec`	`pg_query`
`pg_getlastoid`	`pg_last_oid`
`pg_cmdtuples`	`pg_affected_rows`
`pg_numrows`	`pg_num_rows`
`pg_numfields`	`pg_num_fields`
`pg_fieldname`	`pg_field_name`
`pg_fieldsize`	`pg_field_size`
`pg_fieldnum`	`pg_field_num`
`pg_fieldprtlen`	`pg_field_prtlen`
`pg_fieldisnull`	`pg_field_is_null`
`pg_freeresult`	`pg_free_result`
`pg_result`	`pg_fetch_result`
`pg_loreadall`	`pg_lo_read_all`
`pg_locreate`	`pg_lo_create`
`pg_lounlink`	`pg_lo_unlink`
`pg_loopen`	`pg_lo_open`
`pg_loclose`	`pg_lo_close`
`pg_loread`	`pg_lo_read`
`pg_lowrite`	`pg_lo_write`
`pg_loimport`	`pg_lo_import`
`pg_loexport`	`pg_lo_export`

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 libq est compilé. Si il y a des fonctions qui manquent, libpq ne supporte par 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 libq 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
postmaster &	pg_connect("dbname=MonDbName");	OK
postmaster −i &	pg_connect("dbname=MonDbName");	OK
postmaster &	pg_connect("host=localhost dbname=MonDbName");	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 affecent 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

```
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
```

6.81.6 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)

6.81.7 Exemples

Depuis PostgreSQL 7.1.0, vous opuvez stocker jusqu'à 1Go dans un champs 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)

```
<?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");
?>
```

You should not close the connection to the PostgreSQL server before closing the large object.

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Trucs et astuces d'utilisation

- Constantes Prédefinies
- Exemples
- pg_affected_rows
- pg_client_encoding
- pg_Close
- pg_connect
- pg_DBname
- pg_end_copy
- pg_Fetch_Array
- pg_Fetch_Object
- 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_Host
- pg_lo_close
- pg_lo_create
- pg_lo_export
- pg_lo_import
- pg_lo_open
- pg_lo_read_all
- pg_lo_read
- pg_lo_unlink
- pg_lo_write
- pg_Num_Fields
- pg_num_rows
- pg_Options
- pg_pConnect
- pg_Port
- pg_put_line
- pg_set_client_encoding
- pg_trace
- pg_tty
- pg_untrace

6.81.9 pg_client_encoding() : Lit l'encodage du client

`string` **pg_client_encoding** (*`resource connection`*)
[lien librairie]

`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 .

6.81.10 pg_Close() : Termine une connexion PostgreSQL.

boolean **pg_close** (resource connection)
[lien librairie]

pg_close retourne FALSE si l'index de connexion n'est pas valable, et TRUE sinon. pg_close ferme la connexion au serveur PostgreSQL associé à connection .

Note
Il n'est généralement pas nécessaire de fermer une connexion non persistante, car elles sont automatiquement fermées à la fin d'un script.

pg_close ne ferme pas les connexions persistantes ouvertes avec pg_pconnect .

6.81.11 pg_connect() : Ouvre une connexion.

resource **pg_connect** (string host , string port , string dbname)
[lien librairie]
resource **pg_connect** (string host , string port , string options , string dbname) resource **pg_connect** (string host , string port , string options , string tty , string dbname) resource **pg_connect** (string conn_string)

conn_string retourne un index de connexion en cas de succès, et FALSE sinon. Ouvre une connexion à un serveur PostgreSQL. Les arguments doivent être placé 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
?>
```

Les arguments disponibles comptent notamment dbname port , host , tty , options , 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 .

6.81.12 pg_DBname() : Nom de la base de données.

`string` **pg_dbname** (`resource connection`)
[lien librairie]

`pg_dbname` retourne le nom de la base de données PostgreSQL associée à l'index de connexion connection, ou FALSE si connection n'est pas valide.

6.81.13 pg_end_copy() : Synchronise avec le serveur PostgreSQL

`boolean` **pg_end_copy** (*`resource connection`*)
[lien librairie]

`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). `pg_end_copy` retourne TRUE en cas de succès, et FALSE sinon.

Pour plus de détails et un exemple voyez : `pg_put_line` .

6.81.14 pg_Fetch_Array() : Lit une ligne dans un tableau.

`array` **pg_fetch_array** (`resource result` , `int row` , *`int result_type`*)
[lien librairie]

`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.

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` .

PostgreSQL fetch array

```
<?php
$conn = pg_pconnect("dbname=publisher");
if (!$conn) {
    echo "Erreur de connexion.\n";
    exit;
}
$result = pg_exec($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Erreur durant la requete.\n";
```

```
    exit;
}
$arr = pg_fetch_array($result, 0);
echo $arr[0] . " <- array\n";
$arr = pg_fetch_array($result, 1);
echo $arr["author"] . " <- array\n";
?>
```

6.81.15 pg_Fetch_Object() : Lit une ligne dans un objet.

`object` **pg_fetch_object** (`resource result` , `int row` , *`int result_type`*)
[lien librairie]

`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).

Voir aussi `pg_fetch_array` et `pg_fetch_row` .

Lecture d'un objet Postgres

```
<?php
$database = "verlag";
$db_conn = pg_connect("host=localhost port=5432 dbname=$database");
if (!$db_conn):
?>
    <H1>Connexion impossible à la base postgres <?php echo $database ></H1> <?php
    exit;
endif;
$qu = pg_exec($db_conn, "SELECT * FROM verlag ORDER BY autor");
$row = 0; // postgres réclame un compteur de ligne, d'autres bases ne le font pas.
while ($data = pg_fetch_object($qu, $row)):
    echo $data->autor." (";
    echo $data->jahr ."): ";
    echo $data->titel."<BR>";
    $row++;
endwhile;
?>
<PRE><?php
$fields[] = array("autor", "Author");
$fields[] = array("jahr",  "  Year");
$fields[] = array("titel", " Title");
$row= 0; // Postgres réclame un compteur de ligne, d'autres bases ne le font pas.
while ($data = pg_fetch_object($qu, $row)):
```

```
    echo "----------\n";
    reset($fields);
    while (list(,$item) = each($fields)):
        echo $item[1].": ".$data->$item[0]."\n";
    endwhile;
    $row++;
endwhile;
echo "----------\n";
?>
</PRE>
<?php
pg_freeresult($qu);
pg_close($db_conn);
?>
```

6.81.16 pg_Fetch_Row() : Lit une ligne dans un tableau.

`array` **pg_fetch_row** (`resource result` , `int row`)
[lien librairie]

`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.

Voir aussi `pg_fetch_array` , `pg_fetch_object` et `pg_result` .

Postgres retourne une ligne

```
<?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] ";
  }
  echo "<br>";
}
?>
```

6.81.17 pg_field_is_null() : Teste si un champs est à NULL .

`int` **pg_field_is_null** (`resource result_id` , `int row` , `mixed field`)
[lien librairie]

`pg_field_is_null` teste si un champs est à `NULL` . `pg_field_is_null` retourne 0 si le champs n'est pas `NULL` . `pg_field_is_null` retourne 1 si le champs est à `NULL` . Le champs peut être identifié avec son nom ou son index numérique (commençant à 0).

Note
Auparavant, cette fonction s'appelait `pg_fieldisnull()` .

6.81.18 pg_field_Name() : Retourne le nom d'un champs.

`string` **pg_field_name** (`resource result_id` , `int field_number`)
[lien librairie]

`pg_field_name` va retourne le nom du champs qui occupe la colonne numéro `field_number` dans le résultat `result_id` . La numérotation des champs commence à 0.

Note
Auparavant, cette fonction s'appelait `pg_fieldname()` .

6.81.19 pg_field_Num() : Retourne le numéro d'une colonne.

`int` **pg_field_num** (`resource result_id` , `string field_name`)
[lien librairie]

`pg_field_num` retourne le numéro de la colonne, dont le nom est `field_name` , dans le résultat `result_id` . La numérotation des champs commence à 0. `pg_field_num` retournera −1 en cas d'erreur.

Note
Auparavant, cette fonction s'appelait `pg_fieldnum()` .

6.81.20 pg_field_PrtLen() : Retourne la taille imprimée.

`int` **pg_field_prtlen** (`resource result_id` , `int row_number` , `string field_name`)
[lien librairie]

`pg_field_prtlen` retourne la taille imprimée (nombre de caractères) d'une valeur donnée dans un résultat PostgreSQL. La numérotation des lignes commence à 0. `pg_field_prtlen` retourne −1 en cas d'erreur.

Note
Auparavant, cette fonction s'appelait `pg_fieldprtlen()` .

6.81.21 pg_field_Size() : Retourne la taille interne de stockage d'un champs donné.

`int` **pg_field_size** (`resource result_id` , `int field_number`)
[lien librairie]

`pg_field_size` retourne la taille interne de stockage d'un champs 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 `pg_fieldsize()` .

6.81.22 pg_field_Type() : Retourne le type d'un champs donné par index.

`string` **pg_field_type** (`resource result_id` , `int field_number`)
[lien librairie]

`pg_field_type` retourne une chaîne contenant le type du champs donné par son index `field_number` . La numérotation des champs commence à 0.

Note
Auparavant, cette fonction s'appelait `pg_fieldtype()` .

6.81.23 pg_Free_Result() : Libère la mémoire

`int` **pg_free_result** (`resource result_id`)
[lien librairie]

`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.

Note
Auparavant, cette fonction s'appelait `pg_freeresult()` .

6.81.24 pg_Host() : Retourne le nom d'hôte.

`string` **pg_host** (`resource connection_id`)
[lien librairie]

`pg_host` retourne le nom d'hôte associé à l'index de connexion PostgreSQL.

6.81.25 pg_lo_close() : Ferme un objet de grande taille.

`void` **pg_lo_close** (`resource fd`)
[lien librairie]

pg_lo_close ferme un objet de type Inversion Large Object. fd est un descripteur de fichier, obtenu avec pg_lo_open .

Note
Auparavant, cette fonction s'appelait pg_loclose() .

6.81.26 pg_lo_create() : Crée un objet de grande taille.

resource **pg_lo_create** (resource conn)
[lien librairie]

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).

Note
Auparavant, cette fonction s'appelait pg_locreate() .

6.81.27 pg_lo_export() : Exporte un objet de grande taille vers un fichier

boolean **pg_lo_export** (resource oid , int file , *resource connection_id*)
[lien librairie]

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. RetourneFALSEsi 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.

Note
Auparavant, cette fonction s'appelait pg_loexport() .

6.81.28 pg_lo_import() : Importe un objet de grande taille depuis un fichier

resource **pg_lo_import** (int file , *resource connection_id*)
[lien librairie]

filename 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. N'oubliez pas que la manipulation d'un objet de grande taille dans PostgreSQL doit intervenir dans une transaction.

Note
Auparavant, cette fonction s'appelait pg_loimport() .

6.81.29 pg_lo_open() : Ouvre un objet de grande taille.

`int` **pg_lo_open** (`resource conn` , `resource objoid` , `string mode`)
[lien librairie]

`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".

Note
Auparavant, cette fonction s'appelait `pg_loopen()` .

6.81.30 pg_lo_read_all() : Lit un objet de grande taille en totalité.

`void` **pg_lo_read_all** (`resource fd`)
[lien librairie]

`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.

Note
Auparavant, cette fonction s'appelait `pg_loreadall()` .

6.81.31 pg_lo_read() : Lit un objet de grande taille.

`string` **pg_lo_read** (`resource loid` , `int len`)
[lien librairie]

`pg_lo_read` lit au plus `len` octets d'un objet de grande taille, et retourne les données sous la forme d'une chaîne. `loid` 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 `pg_loread()` .

6.81.32 pg_lo_unlink() : Efface un objet de grande taille

`void` **pg_lo_unlink** (`resource conn` , `resource lobjid`)
[lien librairie]

`pg_lo_unlink` efface l'objet de grande taille dont l'identifiant est `lobjid` .

Note
Auparavant, cette fonction s'appelait `pg_lounlink()` .

6.81.33 pg_lo_write() : Ecrit un objet de grande taille

`int` **pg_lo_write** (`resource fd , string buf`)
[lien librairie]

`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` .

Note
Auparavant, cette fonction s'appelait `pg_lowrite()` .

6.81.34 pg_Num_Fields() : Retourne le nombre de champs

`int` **pg_Num_Fields** (`resource result_id`)
[lien librairie]

`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` .

6.81.35 pg_num_rows() : Retourne le nombre de lignes

`int` **pg_num_rows** (`resource result_id`)
[lien librairie]

`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
Auparavant, cette fonction s'appelait `pg_numrows()` .

Voir aussi `pg_num_fields` et `pg_affected_rows` .

6.81.36 pg_Options() : Retourne les options.

`string` **pg_options** (`resource connection_id`)
[lien librairie]

`pg_options` retourne une chaîne contenant les options de la connexion PostgreSQL.

6.81.37 pg_pConnect() : Etablit une connexion persistante.

`int` **pg_pconnect** (`string conn_string`)
[lien librairie]

`pg_pconnect` retourne un index de connexion en cas de succès, ou FALSE en cas d'erreur. `pg_pconnect` ouvre une connexion permanente à une base PostgreSQL. Les arguments doivent être insérés dans une chaîne à guillemets. Ils incluent : `host` , `port` , `tty` , `options` , `dbname` , `user` et `password` .

`pg_pconnect` retourne un indentifiant de connexion qui sera utilisées par les autres fonctions PostgreSQL. Vous pouvez ouvrir plusieurs connexions en même temps.

L'ancienne syntaxe `$conn = pg_pconnect("host", "port", "options", "tty", "dbname")` est obsolète.

6.81.38 pg_Port() : Retourne le numéro de port.

`int` **pg_port** (`resource connection_id`)
[lien librairie]

`pg_port` retourne le numéro de port de la connexion identifiée `connection_id` .

6.81.39 pg_put_line() : Envoie une chaîne au serveur PostgreSQL

`boolean` **pg_put_line** (*`resource connection_id`* , `string data`)
[lien librairie]

`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. `pg_put_line` retourne TRUE en cas de succès, et FALSE .

Note
Notez que l'application doit explicitement ajouter les deux caractères "\." à la fin de la chaîne pour indiquer au serveur qu'elle a finit d'envoyer des données.

Voir aussi `pg_end_copy` .

Insertion à grande vitesse dans une table

```
<?php
    $conn = pg_pconnect("dbname=foo");
    pg_exec($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);
?>
```

6.81.40 pg_set_client_encoding() : Choisit l'encodage du client

`int` **pg_set_client_encoding** (*`resource connection`* , `string encoding`)
[lien librairie]

`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` .

6.81.41 pg_trace() : Active le suivi d'une connexion PostgreSQL

`boolean` **pg_trace** (`string filename` , *`string mode`* , *`resource connection`*)
[lien librairie]

`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.

`filename` 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 `filename` a pu être ouvert en écriture, et`FALSE`sinon.

Voir aussi `fopen` et `pg_untrace` .

6.81.42 pg_tty() : Retourne le nom de tty.

`string` **pg_tty** (`resource connection_id`)
[lien librairie]

`pg_tty` retourne le nom de tty de la connexion associée à `connection_id` .

6.81.43 pg_untrace() : Termine le suivi d'une connexion PostgreSQL

`boolean` **pg_untrace** (*`resource connection`*)
[lien librairie]

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 .

6.82 Contrôle des processus

6.82.1 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.

Cettte documentation explique l'utilisation générale des fonctions du gestionaire 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 compulser un ouvrage de référence comme "Advanced Programming in the UNIX Environment" par W. Richard Stevens (Addison−Wesley).

6.82.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.82.3 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.

6.82.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.82.5 Types de ressources

Cette extension ne définit aucune ressource.

6.82.6 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)

6.82.7 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 gestinnaires 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
     }

}

?>
```

6.82.8 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

6.82.10 pcntl_fork() : Forks the currently running process

`int` **pcntl_fork** (`void`)
[lien librairie]

`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 processu fils ne sera créé et PHP lèvera une erreur.

Voir aussi `pcntl_waitpid` et `pcntl_signal` .

6.82.11 pcntl_signal() : Installe un gestionnaire de signaux

`bool` **pcntl_signal** (`int signo` , `mixed handle`)
[lien librairie]

`pcntl_signal` installe un nouveau gestionnaire de signaux opur 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` .

`pcntl_signal` retourne `TRUE` en cas de succès et `FALSE` en cas d'échec.

Voir aussi `pcntl_fork` et `pcntl_waitpid` .

6.82.12 pcntl_waitpid() : Attend la fin de l'exécution d'un processus fils

`int` **pcntl_waitpid** (`int pid` , `int status` , `int options`)
[lien librairie]

`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 :

`< −1`	attend que tous les processus fils dont l'identifiant de groupe est égal à la valeur absolue de `pid` soient terminés.
`−1`	attend que tous les processus fils soient terminés. Ceci est le même comportement que celui de la fonction `wait` .

0	attend que tous les processus fils dont l'identifiant de groupe est égal à celui du processus courant soient terminés.
> 0	attend que le processus fils dont l'identifiant est égal à pid soit terminé.

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) :

WNOHANG	retourne immédiatement si aucun processus fils ne s'est terminé.
WUNTRACED	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 .

6.82.13 pcntl_wexitstatus() : Retourne le code d'un processus fils terminé

int **pcntl_wexitstatus** (int status)
[lien librairie]

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 .

6.82.14 pcntl_wifexited() : Retourne TRUE si le code de retour représente une fin normale

int **pcntl_wifexited** (int status)
[lien librairie]

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 .

6.82.15 pcntl_wifsignaled() : Returns TRUE if status code represents a termination due to a signal

`int` **pcntl_wifsignaled** (`int status`)
[lien librairie]

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` .

6.82.16 pcntl_wifstopped() : Returns TRUE if child process is currently stopped

`int` **pcntl_wifstopped** (`int status`)
[lien librairie]

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` .

6.82.17 pcntl_wstopsig() : Returns the signal which caused the child to stop

`int` **pcntl_wstopsig** (`int status`)
[lien librairie]

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` .

6.82.18 pcntl_wtermsig() : Returns the signal which caused the child to terminate

`int` **pcntl_wtermsig** (`int status`)
[lien librairie]

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` .

6.83 Exécution de programmes externes

Ces fonctions fournissent la possibilité de passer directement des commandes au système, mais aussi de protéger le système des commandes passées. Ces fonctions sont complétées par l'opérateur guillemets obliques .

- escapeshellarg
- escapeshellcmd
- exec
- passthru
- proc_close
- proc_open
- shell_exec
- system

6.83.2 escapeshellcmd() : Echappe les méta−caractères Shell.

`string` **escapeshellcmd** (`string command`)
[lien librairie]

`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 :

```
<?php
  system(escapeshellcmd($cmd));
?>
```

Voir aussi `exec` , `popen` , `system` , et les opérateurs guillemets obliques .

6.83.3 exec() : Exécute un programme externe.

`string` **exec** (`string command` , *`string array`* , *`int return_var`*)
[lien librairie]

`exec` éxecute 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 de 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.

Voir aussi system , passthru , popen , escapeshellcmd , et les opérateurs guillemets obliques .

6.83.4 passthru() : Exécute un programme externe et affiche le résultat brut.

void **passthru** (string command , *int return_var*)
[lien librairie]

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 retourne des images.

Voir aussi exec , system , popen , escapeshellcmd , et les opérateurs guillemets obliques .

6.83.5 proc_close() : Ferme un processus ouvert par proc_open() et retourne le code de sortie du processus.

int **proc_close** (resource process)
[lien librairie]

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'appler cette fonction pour éviter des verrouillages : le processus peut ne pas pouvoir sortir tant que les pipes sont ouverts.

6.83.6 proc_open() : Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties

resource **proc_open** (string cmd , array descriptorspec , array pipes)
[lien librairie]

proc_open est similaire à popen mais fournie 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 détuire avec la fonction proc_close lorsque vous aurez fini.

Exemple avec proc_open

```
$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 prcessus fils
    // 1 => fichier accessible en lecture, connecté à la sortie standard du prcessus fils
    // Toute erreur sera ajoutée au fichier /tmp/error-output.txt

    fwrite($pipes[0], "<?php echo
Warning: Unexpected character in input:  '\' (ASCII=92) state=1 in /Users/imac/Desktop/CVS/phpdoc-fr
"Bonjour le monde!\"; ?>");
    fclose($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 limitées 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-dela 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 fonctionne 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 `exec` , `system` , `passthru` , `popen` , `escapeshellcmd` , et l'opérateur de backticks .

6.83.7 shell_exec() : Exécute une commande via le shell et retourne le résultat sous forme de chaîne.

`string` **shell_exec** (`string cmd`)
[lien librairie]

`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 .

6.83.8 system() : Exécute un programme externe et affiche le résultat.

`string` **system** (`string command` , *`int return_var`*)
[lien librairie]

`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` , `escapeshellcmd` et les opérateurs guillemets obliques .

6.84 Printer functions

6.84.1 Introduction

These functions are only available under Windows 9.x, ME, NT4 and 2000. They have been added in PHP 4 (4.0.4).

6.84.2 Installation

Add the line `extension=php_printer.dll` to your `php.ini` file.

6.84.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Name	Default	Changeable
printer.default_printer	""	PHP_INI_ALL

For further details and definition of the PHP_INI_* constants see `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

6.84.5 printer_close() : Close an open printer connection

`void` **printer_close** (`resource handle`)
[lien librairie]

This function closes the printer connection. `printer_close` also closes the active device context.

`handle` must be a valid handle to a printer.

`printer_close` example
`$handle = printer_open();` `printer_close($handle);`

6.84.6 printer_create_brush() : Create a new brush

`mixed` **printer_create_brush** (`int style` , `string color`)
[lien librairie]

The function creates a new brush and returns a handle to it. A brush is used to fill shapes. For an example see `printer_select_brush` . `color` must be a color in RGB hex format, i.e. "000000" for black, `style` must be one of the following constants:

- `PRINTER_BRUSH_SOLID` : creates a brush with a solid color.
- `PRINTER_BRUSH_DIAGONAL` : creates a brush with a 45−degree upward left−to−right hatch (/).
- `PRINTER_BRUSH_CROSS` : creates a brush with a cross hatch (+).
- `PRINTER_BRUSH_DIAGCROSS` : creates a brush with a 45 cross hatch (x).
- `PRINTER_BRUSH_FDIAGONAL` : creates a brush with a 45−degree downward left−to−right hatch (\ ).
- `PRINTER_BRUSH_HORIZONTAL` : creates a brush with a horizontal hatch (−).
- `PRINTER_BRUSH_VERTICAL` : creates a brush with a vertical hatch (|).
- `PRINTER_BRUSH_CUSTOM` : creates a custom brush from an BMP file. The second parameter is used to specify the BMP instead of the RGB color code.

6.84.7 printer_create_dc() : Create a new device context

`void` **printer_create_dc** (`resource handle`)
[lien librairie]

The function creates a new device context. A device context is used to customize the graphic objects of the document. `handle` must be a valid handle to a printer.

printer_create_dc example

```
$handle = printer_open();
printer_start_doc($handle);
printer_start_page($handle);

printer_create_dc($handle);
/* do some stuff with the dc */
printer_set_option($handle, PRINTER_TEXT_COLOR, "333333");
printer_draw_text($handle, 1, 1, "text");
printer_delete_dc($handle);

/* create another dc */
printer_create_dc($handle);
printer_set_option($handle, PRINTER_TEXT_COLOR, "000000");
printer_draw_text($handle, 1, 1, "text");
/* do some stuff with the dc */

printer_delete_dc($handle);

printer_endpage($handle);
printer_end_doc($handle);
printer_close($handle);
```

6.84.8 printer_create_font() : Create a new font

`mixed` **printer_create_font** (`string face` , `int height` , `int width` , `int font_weight` , `bool italic` , `bool underline` , `bool strikeout` , `int orientaton`)
[lien librairie]

The function creates a new font and returns a handle to it. A font is used to draw text. For an example see `printer_select_font` . `face` must be a string specifying the font face. `height` specifies the font height, and `width` the font width. The `font_weight` specifies the font weight (400 is normal), and can be one of the following predefined constants.

- `PRINTER_FW_THIN` : sets the font weight to thin (100).
- `PRINTER_FW_ULTRALIGHT` : sets the font weight to ultra light (200).
- `PRINTER_FW_LIGHT` : sets the font weight to light (300).
- `PRINTER_FW_NORMAL` : sets the font weight to normal (400).
- `PRINTER_FW_MEDIUM` : sets the font weight to medium (500).
- `PRINTER_FW_BOLD` : sets the font weight to bold (700).
- `PRINTER_FW_ULTRABOLD` : sets the font weight to ultra bold (800).
- `PRINTER_FW_HEAVY` : sets the font weight to heavy (900).

`italic` can be `TRUE` or `FALSE` , and sets whether the font should be italic.

`underline` can be `TRUE` or `FALSE` , and sets whether the font should be underlined.

`strikeout` can be `TRUE` or `FALSE` , and sets whether the font should be striked out.

`orientation` specifies a rotation. For an example see `printer_select_font` .

6.84.9 printer_create_pen() : Create a new pen

`mixed` **printer_create_pen** (`int style` , `int width` , `string color`)
[lien librairie]

The function creates a new pen and returns a handle to it. A pen is used to draw lines and curves. For an example see `printer_select_pen` . `color` must be a color in RGB hex format, i.e. "000000" for black, `width` specifies the width of the pen whereas `style` must be one of the following constants:

- `PRINTER_PEN_SOLID` : creates a solid pen.
- `PRINTER_PEN_DASH` : creates a dashed pen.
- `PRINTER_PEN_DOT` : creates a dotted pen.
- `PRINTER_PEN_DASHDOT` : creates a pen with dashes and dots.
- `PRINTER_PEN_DASHDOTDOT` : creates a pen with dashes and double dots.
- `PRINTER_PEN_INVISIBLE` : creates an invisible pen.

6.84.10 printer_delete_brush() : Delete a brush

`bool` **printer_delete_brush** (`resource handle`)
[lien librairie]

The function deletes the selected brush. For an example see `printer_select_brush` . It returns `TRUE` on success, or `FALSE` otherwise. `handle` must be a valid handle to a brush.

6.84.11 printer_delete_dc() : Delete a device context

`bool` **printer_delete_dc** (`resource handle`)
[lien librairie]

The function deletes the device context and returns `TRUE` on success, or `FALSE` if an error occurred. For an example see `printer_create_dc` . `handle` must be a valid handle to a printer.

6.84.12 printer_delete_font() : Delete a font

`bool` **printer_delete_font** (`resource handle`)
[lien librairie]

The function deletes the selected font. For an example see `printer_select_font` . It returns `TRUE` on success, or `FALSE` otherwise. `handle` must be a valid handle to a font.

6.84.13 printer_delete_pen() : Delete a pen

`bool` **printer_delete_pen** (`resource handle`)
[lien librairie]

The function deletes the selected pen. For an example see `printer_select_pen` . It returns `TRUE` on success, or `FALSE` otherwise. `handle` must be a valid handle to a pen.

6.84.14 printer_draw_bmp() : Draw a bmp

`void` **printer_draw_bmp** (`resource handle` , `string filename` , `int x` , `int y`)
[lien librairie]

The function simply draws an bmp the bitmap `filename` at position `x` , `y` . `handle` must be a valid handle to a printer.

The function returns `TRUE` on success, or otherwise `FALSE` .

`printer_draw_bmp` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.15 printer_draw_chord() : Draw a chord

`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`)
[lien librairie]

The function simply draws an chord. `handle` must be a valid handle to a printer.

`rec_x` is the upper left x coordinate of the bounding rectangle.

`rec_y` is the upper left y coordinate of the bounding rectangle.

`rec_x1` is the lower right x coordinate of the bounding rectangle.

`rec_y1` is the lower right y coordinate of the bounding rectangle.

`rad_x` is x coordinate of the radial defining the beginning of the chord.

`rad_y` is y coordinate of the radial defining the beginning of the chord.

`rad_x1` is x coordinate of the radial defining the end of the chord.

`rad_y1` is y coordinate of the radial defining the end of the chord.

`printer_draw_chord` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.16 printer_draw_elipse() : Draw an ellipse

`void` **printer_draw_elipse** (`resource handle` , `int ul_x` , `int ul_y` , `int lr_x` , `int lr_y`)
[lien librairie]

The function simply draws an ellipse. `handle` must be a valid handle to a printer.

`ul_x` is the upper left x coordinate of the ellipse.

`ul_y` is the upper left y coordinate of the ellipse.

`lr_x` is the lower right x coordinate of the ellipse.

`lr_y` is the lower right y coordinate of the ellipse.

`printer_draw_elipse` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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_elipse($handle, 1, 1, 500, 500);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
```

6.84.17 printer_draw_line() : Draw a line

`void` **printer_draw_line** (`resource printer_handle` , `int from_x` , `int from_y` , `int to_x` , `int to_y`)
[lien librairie]

The function simply draws a line from position `from_x` , `from_y` to position `to_x` , `to_y` using the selected pen. `printer_handle` must be a valid handle to a printer.

`printer_draw_line` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.18 printer_draw_pie() : Draw a pie

`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`)
[lien librairie]

The function simply draws an pie. `handle` must be a valid handle to a printer.

`rec_x` is the upper left x coordinate of the bounding rectangle.

`rec_y` is the upper left y coordinate of the bounding rectangle.

`rec_x1` is the lower right x coordinate of the bounding rectangle.

`rec_y1` is the lower right y coordinate of the bounding rectangle.

`rad1_x` is x coordinate of the first radial's ending.

`rad1_y` is y coordinate of the first radial's ending.

`rad2_x` is x coordinate of the second radial's ending.

`rad2_y` is y coordinate of the second radial's ending.

`printer_draw_pie` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.19 printer_draw_rectangle() : Draw a rectangle

`void` **printer_draw_rectangle** (`resource handle` , `int ul_x` , `int ul_y` , `int lr_x` , `int lr_y`)
[lien librairie]

The function simply draws a rectangle.

`handle` must be a valid handle to a printer.

`ul_x` is the upper left x coordinate of the rectangle.

`ul_y` is the upper left y coordinate of the rectangle.

`lr_x` is the lower right x coordinate of the rectangle.

`lr_y` is the lower right y coordinate of the rectangle.

`printer_draw_rectangle` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.20 printer_draw_roundrect() : Draw a rectangle with rounded corners

`void` **printer_draw_roundrect** (`resource handle` , `int ul_x` , `int ul_y` , `int lr_x` , `int lr_y` , `int width` , `int height`)
[lien librairie]

The function simply draws a rectangle with rounded corners.

`handle` must be a valid handle to a printer.

`ul_x` is the upper left x coordinate of the rectangle.

`ul_y` is the upper left y coordinate of the rectangle.

`lr_x` is the lower right x coordinate of the rectangle.

`lr_y` is the lower right y coordinate of the rectangle.

`width` is the width of the ellipse.

`height` is the height of the ellipse.

`printer_draw_roundrect` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.21 printer_draw_text() : Draw text

`void` **printer_draw_text** (`resource printer_handle` , `string text` , `int x` , `int y`)
[lien librairie]

The function simply draws `text` at position `x` , `y` using the selected font. `printer_handle` must be a valid handle to a printer.

`printer_draw_text` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.22 printer_end_doc() : Close document

`bool` **printer_end_doc** (`resource handle`)
[lien librairie]

Closes a new document in the printer spooler. The document is now ready for printing. For an example see `printer_start_doc` . `handle` must be a valid handle to a printer.

6.84.23 printer_end_page() : Close active page

`bool` **printer_end_page** (`resource handle`)
[lien librairie]

The function closes the active page in the active document. For an example see `printer_start_doc` . `handle` must be a valid handle to a printer.

6.84.24 printer_get_option() : Retrieve printer configuration data

`mixed` **printer_get_option** (`resource handle` , `string option`)
[lien librairie]

The function retrieves the configuration setting of `option` . `handle` must be a valid handle to a printer. Take a look at `printer_set_option` for the settings that can be retrieved, additionally the following settings can be retrieved:

- `PRINTER_DEVICENAME` returns the devicename of the printer.
- `PRINTER_DRIVERVERSION` returns the printer driver version.

`printer_get_option` example

```
$handle = printer_open();
print printer_get_option($handle, PRINTER_DRIVERVERSION);
```

```
printer_close($handle);
```

6.84.25 printer_list() : Return an array of printers attached to the server

`array` **printer_list** (`int enumtype` , *`string name`* , *`int level`*)
[lien librairie]

The function enumerates available printers and their capabilities. `level` sets the level of information request. Can be 1,2,4 or 5. `enumtype` must be one of the following predefined constants:

- `PRINTER_ENUM_LOCAL` : enumerates the locally installed printers.
- `PRINTER_ENUM_NAME` : enumerates the printer of `name` , can be a server, domain or print provider.
- `PRINTER_ENUM_SHARED` : this parameter can't be used alone, it has to be OR'ed with other parameters, i.e. PRINTER_ENUM_LOCAL to detect the locally shared printers.
- `PRINTER_ENUM_DEFAULT` : (Win9.x only) enumerates the default printer.
- `PRINTER_ENUM_CONNECTIONS` : (WinNT/2000 only) enumerates the printers to which the user has made connections.
- `PRINTER_ENUM_NETWORK` : (WinNT/2000 only) enumerates network printers in the computer's domain. Only valid if `level` is 1.
- `PRINTER_ENUM_REMOTE` : (WinNT/2000 only) enumerates network printers and print servers in the computer's domain. Only valid if `level` is 1.

`printer_list` example

```
/* detect locally shared printer */
var_dump( printer_list(PRINTER_ENUM_LOCAL | PRINTER_ENUM_SHARED) );
```

6.84.26 printer_logical_fontheight() : Get logical font height

`int` **printer_logical_fontheight** (`resource handle` , `int height`)
[lien librairie]

The function calculates the logical font height of `height` . `handle` must be a valid handle to a printer.

`printer_logical_fontheight` example

```
$handle = printer_open();
print printer_logical_fontheight($handle, 72);
printer_close($handle);
```

6.84.27 printer_open() : Open connection to a printer

`mixed` **printer_open** (*`string devicename`*)
[lien librairie]

This function tries to open a connection to the printer `devicename` , and returns a handle on success or FALSE on failure.

If no parameter was given it tries to open a connection to the default printer (if not specified in `php.ini` as `printer.default_printer` , php tries to detect it).

`printer_open` also starts a device context.

`printer_open` example

```
$handle = printer_open("HP Deskjet 930c");
$handle = printer_open();
```

6.84.28 printer_select_brush() : Select a brush

`void` **printer_select_brush** (`resource printer_handle` , `resource brush_handle`)
[lien librairie]

The function selects a brush as the active drawing object of the actual device context. A brush is used to fill shapes. If you draw an rectangle the brush is used to draw the shapes, while the pen is used to draw the border. If you haven't selected a brush before drawing shapes, the shape won't be filled. `printer_handle` must be a valid handle to a printer. `brush_handle` must be a valid handle to a brush.

`printer_select_brush` example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.29 printer_select_font() : Select a font

void **printer_select_font** (resource printer_handle , resource font_handle)
[lien librairie]

The function selects a font to draw text. printer_handle must be a valid handle to a printer. font_handle must be a valid handle to a font.

printer_select_font example

```
$handle = printer_open();
printer_start_doc($handle, "My 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 is simply cool", 40, 40);
printer_delete_font($font);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
```

6.84.30 printer_select_pen() : Select a pen

void **printer_select_pen** (resource printer_handle , resource pen_handle)
[lien librairie]

The function selects a pen as the active drawing object of the actual device context. A pen is used to draw lines and curves. I.e. if you draw a single line the pen is used. If you draw an rectangle the pen is used to draw the borders, while the brush is used to fill the shape. If you haven't selected a pen before drawing shapes, the shape won't be outlined. printer_handle must be a valid handle to a printer. pen_handle must be a valid handle to a pen.

printer_select_pen example

```
$handle = printer_open();
printer_start_doc($handle, "My 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);
```

6.84.31 printer_set_option() : Configure the printer connection

`bool` **printer_set_option** (`resource handle` , `int option` , `mixed value`)
[lien librairie]

The function sets the following options for the current connection. `handle` must be a valid handle to a printer. For `option` can be one of the following constants:

- `PRINTER_COPIES` : sets how many copies should be printed, `value` must be an integer.
- `PRINTER_MODE` : specifies the type of data (text, raw or emf), `value` must be a string.
- `PRINTER_TITLE` : specifies the name of the document, `value` must be a string.
- `PRINTER_ORIENTATION` : specifies the orientation of the paper, `value` can be either PRINTER_ORIENTATION_PORTRAIT or PRINTER_ORIENTATION_LANDSCAPE
- `PRINTER_RESOLUTION_Y` : specifies the y−resolution in DPI, `value` must be an integer.
- `PRINTER_RESOLUTION_X` : specifies the x−resolution in DPI, `value` must be an integer.
- `PRINTER_PAPER_FORMAT` : specifies the a predefined paper format, set `value` to PRINTER_FORMAT_CUSTOM if you want to specify a custom format with PRINTER_PAPER_WIDTH and PRINTER_PAPER_LENGTH. `value` can be one of the following constants.
    - ♦ `PRINTER_FORMAT_CUSTOM` : let's you specify a custom paper format.
    - ♦ `PRINTER_FORMAT_LETTER` : specifies standard letter format (8 1/2− by 11−inches).
    - ♦ `PRINTER_FORMAT_LETTER` : specifies standard legal format (8 1/2− by 14−inches).
    - ♦ `PRINTER_FORMAT_A3` : specifies standard A3 format (297− by 420−millimeters).
    - ♦ `PRINTER_FORMAT_A4` : specifies standard A4 format (210− by 297−millimeters).
    - ♦ `PRINTER_FORMAT_A5` : specifies standard A5 format (148− by 210−millimeters).
    - ♦ `PRINTER_FORMAT_B4` : specifies standard B4 format (250− by 354−millimeters).
    - ♦ `PRINTER_FORMAT_B5` : specifies standard B5 format (182− by 257−millimeter).
    - ♦ `PRINTER_FORMAT_FOLIO` : specifies standard FOLIO format (8 1/2− by 13−inch).
- `PRINTER_PAPER_LENGTH` : if PRINTER_PAPER_FORMAT is set to PRINTER_FORMAT_CUSTOM, PRINTER_PAPER_LENGTH specifies a custom paper length in mm, `value` must be an integer.
- `PRINTER_PAPER_WIDTH` : if PRINTER_PAPER_FORMAT is set to PRINTER_FORMAT_CUSTOM, PRINTER_PAPER_WIDTH specifies a custom paper width in mm, `value` must be an integer.
- `PRINTER_SCALE` : specifies the factor by which the printed output is to be scaled. the page size is scaled from the physical page size by a factor of scale/100. for example if you set the scale to 50, the output would be half of it's original size. `value` must be an integer.
- `PRINTER_BACKGROUND_COLOR` : specifies the background color for the actual device context, `value` must be a string containing the rgb information in hex format i.e. "005533".
- `PRINTER_TEXT_COLOR` : specifies the text color for the actual device context, `value` must be a string containing the rgb information in hex format i.e. "005533".
- `PRINTER_TEXT_ALIGN` : specifies the text alignment for the actual device context, `value` can be combined through OR'ing the following constants:
    - ♦ `PRINTER_TA_BASELINE` : text will be aligned at the base line.
    - ♦ `PRINTER_TA_BOTTOM` : text will be aligned at the bottom.
    - ♦ `PRINTER_TA_TOP` : text will be aligned at the top.
    - ♦ `PRINTER_TA_CENTER` : text will be aligned at the center.
    - ♦ `PRINTER_TA_LEFT` : text will be aligned at the left.
    - ♦ `PRINTER_TA_RIGHT` : text will be aligned at the right.

`printer_set_option` example

```
$handle = printer_open();
printer_set_option($handle, PRINTER_SCALE, 75);
```

```
printer_set_option($handle, PRINTER_TEXT_ALIGN, PRINTER_TA_LEFT);
printer_close($handle);
```

6.84.32 printer_start_doc() : Start a new document

`bool` **printer_start_doc** (`resource handle` , *`string document`*)
[lien librairie]

The function creates a new document in the printer spooler. A document can contain multiple pages, it's used to schedule the print job in the spooler. `handle` must be a valid handle to a printer. The optional parameter `document` can be used to set an alternative document name.

`printer_start_doc` example

```
$handle = printer_open();
printer_start_doc($handle, "My Document");
printer_start_page($handle);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
```

6.84.33 printer_start_page() : Start a new page

`bool` **printer_start_page** (`resource handle`)
[lien librairie]

The function creates a new page in the active document. For an example see `printer_start_doc` . `handle` must be a valid handle to a printer.

6.84.34 printer_write() : Write data to the printer

`bool` **printer_write** (`resource handle` , `string content`)
[lien librairie]

Writes `content` directly to the printer, and returns TRUE on success or FALSE if it failed.

`handle` must be a valid handle to a printer.

`printer_write` example

```
$handle = printer_open();
printer_write($handle, "Text to print");
printer_close($handle);
```

6.85 Pspell

6.85.1 Introduction

La librairie pspell vous permet de vérifier l'orthographe d'un mot, et suggérer des corrections.

6.85.2 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'otpion `−−with−pspell[=dir]` lors de la compilation de PHP.

6.85.3 Installation

If you have the libraries needed add the `−−with−pspell[=dir]` option when compiling PHP.

6.85.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.85.5 Types de ressources

Cette extension ne définit aucune ressource.

6.85.6 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

6.85.8 pspell_add_to_session() : Ajoute le mot au dictionnaire personnel de la session courante

`int` **pspell_add_to_session** (`resource dictionary_link` , `string word`)
[lien librairie]

`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` .

6.85.9 pspell_check() : Vérifie un mot

`boolean` **pspell_check** (`resource dictionary_link` , `string word`)
[lien librairie]

`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";
}
?>
```

6.85.10 pspell_clear_session() : Remet à zéro la session courante.

`int` **pspell_clear_session** (`resource dictionary_link`)
[lien librairie]

`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
?>
```

6.85.11 pspell_config_create() : Crée une configuration utilisée pour ouvrir un dictionnaire

`resource` **pspell_config_create** (`string language` , *`string spelling`* , *`string jargon`* , *`string encoding`*)
[lien librairie]

`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 créer 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);
?>
```

6.85.12 pspell_config_ignore() : Ignore les mots des moins de N caractères.

`int` **pspell_config_ignore** (`resource dictionary_link` , `int n`)
[lien librairie]

`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
?>
```

6.85.13 pspell_config_mode() : Change le mode de suggestion

`int` **pspell_config_mode** (`resource dictionary_link` , `int mode`)
[lien librairie]

`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)

`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");
?>
```

6.85.14 pspell_config_personal() : Choisit le fichier qui contient le dictionnaire personnel

`int` **pspell_config_personal** (`resource dictionary_link` , `string file`)
[lien librairie]

`pspell_config_personal` doit être appelé dans une configuration avant d'appeler `pspell_new_config` . Le dictionnaire personnel sera chargé est 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");
?>
```

6.85.15 pspell_config_repl() : Choisit le fichier qui contient les paires de remplacement.

int **pspell_config_repl** (resource dictionary_link , string file)
[lien librairie]

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");
?>
```

6.85.16 pspell_config_runtogether() : Considère deux mots accolés comme un composé.

int **pspell_config_runtogether** (resource dictionary_link , boolean flag)
[lien librairie]

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");
?>
```

6.85.17 pspell_config_save_repl() : Active la sauvegarde des paires de remplacement

`int` **pspell_config_save_repl** (`resource dictionary_link` , `boolean flag`)
[lien librairie]

`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.

6.85.18 pspell_new_config() : Charge un nouveau dictionnaire

`resource` **pspell_new_config** (`resource config`)
[lien librairie]

`pspell_new_config` ouvre un nouveau dictionnaire et charge les paramètrages spécifié 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.

`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);
?>
```

6.85.19 pspell_new_personal() : Charge un nouveau dictionnaire avec un dictionnaire personnel

`resource` **pspell_new_personal** (`string personal` , `string language` , *`string spelling`* , *`string jargon`* , *`string encoding`* , *`int mode`*)
[lien librairie]

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 pluspart 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
$pspell_link = pspell_new_personal ("/var/dictionaries/custom.pws", "en", "", "", "", PSPELL_FAST|PS
?>
```

6.85.20 pspell_new() : Charge un nouveau dictionnaire

resource **pspell_new** (string language , *string spelling* , *string jargon* , *string encoding*)
[lien librairie]

pspell_new ouvre un nouveau dictionnaire et retourne un identifiant de dictionnaire, pour utiliser avec d'autres fonctions pspell.

Le paramèegrave;tre 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 paramèegrave;tre est nécessaire pour les langues qui ont plus d'une orthographe, comme l'anglais ou le français. Les valeurs reconnues sont ``americain'', ``britannique'', et ``canadien''.

Le paramèegrave;tre de jargon contient des informations supplémentaires pour distinguer deux listes de mots qui ont le même marquage de langue et d'orthographe.

Le paramèegrave;tre d'encodage 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 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)
- 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 résutlat retournés par `pspell_check` ; `pspell_suggest` retournera toujours les mêmes suggestions.

`Mode` est un champs 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).

`pspell_new`

```
<?php
$pspell_link = pspell_new("en", "", "", "", (PSPELL_FAST|PSPELL_RUN_TOGETHER));
?>
```

6.85.21 pspell_save_wordlist() : Sauve le dictionnaire personnel dans un fichier.

`int` **pspell_save_wordlist** `( resource dictionary_link )`
[lien librairie]

`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);
?>
```

6.85.22 pspell_store_replacement() : Enregistre une paire de remplacement pour un mot

`int` **pspell_store_replacement** (`resource dictionary_link` , `string misspelled` , `string correct`)
[lien librairie]

`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);
?>
```

6.85.23 pspell_suggest() : Suggèegrave;re une orthographe

`array` **pspell_suggest** (`resource dictionary_link` , `string word`)
[lien librairie]

`pspell_suggest` retourne un tableau de suggestions pour le mot `word` .

`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 suggerées : " . $suggestions[$i] . "<br>";
    }
}
?>
```

6.86 Readline (GNU)

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.

Le site du projet GNU Readline est http://cnswww.cns.cwru.edu/~chet/readline/rltop.html . Elle est entretenue par Chet Ramey, qui est aussi l'auteur de Bash.

- readline_add_history
- readline_clear_history
- readline_completion_function
- readline_info
- readline_list_history
- readline_read_history
- readline_write_history
- readline

6.86.2 readline_clear_history() : Efface l'historique

`boolean` **readline_clear_history** (`void`)
[lien librairie]

`readline_clear_history` efface tout l'historique.

6.86.3 readline_completion_function() : Enregistre une fonction de complétion

`boolean` **readline_completion_function** (`string line`)
[lien librairie]

`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.

6.86.4 readline_info() : Lit/modifie diverses variables internes

`mixed` **readline_info** (*`string varname`* , *`string newvalue`*)
[lien librairie]

Appelée sans paramètre, `readline_info` retourne une 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` .

6.86.5 readline_list_history() : Liste l'historique

`array` **readline_list_history** (`void`)
[lien librairie]

`readline_list_history` retourne un tableau avec la liste de toutes les lignes de commandes de l'historique. Les éléments sont indéxés numériquement, à partir de 0.

6.86.6 readline_read_history() : Lit l'historique

`boolean` **readline_read_history** (`string filename`)
[lien librairie]

`readline_read_history` lit une ligne de l'historique.

6.86.7 readline_write_history() : Ecrit dans l'historique

`boolean` **readline_write_history** (`string filename`)
[lien librairie]

`readline_write_history` écrit `filename` dans l'historique.

6.86.8 readline() : Lit une ligne

`string` **readline** (*`string prompt`*)
[lien librairie]

`readline` retourne une ligne entrée par l'utilisateur. Vous pouvez spécifier une chaîne de prompt. La ligne retournée est débarassée 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());
?>
```

6.87 Recode (GNU)

Ce module contient l'interface à la librairie GNU Recode library, version 3.5. Pour pouvoir utiliser ces fonctions, il faut que PHP ait été compilé avec l'option −−with−recode. Pour cela, il faut que vous ayez la librairie GNU Recode 3.5 ou plus récent, installée sur votre système.

La librairie GNU Recode library convertit les fichiers ayant des jeux de caractères différents. Lorsque ce n'est pas possible, elle se débarasse des caractères illégaux, ou bien effectue une approximation. La librairie reconnait 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.

- recode_file
- recode_string
- recode

6.87.2 recode_string() : Recode une chaîne en fonction de la requête.

`string` **recode_string** (`string request` , `string string`)
[lien librairie]

`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 : á");`
`?>`

6.87.3 recode() : Recode une fonction grâce à une requête

`string` **recode** (`string request` , `string string`)
[lien librairie]

Note
`recode` est un alias de `recode_string` . Elle a été ajoutée en PHP 4.

6.88 Expressions régulières compatibles Perl

6.88.1 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 .

6.88.2 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/ .

6.88.3 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` .

6.88.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.88.5 Types de ressources

Cette extension ne définit aucune ressource.

6.88.6 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 $matches[0] contienne les résultats qui correspondent au masque entier, $matches[1] contienne les résultats qui correspondent à la première parenthèse capturante, $matches[2] 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 $matches[0] contient la première série de résultat, $matches[1] 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 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. Cette constante est utilisée avec preg_split .
PREG_SPLIT_OFFSET_CAPTURE	Si cette constante est utilisée avec preg_split , 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.

6.88.7 Exemples

Exemples de masques valides
• /<\/\w+>/ • |(\d{3})−\d+|Sm • /^(?i)php[34]/ • {^\s+(\s+)?$}

Exemples de masques invalides
• /href='(.*)' − délimiteur final manquant • /\w+\s*\w+/J − option 'J' inconnue • 1−\d3−\d3−\d4| − délimiteur initial manquant

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- Exemples
- options de recherche
- syntaxe des masques
- preg_grep
- preg_match_all
- preg_match
- preg_quote
- preg_replace_callback
- preg_replace
- preg_split

6.88.9 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.

[lien librairie]

6.88.10 preg_grep() : Retourne un tableau avec les résultat de la recherche.

`array` **preg_grep** (`string pattern` , `array input`)
[lien librairie]

`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("/^(&#92;d+)?&#92;.&#92;d+$/", $array);
?>
```

6.88.11 preg_match_all() : Expression régulière globale.

`int` **preg_match_all** (`string pattern` , `string subject` , `array matches` , *`int order`*)
[lien librairie]

`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 :

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..

```
<?php
  preg_match_all("|<[^>]+>(.*)</[^>]+>|U", "<b>example: </b><div align=left>a test</div
  print $out[0][0].", ".$out[0][1]."&#92;n";
```

```
  print $out[1][0].", ".$out[1][1]."&#92;n";
?>
```

Cet exemple va afficher :

```
<b>exemple: </b>, <div align=left>ceci est un test</div>

exemple: , ceci est 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...

```
<?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]."&#92;n";
  print $out[1][0].", ".$out[1][1]."&#92;n";
?>
```

Cet exemple va afficher :

```
<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...

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("/&#92;(?  (&#92;d{3})?  &#92;)?  (?(1)  [&#92;-&#92;s] ) &#92;d{3}-&#92;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 (&#92;&#92;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
// capturante qui doit être utilisé (ici, ([&#92;w]+)).
//  L'antislash est nécessaire ici, car la chaîne est entre guillemets doubles
  $html = "<B>Texte en gras</B><a href=salut.html>clique moi</?>
  preg_match_all ("/(<([&#92;w]+)[?>]?>)(.*)(<&#92;/&#92;&#92;?>)/", $html, $matches);
  for ($i=0; $i< count($matches[0]); $i++) {
    echo "trouvé: ".$matches[0][$i]."&#92;n";
    echo "partie 1: ".$matches[1][$i]."&#92;n";
    echo "partie 2: ".$matches[3][$i]."&#92;n";
    echo "partie 3: ".$matches[4][$i]."&#92;n&#92;n";
  }
?>
```

Cet exemple va produire :

```
trouvé: <B>bold text</B>
partie 1: <B>
partie 2: Test en gras
partie 3: </B>
trouvé: <a href=salut.html>clique moi</a>
partie 1: <a href=salut.html>
partie 2: clique moi
partie 3: </a>
```

Voir aussi `preg_match` , `preg_replace` et `preg_split` .

6.88.12 preg_match() : Expression régulière standard.

`int` **preg_match** (`string pattern` , `string subject` , *`array matches`*)
[lien librairie]

`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..

`preg_match` retourne TRUE si la recherche réussit, et FALSE sinon (notamment en cas d'erreur).

Extraction d'un numéro de page d'une chaîne.

```php
<?php
if (preg_match("/page&#92;s+#(&#92;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
<?php
// &#92;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 ("/&#92;bweb&#92;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 ("/&#92;bweb&#92;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
<?php
// repérer le nom de l'hôte dans l'URL
preg_match("/^(http:&#92;/&#92;/)?([^&#92;/]+)/i",
"http://www.php.net/index.html", $matches);
$host = $matches[2];
// repérer les deux derniers segments du nom de l'hôte
preg_match("/[^&#92;.&#92;/]+&#92;.[^&#92;.&#92;/]+$/",$host,$matches);
echo "Le nom de domaine est : ".$matches[0]."&#92;n";
?>
```

Cet exemple va afficher :
`Le nom de domaine est : php.net`
Voir aussi `preg_match_all` , `preg_replace` et `preg_split` .

6.88.13 preg_quote() : Echappement des caractères spéciaux des expressions régulières.

`string` **preg_quote** (`string str` , *`string delimiter`*)
[lien librairie]

`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 &#92;$40 pour un g3&#92;/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'exepression régulière.
$textbody = "Ce livre est *très* difficile à trouver.";
$word = "*très*";
$textbody = preg_replace ("/".preg_quote($word)."/",
                          "<B>".$word."</B>",
                          $textbody);
?>
```

6.88.14 preg_replace_callback() : Rechercher/remplacer avec fonction de callback

`mixed` **preg_replace_callback** (`mixed pattern` , `mixed callback` , `mixed subject` , *`int limit`*)
[lien librairie]

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.

`preg_replace_callback` a été ajoutée en PHP 4.0.5.

Voir aussi `preg_replace` .

6.88.15 preg_replace() : Rechercher et remplacer par expression régulière standard.

`mixed` **preg_replace** (`mixed pattern` , `mixed replacement` , `mixed subject` , *`int limit`*)
[lien librairie]

`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.

Si la recherche n'aboutit à aucun résultat, `subject` sera inchangé.

Tous les paramètres de `preg_replace` peuvent être des tableaux.

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 fonctoin doit retourner la chaine de remplacement. Cette option a été ajoutée en PHP 4.0.4.

Remplacement de plusieurs valeurs

```
<?php
$patterns = array ("/(19|20)(&#92;d{2})−(&#92;d{1,2})−(&#92;d{1,2})/",
                   "/^&#92;s*{(&#92;w+)}&#92;s*=/");
$replace = array ("&#92;&#92;3/&#92;&#92;4/&#92;&#92;1&#92;&#92;2", "$&#92;&#92;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("/(<&#92;/?)(&#92;w+)([^>]*>/e", "'&#92;&#92;1'.strtoupper('&#92;&#92;2').'&#92;&#92;
?>
```

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
                 "'<[&#92;/&#92;!]*?[^<?>]*?>'si",  // Supprime les balises HTML
                 "'([&#92;r&#92;n])[&#92;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",
```

```
                "'&#(&#92;d+);'e");  // Evaluation comme PHP
$replace = array ("",
                  "",
                  "&#92;&#92;1",
                  "&#92;"",
                  "&",
                  "<",
                  ?>",
                  " ",
                  chr(161),
                  chr(162),
                  chr(163),
                  chr(169),
                  "chr(&#92;&#92;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` .

6.88.16 preg_split() : Eclatement d'une chaîne par expression régulière.

`array` **preg_split** (`string pattern` , `string subject` , *`int limit`* , *`int flags`*)
[lien librairie]

`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.

Note
Le paramètre `flags` a été ajouté en PHP Beta 3.

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);
?>
```

Voir aussi `explode` , `spliti` , `split` , `implode` , `preg_match` , `preg_match_all` et `preg_replace` .

6.89 qtdom

- qdom_error
- qdom_tree

6.89.2 qdom_tree() : Crée une arbre à partir d'un code XML

`object` **qdom_tree**`( string )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.90 Expressions régulières

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 inclues dans le répertoire de la distribution PHP.

Expressions régulières

```
<?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  $string. */
eregi("(ozilla.[23]|MSIE.3)",$HTTP_USER_AGENT);
/* Retourne
<TT>TRUE</TT>
 si  le client
   est Netscape 2, 3 ou MSIE 3. */
ereg("([[:alnum:]]+) ([[:alnum:]]+) ([[:alnum:]]+)",
     $string,$regs);
/* Introduit trois mots séparés par des espaces
   dans les chaînes $regs[1], $regs[2] et $regs[3]. */
$string = ereg_replace("^","<BR>",$string);
/* Insère une balise <BR> au début de la chaîne $string. */
$string = ereg_replace("$","<BR>",$string);
/* Insère une balise <BR> à la fin de la chaîne $string. */
$string = ereg_replace("\n","",$string);
/* Supprime toutes les nouvelles lignes de $string. */
?>
```

- ereg_replace
- ereg
- eregi_replace
- eregi
- split
- spliti
- sql_regcase

6.90.2 ereg() : Expression régulière standard.

`int` **ereg** (`string pattern` , `string string` , *`array regs`*)
[lien librairie]

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 `ereg`

```
<?php
if ( ereg( "([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $date, $regs ) ) {
    echo "$regs[3].$regs[2].$regs[1]";
} else {
    echo "Format de date invalide : $date";
}
?>
```

Voir aussi `eregi` , `ereg_replace` et `eregi_replace` .

6.90.3 eregi_replace() : Remplacement par expression régulière insensible à la casse.

`string` **eregi_replace** (`string pattern` , `string replacement` , `string string`)
[lien librairie]

`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` .

6.90.4 eregi() : Recherche par expression régulière insensible à la casse.

`int` **eregi** (`string pattern` , `string string` , *`array regs`*)
[lien librairie]

`eregi` est identique à `ereg` , hormis le fait qu'elle ignore la casse des caractère lors de la recherche sur les caractère alphabétiques.

Voir aussi `ereg` , `ereg_replace` et `eregi_replace` .

6.90.5 split() : Scinde une chaîne en un tableau, grâce à une expression régulière.

`array` **split** (`string pattern` , `string string` , *`int limit`*)
[lien librairie]

`split` retourne une 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
$passwd_list = 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 /, des points ou des tirets
list( $month, $day, $year ) = split( '[/.−]', $date );
echo "Mois: $month; Jour: $day; Annee: $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.

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 : `explode` et `implode` .

6.90.6 spliti() : Scinde une chaîne en un tableau, grâce à une expression régulière.

`array` **spliti** (`string pattern` , `string string` , *`int limit`*)
[lien librairie]

`spliti` est identique à `split` , hormis le fait qu'elle ignore la casse.

Voir aussi `split` , `explode` et `implode` .

6.90.7 sql_regcase() : Prépare une expression régulière pour effectuer une recherche insensible à la casse.

`string` **sql_regcase** (`string string`)
[lien librairie]

`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 remplacant 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.

6.91 Sémaphores et gestion de la mémoire partagée

6.91.1 Introduction

Ce module fourni 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 protégées 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
Ces fonctions ne sont pas accessibles sous Windows.

6.91.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.91.3 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` .

6.91.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
sysvmsg.value	"42"	PHP_INI_ALL
sysvmsg.string	"foobar"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.91.5 Types de ressources

6.91.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- sem_acquire
- sem_get
- sem_release
- shm_attach
- shm_detach
- shm_get_var
- shm_put_var
- shm_remove_var
- shm_remove

6.91.8 sem_get() : Retourne un identifiant de sémaphore.

`resource` **sem_get** (`int key` , *`int max_acquire`* , *`int perm`*)
[lien librairie]

`sem_get` retourne un identifiant positif de sémaphore en cas de succès, et FALSE en cas d'erreur.

`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.

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 `sem_acquire` et `sem_release` .

Note
`sem_get` n'est pas disponibles sous Windows.

6.91.9 sem_release() : Libère un sémaphore.

`int` **sem_release** (`int sem_identifier`)
[lien librairie]

Retourne TRUE en cas de succès, FALSE en cas d'erreur.

sem_release libère le sémaphore s'il a été réservé par le processus courant. Sinon, génère une erreur.

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 disponibles sous Windows.

6.91.10 shm_attach() : Crée ou ouvre un segment de mémoire partagée.

resource **shm_attach** (int key , *int memsize* , *int perm*)
[lien librairie]

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 mem_size (par défaut: sysvshm.init_mem dans php3.ini , sinon 10000 octets) et avec les permissions perm(par défaut : 666).

Aux appels suivants avec la même clé key , shm_attach retournera un nouvel identifiant, mais cet identifiant accédera toujours à la même portion de mémoire partagée. Dans ce cas, memsize et perm seront ignorés.

Note
Cette fonction n'est pas disponibles sous Windows.

6.91.11 shm_detach() : Libère un segment de mémoire partagée.

resource **shm_detach** (int shm_identifier)
[lien librairie]

shm_detach libère le segment de mémoire partagée identifié par shm_identifier et créé par sem_get . N'oubliez pas que cette mémoire partagée existe toujours sous Unix, et que les données sont toujours accessibles.

6.91.12 shm_get_var() : Lit une variable dans la mémoire partagée.

mixed **shm_get_var** (resource shm_identifier , int variable_key)
[lien librairie]

shm_get_var retourne la variable repérée par variable_key . La variable est toujours présente en mémoire partagée.

Note
shm_get_var n'est pas disponibles sous Windows.

6.91.13 shm_put_var() : Insère ou modifie une variable de la mémoire partagée.

`int` **shm_put_var** (`resource shm_identifier` , `int variable_key` , `mixed variable`)
[lien librairie]

`shm_put_var` insère ou modifie la variable `variable` avec la clé `variable_key` . Tous les types de variables (double, int, string, array, objects...) sont supportés.

Note
`shm_put_var` n'est pas disponibles sous Windows.

6.91.14 shm_remove_var() : Efface une variable de la mémoire partagée.

`int` **shm_remove_var** (`int shm_identifier` , `int variable_key`)
[lien librairie]

`shm_remove_var` efface la variable `variable_key` de la mémoire partagée et libère la mémoire.

Note
`shm_remove_var` n'est pas disponibles sous Windows.

6.91.15 shm_remove() : Supprime un segment de mémoire partagée sous Unix.

`int` **shm_remove** (`resource shm_identifier`)
[lien librairie]

`shm_remove` supprime un segment de mémoire partagée sous Unix. Toutes les données seront supprimées.

Note
`shm_remove` n'est pas disponibles sous Windows.

6.92 SESAM

6.92.1 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 systèmes SESAM n'est disponible que dans les dernières versions CVS de PHP3 . PHP 4 ne supporte pas la base de données SESAM.

6.92.2 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 librarie 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 filedoit 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).

```
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'es 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.

6.92.3 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`	Name of BS2000 PLAM library containing the loadable SESAM driver modules. Required for using SESAM functions. Example: `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 fontions SESAM.
	
	Exemple:
	
	`php3_sesam_configfile $SESAM.SESAM.CONF.AW`
	
	Ce fichier va généralement contenir une configuration comme celle−ci. (reportez vous au manuel de référence SESAM).
	
	`CNF=B`
	`NAM=K`
	`NOTYPE`

`php3_sesam_messagecatalog`	Nom du catalogue de messages SESAM. Dans la plupart des cas, cette directive n'est pas nécessaire. Uniquement dans les cas où le fichier de message n'est pas installé dans la table de message du système BS2000, vous pouvez utiliser cette directive : Example: `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 selectionnée 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 litéralement (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` .

6.92.4 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 effecutée à pleine vitesse.

Lorsque vous définissez explicitement le catalogue de messages SESAM, ce catalogue ser 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.

6.92.5 Types de curseurs

Les curseurs de résultat sont alloués pour les requêtes SQL de selection, 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.

6.92.6 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 juqu'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.

6.92.7 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.

6.92.8 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 faîtes. 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(tailles max)`).
- Mots reservés comme identifiants SQL.

  En SESAM (comme dans le standard SQL), les mots reservés 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 connait 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 retounrée 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.

6.92.9 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).

6.92.10 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−dessu, "pkey" prend l'index 0, et les trois colonnes "multi(1..3)" sont accessibles depuis les index 1 à 3.

6.92.11 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

6.92.13 sesam_commit() : Valide la transaction SESAM en cours

`bool` **sesam_commit** (`void`)
[lien librairie]

`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` .

Voir aussi : `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");
}
?>
```

6.92.14 sesam_connect() : Ouvre une connexion SESAM

`bool` **sesam_connect** (`string catalog` , `string schema` , `string user`)
[lien librairie]

`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` spéficie l'un des utilisateurs qui est autorisé à accéder à la comibinaison `catalog` et/ou `schema` . Notez que `user` est complétement indépendant de l'utilisateur système et des protections HTTP par mot de passe. Il n'apparait que dans la configuration SESAM.

Voir aussi `sesam_disconnect` .

Connexion à une base SESAM

```
<?php
if (! sesam_connect ("moncatalogue", "monschema", "toto")
     die("Impossible de se connecter à SESAM");
?>
```

6.92.15 sesam_diagnostic() : Retourne l'état de la dernière requête SESAM

`array` **sesam_diagnostic** (`void`)
[lien librairie]

`sesam_diagnostic` retourne un tableau associatif avec l'état et les codes de la dernière requête SQL. Les éléments du tableau sont :

Elément	Contenu
$array["sqlstate"]	code d'erreur à 5 chiffres (voir le manuel SESAM pour obtenir une description des valeurs possibles de SQLSTATE)
$array["rowcount"]	nombre de lignes affectées dans la dernière requête immédiate (update/insert/delete) : uniquement après une requête immédiate.
$array["errmsg"]	message d'erreur lisible : uniquement après une erreur
$array["errcol"]	numéro de colonne de la dernière erreur (indexée à partir de 0, −1 si indéfinies. uniquement après une erreur).
$array["errlin"]	numéro de ligne de la dernière erreur (indexée à partir de 0, −1 si indéfinies. uniquement après une erreur).

Dans l'exemple suivant, une erreur de syntaxe (E SEW42AE ILLEGAL CHARACTER) est affichée avec la requête SQL, et en désignant la position de l'erreur :

Afficher une erreur SESAM

```
<?php
// Fonction qui affiche un message d'erreur formaté
// en affichant la position de l'erreur dans le message d'erreur
function PrintReturncode($exec_str)
{
   $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=".$colspan."><FONT COLOR=red>ERROR:</FONT> ".
          htmlspecialchars($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"]) { // $lin is less or greater than errlin
         if (! ($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.

6.92.16 sesam_disconnect() : Déconnexion d'une base SESAM

`bool` **sesam_disconnect** (`void`)
[lien librairie]

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.

Voir aussi : sesam_connect .

Déconnexion d'une base SESAM

```
<?php
if (sesam_connect ("moncatalogue", "monschema", "toto")) {
... quelques requêtes et d'autres trucs ...
sesam_disconnect();
}
?>
```

6.92.17 sesam_errormsg() : retourne le message d'erreur

string **sesam_errormsg** (void)
[lien librairie]

sesam_errormsg retourne le message d'erreur SESAM associé à la dernière requête SQL.

```
<?php
if (!sesam_execimm($stmt))
  printf("%s<br>\n", sesam_errormsg());
?>
```

Voir aussi : sesam_diagnostic pour la liste complète des états de requêtes SQL.

6.92.18 sesam_execimm() : Exécute immédiatement une requête SQL

string **sesam_execimm** (string query)
[lien librairie]

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 fonctoin sesam_execimm . sesam_execimm mofidie 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 .

```
<?php
$stmt = "INSERT INTO matable VALUES('un', 'deux')";
$result = sesam_execimm ($stmt);
$err = sesam_diagnostic();
print("sqlstate = ".$err["sqlstate"]."\n".
      "Nombre de lignes affectées = ".$err["rowcount"]." == ".
      sesam_affected_rows($result)."\n");
?>
```

Voir aussi : sesam_query et sesam_affected_rows .

6.92.19 sesam_fetch_array() : Lit une ligne dans un tableau associatif

array **sesam_fetch_array** (string result_id , *int whence* , *int offset*)
[lien librairie]

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 selection 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 noms (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, faîtes un alias de la colonne.

```
SELECT TBL1.COL AS FOO, TBL2.COL AS BAR FROM TBL1, TBL2
```

Une gestion spéciale permet de lire les champs multiples, qui sinon, auraient toutes le même nom. Pour chaque colonne d'un champs 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.

```
CREATE TABLE ... ( ... MULTI(3) INT )
```

Les index associatifs utilisé pour les valeurs individuelles du champs multiple sont : `"multi(1)"` , `"multi(2)"` , et `"multi(3)"` , respectivement.

Les prochains appels à `sesam_fetch_array` liront la prochaine ligne (ou la précédente, ou la n−ième, suivant les attributs de défilement), jusqu'à ce qu'il n'y ait plus de lignes.

Exemple avec `sesam_fetch_array`

```
<?php
$result = sesam_query ("SELECT * FROM phone\n".
                       "  WHERE LASTNAME='".strtoupper($name)."'\n".
                       "  ORDER BY FIRSTNAME", 1);
if (! $result) {
    ... error ...
}
// Affiche la table
print "<TABLE BORDER>\n";
while (($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);
?>
```

Voir aussi : `sesam_fetch_row` qui retourne un tableau numérique.

6.92.20 sesam_fetch_result() : Retourne tout ou partie d'un résultat SESAM

`mixed` **sesam_fetch_result** (`string result_id` , *`int max_rows`*)
[lien librairie]

`sesam_fetch_result` retourne un tableau avec les lignes du résultat identifié par `result_id` , éventuellement limité à un maximum de `max_rows` Noteaz 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 `sesam_fetch_result` 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 `isset` . 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 une exemple complet avec `sesam_fetch_result` .

6.92.21 sesam_fetch_row() : Lit une ligne dans un tableau

`array` **sesam_fetch_row** (`string result_id` , *`int whence`* , *`int offset`*)
[lien librairie]

`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 élémen 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 parmètre 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 commencant 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 parmètre 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 conversion 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.

6.92.22 sesam_field_array() : Retourne des informations sur une colonne

`array` **sesam_field_array** (`string result_id`)
[lien librairie]

`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 $arr["count"]	Nombre total de colonnes dans le résultat (ou zéro si la requête était immédiate). Les champs multiples de SESAM sont linéarisés, et traités

	comme autant de colonnes.
string $arr[col]["name"]	Le nom de la colonne `col` , avec `col` qui vaut entre 0 et `$arr["count"]-1` . La valeur retournée peut être une chaîne vide (pour les colonnes dynamiquement générées). Les champs multiples SESAM sont linéarisés, et traités comme autant de colonnes, avec le même nom.
string $arr[col]["count"]	L'attribut "count" décrit le facteur de répétition quand la colonne a été déclarée comme un champs multiple. Généralement, cet attribut est à 1. La première colonne d'un champs multiple contient le nombre de répétitions, tandis que les colonnes suivantes ont un facteur de répétition mis à 1. Ceci peut être utilisé pour detecter les champs multiples. Reportez vous à l'exemple de la fonction `sesam_query` pour avoir un exemple d'utilisation.
string $arr[col]["type"]	Type de variable PHP pour les données de la colonne `col` , où `col` vaut de 0 à `$arr["count"]-1` . La valeur retournée peut être l'une de celles-ci : • "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 $arr[col]["sqltype"]	Type dedonnées SQL de la colonne `col` , où `col` vaut de 0 à `$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` .

6.92.23 sesam_field_name() : Retourne le nom d'une colonne

`int` **sesam_field_name** (`string result_id` , `int index`)
[lien librairie]

`sesam_field_name` retourne le nom du champs `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 detection des champs multiples.

6.92.24 sesam_free_result() : Libère les ressources

`int` **sesam_free_result** (`string result_id`)
[lien librairie]

`sesam_free_result` libère les ressources reservées par la requête `result_id` . retourne FALSE en cas d'erreur.

6.92.25 sesam_num_fields() : Retourne le nombre de colonne dans un résultat

`int` **sesam_num_fields** (`string result_id`)
[lien librairie]

Après avoir appelé `sesam_query` avec une requête de selection, `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 champs 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.

6.92.26 sesam_query() : Exécute une requête SESAM

`string` **sesam_query** (`string query` , *`bool scrollable`*)
[lien librairie]

`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

```
<?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'entête 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 entêtes de la table au dessus des champs multiples */
    if ($colattr["count"] > 1) {
      echo "  <TH COLSPAN=".$colattr["count"].">".$colattr["name"].
           "(1..".$colattr["count"].")</TH>\n";
      $col += $colattr["count"] - 1;
    }
    else
      echo "  <TH>" . $colattr["name"] . "</TH>\n";
  }
  echo " </TR>\n";
}
do {
  // Lit les résultats par bloc de 100
  $ok = sesam_fetch_result($result,100);
  for ($row=0; $row < $ok["rows"]; ++$row) {
    echo " <TR>\n";
    for ($col = 0; $col < $ok["cols"]; ++$col) {
      if (isset($ok[$col][$row]))
        echo "  <TD>" . $ok[$col][$row] . "</TD>\n";
      else
        echo "  <TD>-empty-</TD>\n";
    }
    echo " </TR>\n";
  }
} while ($ok["truncated"]); // tant qu'il a y encore des données
echo "</TABLE>\n";
// libère les ressources
sesam_free_result($result);
```

```
?>
```

6.92.27 sesam_rollback() : Annule une transaction SESAM

`bool` **sesam_rollback** (`void`)
[lien librairie]

`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.

Voir aussi : `sesam_commit` .

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();
}
?>
```

6.92.28 sesam_seek_row() : Déplace un curseur à défilement

`bool` **sesam_seek_row** (`string result_id` , `int whence` , *`int offset`*)
[lien librairie]

`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 commencant 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 .

6.92.29 sesam_settransaction() : Modifie les paramètres de transaction SESAM

bool **sesam_settransaction** (int isolation_level , int read_only)
[lien librairie]

sesam_settransaction retourne TRUE si les valeurs sont valides et que a 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	Lecutre avec validation
3	SESAM_TXISOL_REPEATABLE_READ	Lecture récurente
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 spécifiée dans le fichier de configuration SESAM .

Modifier les paramètres de configuration SESAM

```
<?php
sesam_settransaction(SESAM_TXISOL_REPEATABLE_READ,
                     SESAM_TXREAD_READONLY);
?>
```

6.93 Sessions

6.93.1 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.

6.93.2 Sessions and security

Using sessions, does not mean, you can be absolutely sure, that the session data can only be viewed by that user. This is important to keep in mind, when storing and displaying sensitive information. When storing data into a session, one should always ask themselves, what the damage is, when somebody else views that information, or how your application is affected when this session is actually somebody else.

For instance, if somebody else takes a session, can he then post a message in a forum, as that user and how big of a problem is that? Or perhaps he can view what the original user was thinking of ordering, because he gets access to that user's shopping cart. Obviously for a flowershop, this is less dramatic, than for a pharmacy.

Therefore, when dealing with sensitive information, there should always be additional methods to decide whether it is a valid session. Sessions are not reliable as a secure authentication mechanism.

Sessions rely on the session ID, meaning one can 'steal' a session, by stealing the session ID. This can be made harder, by using a cookie specifically a session cookie, but does not in any way make it impossible and still relies on the user closing all browser windows, to expire the session cookie. Besides that, even session cookies can be sniffed on a network or logged by a proxyserver.

6.93.3 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.93.4 Installation

Le suport 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.

6.93.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
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	"0"	PHP_INI_SYSTEM|PHP_INI
url_rewriter.tags	"a=href,area=href,frame=src,input=src,form=fakeentry"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

- `session.save_handler` 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` 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 poru que les fonctions de sessions de PHP fonctionnent. Indiquez un chemin de dossier valide, comme par exemple : `c:/temp` .

- `session.name` 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` spécifie si le module de session doit démarrer automatiquement, au début de chaque script PHP. Par défaut, c'est `0` (inactivé).
- `session.cookie_lifetime` spécifie la durée de vie du cookie en secondes. La valeur de `0` siginifie : "Jusqu'à ce que le navigateur soit éteind". La valeur par défaut est : 0.
- `session.serialize_handler` 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` specifies the probability that the gc (garbage collection) routine is started on each request in percent. Defaults to `1` .
- `session.gc_maxlifetime` 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` contient une sous−chaîne que vous souhaitez retrouver dans tous les entêtes HTTP Referer. Si cet entêtes 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` 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` 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` 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` spécifie si le module doit utiliser **seulement** les cookies pour stocker les identifiants de sessions du coté 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` spécifie le chemin utilisé lors de la création du cookie. Par défaut, il vaut `/` .
- `session.cookie_domain` spécifie le domaine utilisé lors de la création du cookie. Par défaut, il ne vaut rien.
- `session.cache_limiter` 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` 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` indique si le support transparent du SID est activé, grâce à l'option de compilation `--enable-trans-sid` . Par défaut, il vaut `0` (inactivé).

Note
En PHP 4.1.2 ou plus ancient, 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.

- `url_rewriter.tags` 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.

6.93.6 Types de ressources

Cette extension ne définit aucune ressource.

6.93.7 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"` .

6.93.8 Exemples

Note
Depuis PHP 4.1.0, `$_SESSION` est disponible comme variable globale, au même titre que `$_POST` , `$_GET` , `$_REQUEST` , etc.. Contrairement à `$HTTP_SESSION_VARS` , `$_SESSION` est toujours globale. Par conséquent, vous n'avez pas besoin d'utiliser le mot réservé `global` avec `$_SESSION` . Notez que cette documentation a été modifiée pour utiliser `$_SESSION` . Vous pouvez toujours le remplacer par `$HTTP_SESSION_VARS` si vous préférez l'ancienne version. Les clés du tableau `$_SESSION` sont sujettes aux mêmes limitations que les variables PHP habituelles, c'est à dire qu'elles ne peuvent pas commencer par un nombre, mais commencer par une lettre ou un souligné '_'. Pour plus de détails, reportez vous à la section sur les variables .

Si `track_vars` est activé et `register_globals` est désactivé, seuls les éléments du tableau

global $_SESSION contiendront les variables enregistrées dans la session. Les variables de sessions relues seront uniquement disponibles dans $_SESSION .

L'utilisation de $_SESSION (ou $HTTP_SESSION_VARS avec PHP 4.0.6 et plus ancien) est recommandé pour une meilleure sécurité et un code plus facilement 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 $_SESSION.

```
<?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']++;
}
?>
```

Retirer une variable de session avec $_SESSION est register_globals inactif.

```
<?php
session_start();
// Utilisez $HTTP_SESSION_VARS avec PHP 4.0.6 ou plus ancien
unset($_SESSION['compteur']);
?>
```

Retirer une variable de session avec $_SESSION est register_globals activé, après l'avoir enregistré avec $_SESSION.

```
<?php
session_start();
// Avec PHP 4.3 et plus récent, vous pouvez simplement utiliser l'exemple précédent
session_unregister('compteur');
?>
```

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. Inactiver `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 `$_SESSION` ne fera pas de référence vers la même valeur jusqu'à la prochaine utilisation de `session_start` (ceci s'applique à PHP 4.2 est avant seulement). C'est à dire qu'une modification dans les variables globales ne seront pas répercutés dans les entrées de `$_SESSION` . Il est peu probable que cela ait un impact en pratique, et de plus, cela a été corrigé en PHP 4.3.

6.93.9 Passer l'identifiant de session (session ID)

Il y a deux méthodes de propagation de l'identifiant de session :

- Cookies
- Paramèter 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` .

Counting the number of hits of a single user

```
<?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</A>.
```

`<?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).

6.93.10 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 and security
- 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_readonly
- session_register
- session_save_path
- session_set_cookie_params
- session_set_save_handler
- session_start
- session_unregister
- session_unset
- session_write_close

6.93.12 session_cache_limiter() : Lit et/ou modifie le limiteur de cache

`string` **session_cache_limiter** (*`string cache_limiter`*)
[lien librairie]

`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 entêtes HTTP envoyés au client. Certains entêtes détermine les règles de mise en cache de la page sur le nivagateur. 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'entête Expire envoyé au client peut poser des problèmes à certains navigateurs, comme notamment Moziall. Vous pouvez éviter ce problème avec le mode `private_no_expire` . L'entête Expire n'est jamais envoyé au navigateur pour ce mode.

Note
`private_no_expire` a été ajouté en PHP 4.2.0dev.

Le limiteur de cache est remis à la valeur par défaut de `session.cache_limiter` à chaque démarrage de script HPP. Donc, vuos devrez appeler `session_cache_limiter` à chaque page, et avant `session_start` .

Exemple avec `session_cache_limiter`

```
<?php

// configure le limiteur de cache à 'private'

session_cache_limiter('private');
$cache_limiter = session_cache_limiter();

echo "Le limiteur de cache vaut maintenant $cache_limiter<p>";
?>
```

6.93.13 session_decode() : Décode les données de session

`bool` **session_decode** (`string data`)
[lien librairie]

`session_decode` décode les données de sessions fournies dans le paramètre `data` , et défini les variables ainsi contenues dans la session.

6.93.14 session_destroy() : Détruit une session

`bool` **session_destroy** (`void`)
[lien librairie]

`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 écrite 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
<?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
<?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();

?>
```

6.93.15 session_encode() : Encode les données de session

`string` **session_encode** (`void`)
[lien librairie]

`session_encode` retourne une chaîne contenant les variables de la session courante encodé, prêt au stockage.

6.93.16 session_get_cookie_params() : Lit la configuration du cookie de session

`array` **session_get_cookie_params** (`void`)
[lien librairie]

`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).

6.93.17 session_id() : Let et/ou modifie l'identifiant courant de session

`string` **session_id** (*`string id`*)
[lien librairie]

`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 consante `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` .

6.93.18 session_is_registered() : Vérifie si une variable est enregistrée dans la session

`bool` **session_is_registered** (`string name`)
[lien librairie]

`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 ancient) 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 .

6.93.19 session_module_name() : Lit et/ou modifie le module de session courant

string **session_module_name** (*string module*)
[lien librairie]

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.

6.93.20 session_name() : Lit et/ou modifie le nom de la session

string **session_name** (*string name*)
[lien librairie]

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 caractère 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 session_name

```
<?php

// choisi le nom de SiteWebSession

$previous_name = session_name("SiteWebSession");

echo "L'ancien nom de la session était $previous_name<p>";
?>
```

6.93.21 session_readonly() : Initialise une session en mode lecture

void **session_readonly** (void)
[lien librairie]

session_readonly lit les données de session sans verrouiller ces données. La modification de données de session n'est donc pas possible, mais les performances sont nettement supérieures.

6.93.22 session_register() : Enregistre une variable dans une session

bool **session_register** (mixed name , *mixed ...*)
[lien librairie]

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. Poru chaque nom, session_register place la variable dans la session courante, pour la sauvegarde de fin de script.

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 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 sessio, simplement en ajoutant l'index approprié dans la variable $_SESSION ou $HTTP_SESSION_VARS .

```
<?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 de3s 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 .

6.93.23 session_save_path() : Lit et/ou modifie le chemin de sauvegarde des sessions

string **session_save_path** (*string path*)
[lien librairie]

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.

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.

6.93.24 session_set_cookie_params() : Modifie les paramètres du cookie de session

void **session_set_cookie_params** (int lifetime , *string path* , *string domain* , *bool secure*)
[lien librairie]

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 pour le script courant.

Note
Le paramètre secure aé ét ajouté en PHP 4.0.4.

6.93.25 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)
[lien librairie]

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
Vous devez configurer l'option session.save_handler à user dans votre fichier php.ini pour

que `session_set_save_handler` soit actif.

Note
Le gestionnaire d'écriture n'est pas exécuté tant que le flot de sortie est ouvert. Par conséquent, l'affichage de messages de debogage 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, pour l'échec.

Exemple avec `session_set_save_handler`

```php
<?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 obsolètes ici.     *
 *****************************************/
function gc ($maxlifetime) {
  return true;
}

session_set_save_handler ("open", "close", "read", "write", "destroy", "gc");

session_start();

// Utilisez vos sessions comme d'habitude
?>
```

6.93.26 session_start() : Initialise une session

`bool` **session_start** (`void`)
[lien librairie]

`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 nivagateur.

`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` .

6.93.27 session_unregister() : Supprime une variable de la session

`bool` **session_unregister** (`string name`)
[lien librairie]

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 puor 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 .

6.93.28 session_unset() : Détruit toutes les variables de session

void **session_unset** (void)
[lien librairie]

session_unset détruit toutes les variables de la session courante.

Note
Si vous utilisez $_SESSION (ou $HTTP_SESSION_VARS), utilisez plutôt ceci : $_SESSION = array();

6.93.29 session_write_close() : Ecrit les données de session et ferme la session

void **session_write_close** (void)
[lien librairie]

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.

Voir aussi session_readonly .

6.94 Mémoire partagée

Shmop est un ensemble de fonctions simples pour gérer la mémoire partagée avec PHP (lecture, écriture, création et suppressions de segments de mémoire paragée UNIX). Ces fonctions ne fonctionnent pas sous Windows, car ce système d'exploitation ne supporte pas la mémoire partagée. Pour utiliser les fonctions shmop, compilez PHP avec l'option −−enable−shmop parameter.

Note
Toutes les fonctions décrites ci−dessous commencent par `shm_` poru les versions jusqu'à PHP 4.0.3, mais en PHP 4.0.4 et plus récent, elles sont préfixées par `shmop_` .

Introduction à la mémoire partagée

```
<?php
// Crée 100 octets de mémoire partagée avec
// un identifiant système "0xff3"
$shm_id = shmop_open(0xff3, "c", 0644, 100);
if(!$shm_id) {
    echo "Impossible de créer la mémoire partagée\n";
}
// Lire la taille de la mémoire partagée
$shm_size = shmop_size($shm_id);
echo "Un bloc de SHM de taille ".$shm_size. " a été créé.\n";
// Ecriture d'une chaîne de test dans ce segment
$shm_bytes_written = shmop_write($shm_id, "mon bloc de mémoire partagée", 0);
if($shm_bytes_written != strlen("mon bloc de mémoire partagée")) {
    echo "Impossible d'écrire toutes les données en mémoire\n";
}
// Lecture du segment
$my_string = shmop_read($shm_id, 0, $shm_size);
if(!$my_string) {
    echo "Impossible de lire toutes les données en mémoire\n";
}
echo "Les données mis en mémoire partagées sont : ".$my_string."\n";
//Maintenant, effacons le bloc, et fermons le segment de mémoire
if(!shmop_delete($shm_id)) {
    echo "Impossible d'effacer le segment de mémoire";
}
shmop_close($shm_id);
?>
```

- shmop_close
- shmop_delete
- shmop_open
- shmop_read
- shmop_size
- shmop_write

6.94.2 shmop_delete() : Détruit un bloc de mémoire partagée

`int` **shmop_delete** (`resource shmid`)
[lien librairie]

`shmop_delete` sert à détruire un bloc de mémoire partagée.

shmop_delete prend un identifiant de mémoire partagée shmid , 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 .

6.94.3 shmop_open() : Crée ou ouvre un bloc de mémoire partagée

`resource` **shmop_open** `( int key , string flags , int mode , int size )`
[lien librairie]

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 héxadécimal. 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.

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 ocets.

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

`<?php`
`$shm_id = shmop_open(0x0fff, "c", 0644, 100);`
`?>`

Cet exemple ouvre un nouveau bloc de mémoire partagée, dont l'identifiant est 0x0fff.

6.94.4 shmop_read() : Lit un bloc

`string` **shmop_read** `( resource shmid , int start , int count )`
[lien librairie]

shmop_read lit une chaîne dans une bloc de mémoire partagée.

shmop_read prend 3 paramètres: shmid , qui est un identifiant de mémoire partagée, créé par

shmop_open , start qui est la position à partir de laquelle on commence à lire dans la mémoire et count , le nombre d'octets à lire.

Lire un bloc de mémoire partagée

```
<?php
$shm_data = shmop_read($shm_id, 0, 50);
?>
```

Cet exemple lit 50 octets dans le bloc de mémoire partagée $shm_data .

6.94.5 shmop_size() : Lire la taille du bloc de mémoire partagée

int **shmop_size** (resource shmid)
[lien librairie]

shmop_size sert à connaître la taille, en octets, d'un bloc de mémoire partagée.

shmop_size prend comme argument shmid , un identifiant de bloc de mémoire partagée créé par shmop_open , et retourne un entier, qui représente le taille de ce bloc.

Lire la taille d'un bloc de mémoire partagée

```
<?php
$shm_size = shmop_size($shm_id);
?>
```

Cet exemple lit la taille du bloc identifié par $shm_id , et le place dans $shm_size .

6.94.6 shmop_write() : Ecrire dans un bloc de mémoire partagée

int **shmop_write** (resource shmid , string data , int offset)
[lien librairie]

shmop_write écrit une chaîne dans un bloc de mémoire partagée.

shmop_write prend 3 paramètres: shmid , qui est un identifiant de mémoire partagée, créé par shmop_open , data qui est la chaîne à écrire dans la mémoire et offset , la position à partir de laquelle il faut commencer à écrire.

Ecrire un bloc de mémoire partagée

```
<?php
$shm_bytes_written = shmop_write($shm_id, $my_string, 0);
?>
```

Cet exemple écrit les données de la chaîne $my_string dans un bloc de mémoire partagée. $shm_bytes_written représentera le nombre d'octets écrits.

6.95 Shockwave Flash

6.95.1 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 .

6.95.2 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/sgi/graphics/grafica/flash .

6.95.3 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.

6.95.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.95.5 Types de ressources

Cette extension ne définit aucune ressource.

6.95.6 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)

6.95.7 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 Prédefinies
- 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_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

6.95.9 swf_actiongotoframe() : Joue un frame puis stoppe.

`void` **swf_actiongotoframe** (`int framenumber`)
[lien librairie]

`swf_actiongotoframe` se déplace jusqu'au frame `framenumber` , le joue, puis s'arrête.

6.95.10 swf_actiongotolabel() : Affiche le frame nommé.

`void` **swf_actiongotolabel** (`string label`)
[lien librairie]

`swf_actiongotolabel` affiche le frame de nom `label` , puis stoppe.

6.95.11 swf_actionnextframe() : Avance d'un frame.

`void` **swf_actionnextframe** (`void`)
[lien librairie]

swf_actionnextframe avance d'un frame le frame courant.

6.95.12 swf_actionplay() : Joue l'animation flash à partir du frame courant.

void **swf_actionplay** (void)
[lien librairie]

swf_actionplay joue l'animation Flash à partir du frame courant.

6.95.13 swf_actionprevframe() : Recule d'un frame.

void **swf_actionprevframe** (void)
[lien librairie]

swf_actionnextframe recule d'un frame le frame courant.

6.95.14 swf_actionsettarget() : Fixe le contexte des actions.

void **swf_actionsettarget** (string target)
[lien librairie]

swf_actionsettarget fixe le contexte des actions. Vous pouvez utiliser cette fonction pour contrôler d'autres animations Flash qui seraient en fonctionnement.

6.95.15 swf_actionstop() : Arrête l'animation flash.

void **swf_actionstop** (void)
[lien librairie]

swf_actionstop arrête l'animation Flash au frame courant.

6.95.16 swf_actiontogglequality() : Choisit le niveau de qualité haut ou bas.

void **swf_actiontogglequality** (void)
[lien librairie]

swf_actiontogglequality modifie le niveau de qualité haut ou bas.

6.95.17 swf_actionwaitforframe() : Ignore les actions si le frame n'est pas chargé.

void **swf_actionwaitforframe** (int framenumber , int skipcount)
[lien librairie]

swf_actionwaitforframe vérifie que le frame framenumber 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...".

6.95.18 swf_addbuttonrecord() : Contrôle la situation, l'aparance et la zone active du bouton courant.

void **swf_addbuttonrecord** (int states , int shapeid , int depth)
[lien librairie]

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'apparence 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

```
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 ();
```

6.95.19 swf_addcolor() : Fixe la couleur globale d'addition (? : the global add color).

void **swf_addcolor** (float r , float g , float b , float a)
[lien librairie]

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 ajouté à rgba lorsque l'objet est placé sur la scène.

Note

Les valeurs de rgba peuvent être positives ou négatives.

6.95.20 swf_closefile() : Ferme le fichier courant Shockwave Flash.

void **swf_closefile** (*int return_file*)
[lien librairie]

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
<?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);
}
?>
```

6.95.21 swf_definebitmap() : Définit une image bitmap.

`void` **swf_definebitmap** (`int objid` , `string image_name`)
[lien librairie]

`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.

6.95.22 swf_definefont() : Définit une police.

`void` **swf_definefont** (`int fontid` , `string fontname`)
[lien librairie]

`swf_definefont` définit la police `fontname` et lui affecte l'identifiant `fontid` . Cette police devient alors la police courante.

6.95.23 swf_defineline() : Définit une ligne.

`void` **swf_defineline** (`int objid` , `float x1` , `float y1` , `float x2` , `float y2` , `float width`)
[lien librairie]

`swf_defineline` définit une ligne commençant aux coordonnées (`x1` , `y1`), et finissant au point de coordonnées (`x2` , `y2`). Elle aura la largeur de `width` .

6.95.24 swf_definepoly() : Définit un polygone.

`void` **swf_definepoly** (`int objid` , `array coords` , `int npoints` , `float width`)
[lien librairie]

`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.

6.95.25 swf_definerect() : Définit un rectangle.

`void` **swf_definerect** (`int objid` , `float x1` , `float y1` , `float x2` , `float y2` , `float width`)
[lien librairie]

`swf_definerect` définit un rectangle, de coin supérieur gauche aux coordoonées (`x1` , `y1`), et de coin inférieur droit aux coordonnées (`x2` , `y2`). L'épaisseur des bords est données par le paramètre `width` . `width` , 0.0 le rectangle sera rempli.

6.95.26 swf_definetext() : Définit une chaîne de texte.

`void` **swf_definetext** (`int objid` , `string str` , `int docenter`)
[lien librairie]

`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.

6.95.27 swf_endbutton() : Termine la définition du bouton courant.

`void` **swf_endbutton** (`void`)
[lien librairie]

swf_endbutton termine la définition du bouton courant.

6.95.28 swf_enddoaction() : Termine l'action courante.

void **swf_enddoaction** (void)
[lien librairie]

swf_startdoaction termine l'action courante, démarrée par swf_startdoaction .

6.95.29 swf_endshape() : Complète la définition de la forme courante.

void **swf_endshape** (void)
[lien librairie]

swf_endshape complète la définition de la forme courante.

6.95.30 swf_endsymbol() : Termine la définition de symbole.

void **swf_endsymbol** (void)
[lien librairie]

swf_endsymbol termine la définition de symble, qui a été commencée avec swf_startsymbol .

6.95.31 swf_fontsize() : Change la taille de la police.

void **swf_fontsize** (float size)
[lien librairie]

swf_fontsize remplace la taille de la police par la taille size .

6.95.32 swf_fontslant() : Change l'inclinaison de la police courante.

void **swf_fontslant** (float slant)
[lien librairie]

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.

6.95.33 swf_fonttracking() : Change l'espacement des caractères.

void **swf_fonttracking** (float tracking)
[lien librairie]

swf_fonttracking 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.

6.95.34 swf_getbitmapinfo() : Lit les informations sur une image.

`array` **swf_getbitmapinfo** (`int bitmapid`)
[lien librairie]

`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.

6.95.35 swf_getfontinfo() : Retourne la hauteur du A majuscule, et du x minuscule.

`array` **swf_getfontinfo** (`void`)
[lien librairie]

`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.

6.95.36 swf_getframe() : Retourne le numéro de frame courant.

`int` **swf_getframe** (`void`)
[lien librairie]

`swf_getframe` retourne le numéro de frame courant.

6.95.37 swf_labelframe() : Nomme le frame courant.

`void` **swf_labelframe** (`string name`)
[lien librairie]

`swf_labelframe` donne le nom `name` au frame courant.

6.95.38 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`)
[lien librairie]

`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.

6.95.39 swf_modifyobject() : Modifie un objet.

`void` **swf_modifyobject** (`int depth` , `int how`)
[lien librairie]

`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.

6.95.40 swf_mulcolor() : Fixe la couleur globale de multiplication (? : the global multiply color).

`void` **swf_mulcolor** (`float r` , `float g` , `float b` , `float a`)
[lien librairie]

`swf_mulcolor` fixe la valeur globale de multiplication (the global multiply color...) à la couleur `rgba` . Cette couleur est utilisée (implicitement) par `swf_placeobject` , `swf_modifyobject` et `swf_addbuttonrecord` . La couleur d'un objet sera multipliée par `rgba` lorsque l'objet est placé sur la scène.

Note
Les valeurs de `rgba` peuvent être positives ou négatives.

6.95.41 swf_nextid() : Retourne le prochain identifiant d'objet libre.

`int` **swf_nextid** (`void`)
[lien librairie]

`swf_nextid` retourne le prochain identifiant d'objet libre.

6.95.42 swf_oncondition() : Décrit une transition utilisée pour déclencher une liste d'actions.

`void` **swf_oncondition** (`int transition`)
[lien librairie]

`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 (OverUptoIdle|OverDowntoOutDown)

6.95.43 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`)
[lien librairie]

`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 ultérieur).

6.95.44 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`)
[lien librairie]

`swf_ortho` définit une projection orthogonale entre les coordonnées utilisateur et le port courant.

6.95.45 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`)
[lien librairie]

`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` .

6.95.46 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`)
[lien librairie]

`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.

6.95.47 swf_placeobject() : Place un objet sur la scène.

`void` **swf_placeobject** (`int objid` , `int depth`)
[lien librairie]

`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.

6.95.48 swf_polarview() : Défini le point de vue de l'utilisateur en coordonnées polaire.

`void` **swf_polarview** (`float dist` , `float azimuth` , `float incidence` , `float twist`)
[lien librairie]

`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.

6.95.49 swf_popmatrix() : Dépile la matrice de transformation.

`void` **swf_popmatrix** (`void`)
[lien librairie]

`swf_popmatrix` dépile la matrice de transformation.

6.95.50 swf_posround() : Active l'approximation des translation d'objets.

`void` **swf_posround** (`int round`)
[lien librairie]

`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).

6.95.51 swf_pushmatrix() : Empile la matrice de transformation courante dans la pile.

`void` **swf_pushmatrix** (`void`)
[lien librairie]

`swf_pushmatrix` empile la matrice de transformation courante dans la pile.

6.95.52 swf_removeobject() : Enlève un objet.

`void` **swf_removeobject** (`int depth`)
[lien librairie]

`swf_removeobject` enlève l'objet situé à la profondeur `depth` de la scène.

6.95.53 swf_rotate() : Rotation de la transformation courante.

`void` **swf_rotate** (`float angle` , `string axis`)
[lien librairie]

`swf_rotate` fait subir la roation d'angle `angle` , autour de l'axe `axis` . Les valeurs possibles pour `axis` sont : 'x' (axe x), 'y' (axe y) ou 'z' (axe z).

6.95.54 swf_scale() : Homothétie.

`void` **swf_scale** (`float x` , `float y` , `float z`)
[lien librairie]

`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.

6.95.55 swf_setfont() : Change la police courante.

`void` **swf_setfont** (`int fontid`)
[lien librairie]

`swf_setfont` remplace la police courante par la police repérée par l'identifiant `fontid` .

6.95.56 swf_setframe() : Fixe le frame courant.

`void` **swf_setframe** (`int framenumber`)
[lien librairie]

`swf_setframe` sélectionne le frame `framenumber` comme frame actif.

6.95.57 swf_shapearc() : Dessine une arc de cercle.

`void` **swf_shapearc** (`float x` , `float y` , `float r` , `float ang1` , `float ang2`)
[lien librairie]

`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` .

6.95.58 swf_shapecurveto() : Dessine une courbe de Bézier quadratique entre deux points.

`void` **swf_shapecurveto** (`float x1` , `float y1` , `float x2` , `float y2`)
[lien librairie]

`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`).

6.95.59 swf_shapecurveto3() : Dessine une courbe Bézier cubique.

`void` **swf_shapecurveto3** (`float x1` , `float y1` , `float x2` , `float y2` , `float x3` , `float y3`)
[lien librairie]

Dessine une courbe de Bézier cubique, en utilisant les points de coordoonnées (`x1` , `y1`) et (`x2` , `y2`) comme points de contrôle et le point de coordonnées (`x3` , `y3`) comme point final. La position finale devient alors la position courante.

6.95.60 swf_shapefillbitmapclip() : Choisit le mode de remplissage par texture.

`void` **swf_shapefillbitmapclip** (`int bitmapid`)
[lien librairie]

Choisit le mode de remplissage par texture : les espaces vides seront remplis avec la bitmap `bitmapid` .

6.95.61 swf_shapefillbitmaptile() : Choisit le mode de remplissage par texture repétée.

`void` **swf_shapefillbitmaptile** (`int bitmapid`)
[lien librairie]

Choisit le mode de remplissage par texture : les espaces vides seront remplis avec la bitmap `bitmapid` , repétée autant de fois qu'il le faut (mode carrelage).

6.95.62 swf_shapefilloff() : Inactive le remplissage.

`void` **swf_shapefilloff** (`void`)
[lien librairie]

`swf_shapefilloff` inactive le remplissage pour la forme courante.

6.95.63 swf_shapefillsolid() : Fixe la couleur pour le style courant de remplissage.

`void` **swf_shapefillsolid** (`float r` , `float g` , `float b` , `float a`)
[lien librairie]

`swf_shapefillsolid` fixe la couleur pour le style courant de remplissage à `rgba` .

6.95.64 swf_shapelinesolid() : Fixe le style courant de ligne.

`void` **swf_shapelinesolid** (`float r` , `float g` , `float b` , `float a` , `float width`)
[lien librairie]

`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.

6.95.65 swf_shapelineto() : Dessine une ligne.

`void` **swf_shapelineto** (`float x` , `float y`)
[lien librairie]

`swf_shapelineto` dessine une ligne entre la position courante et le point de coordonnées (`x` , `y`). La position courante devient alors (`x` , `y`).

6.95.66 swf_shapemoveto() : Change la position courante.

`void` **swf_shapemoveto** (`float x` , `float y`)
[lien librairie]

`swf_shapemoveto` fixe la position courante au point de coordonnées (`x` , `y`).

6.95.67 swf_showframe() : Affiche le frame courant.

`void` **swf_showframe** (`void`)
[lien librairie]

`swf_showframe` affiche le frame courant.

6.95.68 swf_startbutton() : Commence la définition d'un bouton.

`void` **swf_startbutton** (`int objid` , `int type`)
[lien librairie]

`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.

6.95.69 swf_startdoaction() : Commence la description d'une liste d'action pour la frame courante.

`void` **swf_startdoaction** (`void`)
[lien librairie]

`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.

6.95.70 swf_startshape() : Commence une forme complexe.

`void` **swf_startshape** (`int objid`)
[lien librairie]

`swf_startshape` commence une forme complexe, qui sera reperé par l'identifiant d'objet `objid` .

6.95.71 swf_startsymbol() : Définit un symbole.

`void` **swf_startsymbol** (`int objid`)
[lien librairie]

`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.

6.95.72 swf_textwidth() : Retourne la longueur d'une chaîne.

`float` **swf_textwidth** (`string str`)
[lien librairie]

`swf_textwidth` retourne la longueur de la chaîne `str` , en pixels, en utilisant la police courante.

6.95.73 swf_translate() : Translate la transformation courante.

`void` **swf_translate** (`float x` , `float y` , `float z`)
[lien librairie]

`swf_translate` déplace la transformation courante de `x` , `y` et `z` , dans les directions x, y et z.

6.95.74 swf_viewport() : Sélectionne une nouvelle zone pour un dessin ultérieur.

`void` **swf_viewport** (`float xmin` , `float xmax` , `float ymin` , `float ymax`)
[lien librairie]

`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.

6.96 SNMP

Afin de pouvoir utiliser les fonctions SNMP sous Unix, vous aurez besoin d'installer le package UCD SNMP . Sous Windows ces fonctions ne sont disponibles que sous NT, et pas sous Win95/98.

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
```

Si vous avez des erreurs "segmentation faults", lors de l'utilisation des commandes SNMP , c'est que vous n'avez pas suivi les recommendations précédentes. Si vous ne voulez pas recompiler UCD SNMP , vous pouvez aussi recompiler PHP avec l'option −−enable−ucd−snmp−hack qui évitera cette erreur.

- snmp_get_quick_print
- snmp_set_quick_print
- snmpget
- snmprealwalk
- snmpset
- snmpwalk
- snmpwalkoid

6.96.2 snmp_set_quick_print() : Ecrit la valeur courante de l'option quick_print de la librairie UCD.

`void` **snmp_set_quick_print** (`boolean quick_print`)
[lien librairie]

`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.

```
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 .

6.96.3 snmpget() : Reçoit un objet SNMP .

`string` **snmpget** (`string hostname` , `string community` , `string object_id` , *`int timeout`* , *`int retries`*)
[lien librairie]

`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` .

```
$syscontact = snmpget("127.0.0.1", "public", "system.SysContact.0")
```

6.96.4 snmprealwalk() : Retourne tous les objets, y compris leur ID d'objet

`array` **snmprealwalk** (`string host` , `string community` , `string object_id` , *`int timeout`* , *`int retries`*)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.96.5 snmpset() : Envoie un objet SNMP.

`bool` **snmpset** (`string hostname` , `string community` , `string object_id` , `string type` , `mixed value` , *`int timeout`* , *`int retries`*)
[lien librairie]

`snmpset` modifie la valeur de l'objet SNMP spécifié, en retournant TRUE en cas de succès et FALSE en cas d'erreur.

`snmpget` 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` .

6.96.6 snmpwalk() : Reçoit tous les objets SNMP d'un agent.

array **snmpwalk** (string hostname , string community , string object_id , *int timeout* , *int retries*)
[lien librairie]

snmpwalk retourne un tableau d'objets SNMP , en commencant 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.

```
<?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.

```
<?php
for ($i=0; $i<count($a); $i++) {
    echo $a[$i];
}
?>
```

6.96.7 snmpwalkoid() : Demande d'informations d'arbre sur une entité du réseau.

array **snmpwalkoid** (string hostname , string community , string object_id , *int timeout* , *int retries*)
[lien librairie]

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 compatbilité ascendante.

```
<?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 :

```
for (reset($a); $i = key($a); next($a)) {
    echo "$i: $a[$i]<br>\n";
}
```

6.97 Socket

6.97.1 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 utiliser ce module à vos risques et périls.

6.97.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.97.3 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` .

6.97.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.97.5 Types de ressources

Cette extension ne définit aucune ressource.

6.97.6 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)

6.97.7 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` généré 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`

6.97.8 Exemples

Exemple de socket : serveur TCP/IP simple
Cet exemple montre comment créer un serveur simple. Changez les variables `address` et `port` pour vous ad `#!/usr/local/bin/php −q<?phperror_reporting (E_ALL);/* Autorise l'éxécution infinie du script, en at`

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.

- 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_nonblock
- socket_set_option
- socket_shutdown
- socket_strerror
- socket_write
- socket_writev

6.97.10 socket_bind() : Lie un nom à une socket

`bool` **socket_bind** (`resource socket` , `string address` , *`int port`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.97.11 socket_clear_error() : Efface toutes les erreurs précédemment générées par une socket

`void` **socket_clear_error** (*`resource socket`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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'avénement d'une erreur durant une partie de l'applcation.

Voir aussi `socket_last_error` et `socket_strerror` .

6.97.12 socket_close() : Ferme une socket

`void` **socket_close** (`resource socket`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 `psockopen` ; cette fonction est destinée à traiter les sockets créées par les fonctions `socket_create` ou `socket_accept` .

See also `socket_bind` , `socket_listen` , `socket_create` et `socket_strerror` .

6.97.13 socket_connect() : Crée une connexion sur une socket

`bool` **socket_connect** (`resource socket` , `string address` , *`int port`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` créée 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` .

6.97.14 socket_create_listen() : Ouvre une socket sur un port pour accepter les connexions

`resource` **socket_create_listen** (`int port` , *`int backlog`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` créée 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` .

6.97.15 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`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.16 socket_create() : Crée une socket (point de communication)

`resource` **socket_create** (`int domain` , `int type` , `int protocol`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` 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 .

protocol sélectionne le protocole qui peut être soit SOL_UDP , soit SOL_TCP .

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'aleret (E_WARNING).

Voir aussi socket_accept , socket_bind , socket_connect , socket_listen , socket_last_error , et socket_strerror .

6.97.17 socket_get_option() : Lit les options de socket

mixed **socket_get_option** (resource socket , int level , int optname)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, 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.

6.97.18 socket_getpeername() : Interroge l'autre extrémité de la communication

bool **socket_getpeername** (resource socket , string &addr , *int &port*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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_getpeername` , `socket_last_error` et `socket_strerror` .

6.97.19 socket_getsockname() : Interroge la socket locale

`bool` **socket_getsockname** (`resource socket` , `string &addr` , *`int &port`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` .

6.97.20 socket_iovec_add() : Ajoute un nouveau vecteur dans le tableau

`bool` **socket_iovec_add** (`resource iovec` , `int iov_len`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.21 socket_iovec_alloc() : Construit un 'struct iovec'

`resource` **socket_iovec_alloc** (`int num_vectors` , *`int`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.22 socket_iovec_delete() : Efface un vecteur

`bool` **socket_iovec_delete** (`resource iovec` , `int iov_pos`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.23 socket_iovec_fetch() : Lit les données du vecteur specifié par iovec_id[iovec_position]

`string` **socket_iovec_fetch** (`resource iovec` , `int iovec_position`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.24 socket_iovec_free() : Détruit le vecteur iovec specifié par iovec_id

`bool` **socket_iovec_free** (`resource iovec`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.25 socket_iovec_set() : Modifie les données du vecteur iovec_id[iovec_position]

`bool` **socket_iovec_set** (`resource iovec` , `int iovec_position` , `string new_val`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.26 socket_last_error() : Lit la dernière erreur générée par une socket

`int` **socket_last_error** (*`resource socket`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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
```

Note
socket_last_error n'efface pas le code d'erreur. Utilisez plutôt la fonction socket_clear_error pour cela.

6.97.27 socket_listen() : Attend une connexion (listen) sur une socket

bool **socket_listen** (resource socket , *int backlog*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 focntion 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 .

6.97.28 socket_read() : Lit des données d'une socket

string **socket_read** (resource socket , int length , *int type*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 chaine 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 optinnel 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 .

6.97.29 socket_readv() : Lit des données dans un fd, en utilisant le tableau scatter−gather définit par iovec_id

bool **socket_readv** (resource socket , resource iovec_id)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.30 socket_recv() : Reçoit des données d'une socket connectée

string **socket_recv** (resource socket , int len , int flags)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.31 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*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.32 socket_recvmsg() : Reçoit des messages sur uen 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*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.33 socket_select() : Exécute l'appel système select() un tableau de socket avec une durée d'expiration

int **socket_select** (resource &read , resource &write , resource &except , int tv_sec , *int tv_usec*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 reconnaitront 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 modifiées pour indiquer quelles sockets ont changés 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:

```
<?php  /* Prépare le tableau read (socket surveillées en lecture) */  $read = array($socket1, $socke
```

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 focntion, qui attend une valeur par référence. A la place, utiliser un tableau temporaireou une expression dont le membre de gauche est une variable temporaire : `<?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 opur 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 modifiées. 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 === lorsque vous vérifiez les erreurs. Etant donnée que socket_sele `<?php  if (false === socket_select($r, $w, $e = NULL, 0)) {    echo "socket_select() a échoué. Raiso`

Note
Méfiez vous des implémentations de sockets, qui doivent être manipulées avec délicatesse. Quelques règles de base : • Vous devez toujours essayer d'utiliser socket_select 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 peut 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 socket_select . Après le retour de socket_select , 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` .

6.97.34 socket_send() : Envoie des données à une socket connectée

`int` **socket_send** (`resource socket` , `string buf` , `int len` , `int flags`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.35 socket_sendmsg() : Envoie un message à une socket, qu'elle soit orientée connexion ou pas

`bool` **socket_sendmsg** (`resource socket` , `resource iovec` , `int flags` , `string addr` , *`int port`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.36 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`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.37 socket_set_nonblock() : Sélectionne le mode non bloquant d'un descripteur de fichier

`bool` **socket_set_nonblock** (`resource socket`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.38 socket_set_option() : Modifie les options de socket

`bool` **socket_set_option** (`resource socket` , `int level` , `int optname` , `int`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, 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.

6.97.39 socket_shutdown() : Eteind une socket en lecture et/ou écriture

`bool` **socket_shutdown** (`resource socket` , *`int how`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.97.40 socket_strerror() : Retourne une chaîne décrivant un message d'erreur

`string` **socket_strerror** (`int errno`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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`

```
<?phpif (false == ($socket = @socket_create(AF_INET, SOCK_STREAM, 0))) {    echo "socket_create() a é
```

Voir aussi `socket_accept` , `socket_bind` , `socket_connect` , `socket_listen` et `socket_create` .

6.97.41 socket_write() : Ecrit dans une socket

`int` **socket_write** (`resource socket` , `string buffer` , *`int length`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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 automatiquemetn à la taille du buffer lui−même.

`socket_write` retourne le nombre d'octets qui ont pus êter 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` .

6.97.42 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`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.98 Chaînes de caractères (Strings)

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 du manuel.

Pour des fonctions encore plus puissantes de gestion et manipulation des chaînes, reportez−vous aux expressions régulières POSIX et expressions régulières Perl .

- addcslashes
- addslashes
- bin2hex
- chop
- chr
- chunk_split
- convert_cyr_string
- count_chars
- crc32
- crypt
- echo
- explode
- get_html_translation_table
- get_meta_tags
- hebrev
- hebrevc
- htmlentities
- htmlspecialchars
- implode
- join
- levenshtein
- localeconv
- ltrim
- md5_file
- md5
- metaphone
- nl_langinfo
- nl2br
- number_format
- ord
- parse_str
- print
- printf
- quoted_printable_decode
- quotemeta
- rtrim
- setlocale
- similar_text
- soundex
- sprintf
- sscanf

- str_pad
- str_repeat
- str_replace
- str_rot13
- str_word_count
- strcasecmp
- strchr
- strcmp
- strcoll
- strcspn
- strip_tags
- stripcslashes
- stripslashes
- stristr
- strlen
- strnatcasecmp
- strnatcmp
- strncasecmp
- strncmp
- strpos
- strrchr
- strrev
- strrpos
- strspn
- strstr
- strtok
- strtolower
- strtoupper
- strtr
- substr_count
- substr_replace
- substr
- trim
- ucfirst
- ucwords
- vprintf
- vsprintf
- wordwrap

6.98.2 addslashes() : Ajoute des anti−slashs dans une chaîne

`string` **addslashes** (`string str`)
[lien librairie]

`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
magic_quotes_gpc est activé par défaut.

Voir aussi `stripslashes` , `htmlspecialchars` et `quotemeta` .

6.98.3 bin2hex() : Convertit des données binaires en représentation hexadécimale

`string` **bin2hex** (`string str`)
[lien librairie]

bin2hex retourne la chaîne `str` dont tous les caractères sont représentés par leur équivalent hexadécimal. La chaîne retourneé est une chaîne ASCII. La conversion supporte les caractères binaires, et utilise les bits de poids forts.

Voir aussi `pack` et `unpack` .

6.98.4 chop() : Alias de `rtrim`

`chop` est un alias de `rtrim` .

[lien librairie]

Note
`chop` est différente de son équivalent Perl `chop()` , qui supprime le dernier caractère dans la chaîne.

6.98.5 chr() : Retourne un caractère spécifique

`string` **chr** (`int ascii`)
[lien librairie]

chr retourne une chaîne d'un seul caractère, dont le code ASCII est donné par le paramètre `ascii` .

Exemple avec `chr`

```
<?php
  $str .= chr(27);
/* Ajoute un caractère d'échappement à la fin de la chaîne $str */

/* Ceci est souvent plus pratique, fait réalise la même chose */
  $str = sprintf("The string ends in escape: %c", 27);
?>
```

Vous pouvez trouvez une table des codes ASCII ici : http://www.asciitable.com .

Cette fonction est l'inverse de `ord` . Voir aussi `sprintf` , et le caractère de formattage %c .

6.98.6 chunk_split() : Scinde une chaîne

`string` **chunk_split** (`string body` , *`int chunklen`* , *`string end`*)
[lien librairie]

chunk_split scinde la chaîne body en segments de chunklen octets de longeur. 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 défault, 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 et wordwrap .

6.98.7 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)
[lien librairie]

convert_cyr_string retourne la chaîne str , après l'avoir traduit 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

6.98.8 count_chars() : Retourne des statistiques sur les caractères utilisés dans une chaîne

mixed **count_chars** (string string , *int mode*)
[lien librairie]

count_chars compte le nombre d'occurences 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 utilisé est retournée.
- 4 − une chaîne contenan tous les octets non utilisés est retournée.

6.98.9 crc32() : Calculate la somme de vérification CRC32

`int` **crc32** `( string str )`
[lien librairie]

crc32 génère la somme de vérification cyclique CRC 32, calculée sur 32 bits de long, appliqués à la chaîne `str` . Cette fonction généralement utilisée pour valider l'intégritée de données durant une transmission.

Voir aussi `md5`

6.98.10 crypt() : Chiffrage indéchiffrable (hashing)

`string` **crypt** `( string str , string salt )`
[lien librairie]

`crypt` retourne la chaîne `str` chiffrée avec l'algorithme standard Unix *DES* , ou bien un des algorithme qui est disponible sur la machine. L'argument optionnel `salt` sera utilisée comme base pour le chiffrement. Reportez vous aux pages de manuel Unix pour plus de détails sur votre fonction crypt.

Si l'argument `salt` n'est pas fourni, PHP en générera un lui−même.

Certains systèmes supoprtent plus d'un type de chiffrage. En fait, il arrive que le chiffrage DES standard soit remplacé par un algorithme de chiffrage MD5. Le choix du type de chiffrage est effectué en se basant sur la valeur du `salt` . A l'installation, PHP détermine les possibitilités de la fonction crypt(), et acceptera des `salt` pour d'autres types de chiffrages. Is 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 utiilsez 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 sytèmes 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 PHP

/*
  Il vaut mieux passer le résultat complet de crypt() comme salt nécessaire
  pour le chiffrage du mot de passe, pour éviter les problèmes entre les
  algorithmes utilisés (comme nous le disons ci-dessus, le chiffrage
  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 .

6.98.11 echo() : Affiche une chaîne de caractères

echo `string arg1(`*`string argn...`*`)`
[lien librairie]

Outputs all parameters.

`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 `echo`

```
<?php
  echo "Bonjour le monde";

echo "Cet echo() se
répartit sur plusieurs lignes. Les nouvelles lignes
seront aussi affichées";

echo "Cet echo() se\nrépartit sur plusieurs lignes. Les nouvelles lignes\nseront aussi affichées.";

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"

// Les guillemets simples évitent le remplacement des variables
echo 'foo is $foo'; // foo vaut "$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
supplémentaire!
END;

// comme echo() n'est pas une fonction, le code suivant est invalide
($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 (=).

```
I have <?=$foo?> foo.
```

Note
Cette syntaxe courte n'est possible que si la configuration de short_open_tag a été activée.

Voir aussi `print` , `printf` and `flush` .

6.98.12 explode() : Coupe une chaîne en segments

`array` **explode** (`string separator` , `string string` , *`int limit`*)
[lien librairie]

explode retourne un tableau de chaines. Ce sont les sous-chaînes, extraites de `string` , en utilisant le séparateur `separator` . Si `limit` est fourni, le tableau retourné aurant 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'argumetn `string` .

Voir aussi `preg_split` , `spliti` , `split` et `implode` .

6.98.13 get_html_translation_table() : Returns the translation table used by `htmlspecialchars` and `htmlentities`

`string` **get_html_translation_table** (`int table` , *`int quote_style`*)
[lien librairie]

`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

```
<?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.

```
<?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` .

6.98.14 get_meta_tags() : Extrait tous les méta tags d'un fichier

`array` **get_meta_tags** (`string filename` , *`int use_include_path`*)
[lien librairie]

`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="tags" content="php3 documentation">`
`</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 replacés 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.

6.98.15 hebrev() : Convertit un texte logique hébreux en texte visuel

`string` **hebrev** (`string hebrew_text` , *`int max_chars_per_line`*)
[lien librairie]

hebrev convertit le texte logique hébreux `hebrew_text` en texte visuel. 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 `hebrevc` .

6.98.16 hebrevc() : Convertit un texte logique hébreux en texte visuel, avec retours à la ligne

`string` **hebrevc** (`string hebrew_text` , *`int max_chars_per_line`*)
[lien librairie]

`hebrevc` est similaire à `hebrev` à la différence qu'elle convertit les nouvelles lignes (\n) en "
\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 `hebrev` .

6.98.17 htmlentities() : Convertit tous les caractères éligibles en entités HTML

`string` **htmlentities** (`string string` , *`int quote_style`* , *`string charset`*)
[lien librairie]

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 traité 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.

Actuellement, le jeu de caractères ISO−8859−1 est utilisé par défaut. Le support pour le second argument optionnel a été ajouté en PHP 3.0.17 et PHP 4.0.3.

Comme `htmlspecialchars` , cette fonction prend un troisième argument optionnel qui définit le jeux de caractères utilisé durant la conversion. Le support de cet argumetn a été ajouté en PHP 4.1.0.

Il n'y a pas de fonction inverse de cette fonction. Toutes fois, vous pouvez en créer une facilement. Voici comment faire :

Inverse de `htmlentities`

```
<?php
  function unhtmlentities ($string){
    $trans_tbl = get_html_translation_table (HTML_ENTITIES);
    $trans_tbl = array_flip ($trans_tbl);
    return strtr ($string, $trans_tbl);
  }
?>
```

Voir aussi `htmlspecialchars` et `nl2br` .

6.98.18 htmlspecialchars() : Convertit les caractères spéciaux en entités HTML

`string` **htmlspecialchars** (`string string` , *`int quote_style`* , *`string charset`*)
[lien librairie]

Certains caractères ont des significations spéciales en HTML, et doivent être replacés 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` instead.

`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 traité 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 '&'
- '"' (guillemets doubles) devient '"' lorsque ENT_NOQUOTES n'est pas utilisé.
- ''' (single quote) devient ''' uniquement lorsque ENT_QUOTES est utilisé.
- '<' (supérieur à) devient '<'
- '>' (inférieur à) devient '>'

Exemple avec `htmlspecialchars`

```
<?php
  $new = htmlspecialchars("<a href='test'>Test</a>", ENT_QUOTES);
?>
```

Notez que cette fonction ne fait aucune autre remplacement que ceux qui sont listés ci−dessus. Pour faire un remplacement total, voyez plutôt `htmlentities` . Le support pour l 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.

Voir aussi `htmlentities` et `nl2br` .

6.98.19 implode() : Rassemble les éléments d'un tableau en une chaîne

`string` **implode** (`string glue` , `array pieces`)
[lien librairie]

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` , toutes fois, il est préférable d'utiliser l'ordre des arguments tels que documenté.

Voir aussi explode et split .

6.98.20 join() : Rassemble les éléments d'un tableau en une chaîne

string **join** (string glue , array pieces)
[lien librairie]

join est un alias de implode .

Voir aussi explode , implode et split .

6.98.21 levenshtein() : Calcule la distance Levenshtein entre deux chaînes

int **levenshtein** (string str1 , string str2)
[lien librairie]
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 distance 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 paramèters, et calculer simplement le nombre d'insertions, de remplacements et d'effacements nécessaires pour tranformer 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 appelera 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 prendr 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 faîtes en terme d'exploitation du CPU et des buffers.

Voir aussi `soundex` , `similar_text` et `metaphone` .

6.98.22 localeconv() : Lit la configuration locale

`array` **localeconv** (`void`)
[lien librairie]

`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
decimal_point	Séparateur décimal
thousands_sep	Séparateur de milliers
grouping	Tableau contenant les regroupements numériques
int_curr_symbol	Symbole monétaire international (i.e. EUR)
currency_symbol	Symbole monétaire local (i.e. €)
mon_decimal_point	Séparateur décimal monétaire
mon_thousands_sep	Séparateur de milliers monétaires
mon_grouping	Tableau contenant les regroupements numériques monétaires
positive_sign	Signe des valeurs positives
negative_sign	Signe des valeurs négatives
int_frac_digits	Nombre international de décimales
frac_digits	Nombre local de décimales
p_cs_precedes	TRUE si le currency_symbol précède une valeur positive et FALSE s'il lui succède.
p_sep_by_space	TRUE si un espace sépare currency_symbol d'une valeur positive, et FALSE sinon.
n_cs_precedes	TRUE si currency_symbol précède une valeur négative, et FALSE s'il lui succède.
n_sep_by_space	TRUE si un espace sépare currency_symbol d'une valeur négative, et FALSE sinon.
p_sign_posn	• 0 • Des parenthèses entourent la valeur et le symbole monétaire • 1 • Le signe précède la valeur et le symbole monétaire • 2 • Le signe suit la valeur et le symbole monétaire

	• 3 • Le signe précède immédiatement la valeur et le symbole monétaire • 4 • Le signe suit immédiatement la valeur et le symbole monétaire
n_sign_posn	• 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 champs 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.

`localeconv` example

```
<?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` .

6.98.23 ltrim() : Supprime les caractères invisibles de début de chaîne

`string` **ltrim** (`string str` , *`string charlist`*)
[lien librairie]

`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
<?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` .

6.98.24 md5_file() : Calcule le md5 d'un fichier

`string` **md5_file** (`string filename`)
[lien librairie]

`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.

`md5_file` a la même utilité que la commande en ligne md5sum.

Voir aussi `md5` et `crc32`

6.98.25 md5() : Calcule le md5 d'une chaîne

`string` **md5** (`string str`)
[lien librairie]

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.

Voir aussi `crc32` et `md5_file`

6.98.26 metaphone() : Calcule la clé métaphone

`string` **metaphone** (`string str`)
[lien librairie]

`metaphone` calcule la clé metaphone de la chaîne de caractères `str` .

`metaphone` est similaire à la fonction `soundex` : elle créée une clé similaire pour des mots dont la prononciation est proche. C'est une fonction qui est plus précise que `soundex` car elle prend en compte la prononciation anglaise. La clé metaphone générée est de taille variable.

`metaphone` fonctionne en anglais, et n'utilise pas les spécificités de la langue française. Les résultats sur des termes français peuvent être 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].

6.98.27 nl_langinfo() : Rassemble des informations sur la langue et la configuration locale.

`string` **nl_langinfo** (`int item`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.98.28 nl2br() : Insère des retours à la ligne HTML à chaque nouvelle ligne

`string` **nl2br** (`string string`)
[lien librairie]

nl2br retourne `string` après avoir inséré '<br />' 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 et wordwrap .

6.98.29 number_format() : Formate un nombre pour l'affichage.

string **number_format** (float number , *int decimals* , *string dec_point* , *string thousands_sep*)
[lien librairie]

number_format retourne une chaîne représentant number formaté. number_format accepte un, deux ou 4 paramètres (mais pas trois).

Si le seul 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 .

6.98.30 ord() : Retourne le code ASCII d'un caractère

int **ord** (string string)
[lien librairie]

ord retourne le code ASCII du premier caractère de string . Cette fonction est le contraire de chr .

Exemple avec ord

```php
<?php
  if (ord($str) == 10) {
    echo "Le premier caractère de \$str est une nouvelle ligne\n";
  }
?>
```

Vous pouvez trouvez une table des codes ASCII ici : http://www.asciitable.com .

Voir aussi `chr` .

6.98.31 parse_str() : Analyse une requête HTTP

`void` **parse_str** (`string str` , *`array arr`*)
[lien librairie]

`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.

Exemple avec `parse_str`

```php
<?php
  $str = "premier=valeur&second[]=qui+fonctionne&second[]=autre";
  parse_str($str);
  echo $premier;     /* affiche "valeur" */
  echo $second[0]; /* affiche "qui fonctionne" */
  echo $second[1]; /* affiche  "autre" */
?>
```

Voir aussi `set_magic_quotes_runtime` et `urldecode` .

6.98.32 print() : Affiche une chaîne de caractères

print `string arg` ()
[lien librairie]

print affiche la chaine de caractères `arg` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`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. En fait, si vous voulez passer plusieurs paramètres à afficher, vous ne devez pas utiliser de parenthèses. Il n'est pas possible d'utiliser `print` dans un contexte de fonction variable .

Exemples avec `print`

```
<?php
  print("Bonjour le monde");

  print "print() fonctionne aussi avec des parenthèses.";

  print "Ce print() se
répartit sur plusieurs lignes. Les nouvelles lignes
seront aussi affichées";

  print "Ce print() se\nrépartit sur plusieurs lignes. Les nouvelles lignes\nseront aussi affichées.

  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"

// Les guillemets simples évitent le remplacement des variables
print 'foo is $foo'; // foo vaut "$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
supplémentaire!
END;
?>
```

Voir aussi `echo` , `printf` et `flush` .

6.98.33 printf() : Affiche une chaîne de caractères formattée

`void` **printf** (`string format` , *`mixed args`*)
[lien librairie]

`printf` affiche les arguments `args` formaté sous forme de chaîne de caractères grâce à `args` .
Les options de formatage sont documentés avec la fonction `sprintf` .

Voir aussi `print` , `sprintf` , `sscanf` , `fscanf` et `flush` .

6.98.34 quoted_printable_decode() : Convertit une chaîne quoted-printable en chaîne 8 bits

`string` **quoted_printable_decode** (`string str`)
[lien librairie]

quoted_printable_decode retourne la chaîne de caractères `str` , après l'avoir converti 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.

6.98.35 quotemeta() : Echappe les méta−caractères

`string` **quotemeta** (`string str`)
[lien librairie]

quotemeta retourne la chaîne `str` après avoir introduit un anti−slash (`\`) devant tous les caractères suivants : `. \\ + * ? [ ^ ] ( $ )`

Voir aussi `addslashes` , `htmlentities` , `htmlspecialchars` , `nl2br` et `stripslashes` .

6.98.36 rtrim() : Supprime les espaces de fin de chaîne

`string` **rtrim** (`string str` , *`string charlist`*)
[lien librairie]

`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.
- " " (`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
<?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` .

6.98.37 setlocale() : Modifie les informations de localisation

`string` **setlocale** (`mixed category` , `string locale`)
[lien librairie]

`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 spéarateur décimal. Voir `localeconv`
- `LC_TIME` pour le format de date et d'heur 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` vaut zéro ou `"0"` , la configuration de localisation ne sera pas modifiée, et seule la valeur courante sera retournée.

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));
?>
```

6.98.38 similar_text() : Calcule la similarité de deux chaînes

`int` **similar_text** (`string first` , `string second` , *`float percent`*)
[lien librairie]

`similar_text` calcule la similarité entre els deux chaînes `first` et `second` , selon la méthode de Oliver [1993]. Notez que cette implémentation n'utilise par 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.

6.98.39 soundex() : Calcule la clé soundex

`string` **soundex** (`string str`)
[lien librairie]

soundex calcule la clé soundex de la chaîne `str` .

La clé soundex possède la propriété qui afit que deux mots prononcés similairements auront la même clé soundex. Cette fonction est donc utilisé 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` .

6.98.40 sprintf() : Retourne une chaîne formattée

`string` **sprintf** (`string format` , *`mixed args`*)
[lien librairie]

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 **des spécifications de conversion** , qui exploite chacune un des arguments passé après la chaîne de format. Ces formats s'appliquent à `sprintf` et `printf` .

Chaque spécification de conversion est consistitué 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 auter fonction pratique pour formatter 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 litéral. 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 support 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 avnos 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 formattage 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` .

6.98.41 sscanf() : Analyse une chaîne à l'aide d'un format

`mixed` **sscanf** (`string str` , `string format` , *`string var1`*)
[lien librairie]

`sscanf` est l'inverse de la fonction `printf` . `sscanf` lit des données dans la chaîne `str` , et l'interpéte 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`

```
ç
<?php
// Lecture d'u 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 optionels 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
<?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` .

6.98.42 str_pad() : Complète une chaîne jusqu'à une taille donnée

`string` **str_pad** (`string input` , `int pad_length` , *`string pad_string`* , *`int pad_type`*)
[lien librairie]

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 complété 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 spécifiée, il prend la avleur par défaut de `STR_PAD_RIGHT` .

Si la valuer de `pad_length` est négative, ou inférieure à la taille courante de la chaîne `input` , `input` est retournée inchangée.

Exemple avec `str_pad`

```php
<?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___"
?>
```

6.98.43 str_repeat() : Répéte une chaîne

`string` **str_repeat** (`string input` , `int multiplier`)
[lien librairie]

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
<?php
  echo str_repeat("-=", 10);
```

```
?>
```

Ceci affichera "-=-=-=-=-=-=-=-=-=-=".

Voir aussi for , str_pad et substr_count .

6.98.44 str_replace() : Remplace toutes les occurrences dans une chaînes

mixed **str_replace** (mixed search , mixed replace , mixed subject)
[lien librairie]

str_replace retourne une chaîne ou un tablaeu, dont toutes les occurences de search dans subject ont été remplacée 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 paramèters de str_replace peuvent être des tableaux.

Si subject est un tablaeu, alors le remplacement se fera sur chaque élément du tableau, 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
  $bodytag = str_replace("%body%", "black", "<body text=%body%>");
?>
```

str_replace est compatible avec les chaînes binaires.

Note

str_replace a été ajoutée en PHP 3.0.6, mais étaient buggée jusqu'en PHP 3.0.8.

Voir aussi ereg_replace , preg_replace et strtr .

6.98.45 str_rot13() : Effectue une transformation rot13

string **str_rot13** (string str)
[lien librairie]

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 de décodage est fait par la même fonction : passer le résultat de `str_rot13` à nouveau comme argument, retournera la chaîne originale.

6.98.46 str_word_count() : Compte le nombre de mots utilisés dans une chaîne

`mixed` **str_word_count** (`string string` , *`int format`*)
[lien librairie]

`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 "−".

6.98.47 strcasecmp() : Comparaison de chaînes binaires

`int` **strcasecmp** (`string str1` , `string str2`)
[lien librairie]

`strcasecmp` retourne < 0 si `str1` est inférieure à `str2` ; > 0 si `str1` est plus grand que `str2` , et 0 si les deux chaînes sont égales.

Notez que cette comparaison est sensible à la casse.

Exemple avec `strcasecmp`

```
<?php
  $var1 = "Hello";
  $var2 = "hello";
  if (!strcasecmp($var1, $var2)) {
    echo "$var1 est égal à $var2 (comparaison insensible à la casse)";
  }
?>
```

Vior aussi `ereg` , `strcmp` , `substr` , `stristr` , `strncasecmp` et `strstr` .

6.98.48 strchr() : Trouve la première occurence d'un caractère dans une chaîne

`string` **strchr** (`string haystack` , `string needle`)
[lien librairie]

strchr est un alias de strstr .

6.98.49 strcmp() : Comparaison binaire de chaînes

int **strcmp** (string str1 , string str2)
[lien librairie]

strcmp retourne < 0 si str1 est inférieur à str2 ; > 0 si str1 est supérieur 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 .

6.98.50 strcoll() : Comparaison de chaînes localisée

int **strcoll** (string str1 , string str2)
[lien librairie]

strcoll retourne < 0 si str1 est inférieur à str2 ; > 0 si str1 est supérieur 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 .

6.98.51 strcspn() : Trouve un segment de chaîne ne contenant pas certains caractères

int **strcspn** (string str1 , string str2)
[lien librairie]

strcspn retourne la taille du premier segment de str1 qui ne contient **aucun** des caractères de str2 .

Voir aussi strspn .

6.98.52 strip_tags() : Supprime les balises HTML et PHP d'une chaîne

string **strip_tags** (string str , *string allowable_tags*)
[lien librairie]

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 erronnées. 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 `strip_tags`

`<?php`
`  $string = strip_tags($string, '<a><b><i><u>');`
`?>`

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 pernicieux peuvent utiliser.

6.98.53 stripcslashes() : Supprime les anti−slash d'une chaîne C

`string` **stripcslashes** (`string str`)
[lien librairie]

`stripcslashes` retourne la chaîne `str` après avoir supprimé tous les anti−slash. `stripcslashes` respecte les séquences spéciales du C, telles que `\n` , `\r` ..., les nombres octaux et héxadecimaux.

Voir aussi `addcslashes` .

6.98.54 stripslashes() : Supprimer les anti−slash d'une chaîne

`string` **stripslashes** (`string str`)
[lien librairie]

`stripslashes` retourne une chaîne dont les anti−slash on été supprimé. `\'` devient `'` , etc... Les doubles anti−slash sont réduits en un seul anti−slash. `stripslashes` est la fonction inverse de `addslashes` .

Voir aussi `addslashes` .

6.98.55 stristr() : Trouve la première occurence dans une chaîne (insensible à la casse)

`string` **stristr** (`string haystack` , `string needle`)
[lien librairie]

`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 utilisé comme code ascii du caractère correspondant.

Exemple avec stristr

```
<?php
  $email = 'USER@EXEMPLE.com';
  $domaine = stristr($email, 'e');
  print $domaine;
// Affiche ER@EXEMPLE.com
?>
```

Voir aussi strchr , strrchr , substr et ereg .

6.98.56 strlen() : Calcule la taille d'une chaîne

int **strlen** (string str)
[lien librairie]

strlen retourne la taille de la chaîne string .

6.98.57 strnatcasecmp() : Comparaison de chaînes avec l'algorithme d'"ordre naturel" (insensible à la casse)

int **strnatcasecmp** (string str1 , string str2)
[lien librairie]

strnatcasecmp implémente l'algorithme de comparaison qui ordonne les chaînes tel qu'un homme le ferai. 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 inférieur à str2 ; > 0 si str1 est supérieur à str2 , et 0 si les deux chaînes sont égales.

Voir aussi ereg , strcasecmp , substr , stristr , strcmp , strncmp , strncasecmp , strnatcmp et strstr .

6.98.58 strnatcmp() : Comparaison de chaînes avec l'algorithme d'"ordre naturel"

int **strnatcmp** (string str1 , string str2)
[lien librairie]

strnatcmp implémente l'algorithme de comparaison qui ordonne les chaînes tel qu'un homme le ferai. Cette fonction est similaire à la fonction strnatcmp , 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 :

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

The code above will generate the following output:

```
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 inférieur à `str2` ; > 0 si `str1` est supérieur 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` .

6.98.59 strncasecmp() : Compare en binaire des chaînes de caractères

`int` **strncasecmp** (`string str1` , `string str2` , `int len`)
[lien librairie]

`strncasecmp` est similaire à `strcasecmp` , à la différence près qu'elle permet de limiter le nombre de caractères utilisés pour comparer `str1` et `str1` , 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 petit que str2 ; > 0 si str1 est plus grand que str2 , et 0 si elles sont égales.

Voir aussi ereg , strcasecmp , strcmp , substr , stristr et strstr .

6.98.60 strncmp() : Comparaison binaire des n premiers caractères

int **strncmp** (string str1 , string str2 , int len)
[lien librairie]

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 inférieur à str2 ; > 0 si str1 est supérieur 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 .

6.98.61 strpos() : Trouve la position d'un caractère dans une chaîne

int **strpos** (string haystack , string needle , *int offset*)
[lien librairie]

strpos retourne la position numérique de la première occurence 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 .

Note
Il est facile de confondre le résultat "caractère trouvé à la position 0" et "caractère non trouvé". Voici comment faire la différence :

```php
<?php
// en PHP 4.0b3 et plus récent :
  $pos = strpos($machaine, "b");
  if ($pos === false) { // note : trois égals
    // pas trouvé
  }

// en versions plus ancienne 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 utilisé comme caractère de code ASCII correspondant.

Le paramètre optionnel offset vous permet de spécifier à partir de quel caractère dans haystack vous souhaitez commencer la recherche. La position retournée sera toujours relative au début de la chaîne haystack .

Voir aussi strrpos , strrchr , substr , stristr et strstr .

6.98.62 strrchr() : Retourne la fin de la chaîne

string **strrchr** (string haystack , string needle)
[lien librairie]

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 utilisé comme caractère de code ASCII correspondant.

Exemple avec strrchr

```
<?php
// Lit le dernier dossier dans le chemin $PATH
  $dir = substr(strrchr($PATH, ":"), 1);

// Lit tout, après la dernière nouvelle ligne
  $text = "Ligne 1\nLigne 2\nLigne 3";
  $last = substr(strrchr($text, 10), 1 );
?>
```

Voir aussi strchr , substr , stristr et strstr .

6.98.63 strrev() : Inverse une chaîne

string **strrev** (string string)
[lien librairie]

strrev retourne la chaîne string , après avoir changé l'ordre des caractères.

Inverser une chaîne avec strrev

```php
<?php
echo strrev("Bonjour le monde!");
// affiche "ednom el ruojnoB"
?>
```

6.98.64 strrpos() : Trouve la position de la dernière occurence d'un caractère dans une chaîne

`int` **strrpos** (`string haystack` , `char needle`)
[lien librairie]

`strrpos` retourne la position numérique de la dernière occurence 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` `FALSE` .

Note

Il est facile de confondre le résultat "caractère trouvé à la position 0" et "caractère non trouvé". Voici comment faire la différence :

```php
<?php
// en PHP 4.0b3 et plus récent :
  $pos = strrpos($machaine, "b");
  if ($pos === false) { // note : trois égals
    // pas trouvé
  }

// en versions plus ancienne que 4.0b3 :
  $pos = strrpos($machaine, "b");
  if (!is_integer($pos)) {
    // pas trouvé
  }
?>
```

Si `needle` n'est pas une chaîne, elle est convertie en entier, et utilisé comme caractère de code ASCII correspondant.

Voir aussi `strpos` , `strrchr` , `substr` , `stristr` et `strstr` .

6.98.65 strspn() : Trouve le premier segment de chaîne

`int` **strspn** (`string str1` , `string str2`)
[lien librairie]

`strspn` retourne la taille du premier segment de chaîne `str1` qui est entièrement constitué de caractères contenus dans `str2` .

```
<?php
  echo strspn("421 est la reponse, mais quelle est la question ...", "1234567890");
?>
```

va afficher 3, car la chaîne "421" est le plus long segment de "421 est la reponse, mais quelle est la question ...", dont tous les caractères sont contenus dans "1234567890".

Voir aussi `strcspn` .

6.98.66 strstr() : Trouve la première occurence dans une chaîne

`string` **strstr** (`string haystack` , `string needle`)
[lien librairie]

`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 utilisé 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` , `strchr` , `stristr` , `strpos` , `strrchr` et `substr` .

6.98.67 strtok() : Coupe une chaîne en segments

`string` **strtok** (`string arg1` , `string arg2`)
[lien librairie]

`strtok` coupe la chaîne `arg1` en segments, chaque segement 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
<?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 seule le premier appel à `strtok` nécessite les deux arguments. Tous les appels ultérieurs à `strtok` ne nécessite 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 plsu, soyez prudent avec les séparateurs qui sont égaux à "0". Ces valeurs s'évaluent comme FALSE dans les expression conditionnelles.

Voir aussi `split` et `explode` .

6.98.68 strtolower() : Make a string lowercase

`string` **strtolower** (`string str`)
[lien librairie]

`strtolower` retourne `string` , après avoir converti tous les caractères alphabétique en minuscules.

Notez que la notion d'"alphabétique" est déterminé 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.

`strtolower` example

```
<?php
  $str = "Marie A un Petit Agneau, et l'aime TRes fORt.";
  $str = strtolower($str);
  print $str;
// Affiche : marie a un petit agneau, et l'aime tres fort.
?>
```

Voir aussi `strtoupper` , `ucfirst` et `ucwords` .

6.98.69 strtoupper() : Make a string uppercase

`string` **strtoupper** (`string string`)
[lien librairie]

`strtoupper` retourne `string` , après avoir converti tous les caractères alphabétique en majuscules.

Notez que la notion d'"alphabétique" est déterminé 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` example

```
<?php
  $str = "Marie A un Petit Agneau, et l'aime TRes fORt.";
  $str = strtolower($str);
  print $str;
// Affiche : MARIE A UN PETIT AGNEAU, ET L'AIME TRES FORT.
?>
```

Voir aussi `strtolower` , `ucfirst` et `ucwords` .

6.98.70 strtr() : Remplace des caractères

`string` **strtr** (`string str` , `string from` , `string to`)
[lien librairie]
`string` **strtr** (`string str` , `array replace_pairs`)

`strtr` retourne la chaîne `str` , après avoir remplacé chaque caractères du paramètre `from` par son équivalent dans le paramètre `to` .

Si `from` et `to` sont de taille différentes, les caractères en trop dans l'un ou l'autre seront ignorés.

Exemple avec `strtr`

```
<?php
  $addr = strtr($addr, "äâö", "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 string −> string pairs, 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.

```
<?php
  $trans = array("Bonjour" => "salut", "salut" => "bonjour");
  echo strtr("Bonjour à tous, j'ai dit bonjour", $trans) . "\n";
?>
```

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` .

6.98.71 substr_count() : Compte de le nombre d'occurences de segments dans une chaîne

`int` **substr_count** (`string haystack` , `string needle`)
[lien librairie]

`substr_count` retourne le nombre d'occurence de `needle` dans la chaîne `haystack` .

Exemple avec `substr_count`

```
<?php
  print substr_count("This is a test", "is");
  // affiche 2
?>
```

6.98.72 substr_replace() : Remplace un segment dans une chaîne

`string` **substr_replace** (`string string` , `string replacement` , `int start` , *`int length`*)
[lien librairie]

`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 le nombre 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
<?php
$var = 'ABCDEFGH:/MNRPQR/';
echo "Original : $var<hr>\n";

// Remplace toute la chaÔne $var par  'bob'.
echo substr_replace($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` .

6.98.73 substr() : Retourne un segment de chaîne

`string` **substr** (`string string` , `int start` , *`int length`*)
[lien librairie]

`substr` retourne le segment de `string` définit 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 vis 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 retourné 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 seront 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 `strrchr` et `ereg` .

6.98.74 trim() : Supprime les espaces en début et fin de chaîne

`string` **trim** (`string str` , *`string charlist`*)
[lien librairie]

`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.
- " " (`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.

Exemle avec `trim`

`<?php`
 	
`$text = "  Voici quelques mots :) ...  ";`
`$trimmed = trim($text);`
`// $trimmed = "Voici quelques mots :) ..."`
`$trimmed = trim($text,"  .");`
`// $trimmed = "Voici quelques mots :)"`
`$clean = trim($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 `ltrim` et `rtrim` .

6.98.75 ucfirst() : Met le premier caractère en majuscule

`string` **ucfirst** (`string str`)
[lien librairie]

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 déterminé par la configuration de localisation. Cela signifie que pour la configuration par défaut "C", les caractères tels que les voyelles accentuées (comme é, è ou à) ne seront pas converties.

Exemple avec `ucfirst`

```
$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` .

6.98.76 ucwords() : Met en majuscule la première lettre de tous les mots

`string` **ucwords** `( string str )`
[lien librairie]

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 suiv immédiatement un caractère blanc (espace, tabulation, nouvelle ligne, retour chariot, tabulation verticale).

Voir aussi `strtoupper` , `strtolower` et `ucfirst` .

6.98.77 vprintf() : Affiche une chaîne formatée

`void` **vprintf** `( string format , array args )`
[lien librairie]

`vprintf` affiche le tableau `args` , sous forme de chaîne formaté 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`

6.98.78 vsprintf() : Retourne une chaîne formatée

`string` **vsprintf** (`string format` , `array args`)
[lien librairie]

`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`

6.98.79 wordwrap() : Effectue la césure d'une chaîne

`string` **wordwrap** (`string str` , *`int width`* , *`string break`* , *`int cut`*)
[lien librairie]

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 morceau.

Note
Le paramètre optionnel `cut` a été ajouté en PHP 4.0.3.

Exemple avec `wordwrap`

```
<?php
  $text = "The quick brown fox jumped over the lazy dog.";
  $newtext = wordwrap( $text, 20 );

  echo "$newtext\n";
?>
```

Cet exemple va afficher :

```
The quick brown fox
jumped over the
lazy dog.
```

Exemple avec `wordwrap`

```
<?php
```

```
  $text = "Un mot tres tres looooooooooooooooong.";
  $newtext = wordwrap( $text, 8, "\n", 1);

  echo "$newtext\n";
?>
```

Cet exemple va afficher :

```
Un mot
tres
tres
looooooo
oooooooo
ooong.
```

Voir aussi `nl2br` .

6.99 Sybase

6.99.1 Introduction

6.99.2 Pré−requis

6.99.3 Installation

6.99.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Sybase

Nom	Par défaut	Modifiable
sybase.allow_persistent	"On"	PHP_INI_SYSTEM
sybase.max_persistent	"−1"	PHP_INI_SYSTEM
sybase.max_links	"−1"	PHP_INI_SYSTEM
sybase.interface_file	"/usr/sybase/interfaces"	PHP_INI_SYSTEM
sybase.min_error_severity	"10"	PHP_INI_ALL
sybase.min_message_severity	"10"	PHP_INI_ALL
sybase.compatability_mode	"Off"	PHP_INI_SYSTEM
magic_quotes_sybase	"Off"	PHP_INI_ALL

Voici de détails sur les 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 d'inactiver 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.

Nom	Par défaut	Modifiable
sybct.allow_persistent	"On"	PHP_INI_SYSTEM
sybct.max_persistent	"−1"	PHP_INI_SYSTEM
sybct.max_links	"−1"	PHP_INI_SYSTEM
sybct.min_server_severity	"10"	PHP_INI_ALL
sybct.min_client_severity	"10"	PHP_INI_ALL
sybct.hostname	NULL	PHP_INI_ALL

Voici de détails sur les 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.

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.99.5 Types de ressources

6.99.6 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_fetch_array
- 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

6.99.8 sybase_close() : Ferme une connexion Sybase.

`boolean` **sybase_close** (`resource link_identifier`)
[lien librairie]

`sybase_close` retourne TRUE en cas de succès et FALSE en cas d'erreur.

`sybase_close` termine la connexion avec le serveur Sybase associé à l'identifiant de connexion `link_identifier` .

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` .

6.99.9 sybase_connect() : Ouvre une connexion à un serveur Sybase.

`resource` **sybase_connect** (`string servername` , `string username` , `string password`)
[lien librairie]

`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` .

Voir aussi `sybase_pconnect` et `sybase_close` .

6.99.10 sybase_data_seek() : Déplace le pointeur interne de lignes.

`boolean` **sybase_data_seek** (`resource result_identifier` , `int row_number`)
[lien librairie]

`sybase_data_seek` retourne TRUE en cas de succès, et FALSE en cas d'échec.

`sybase_data_seek` déplace le pointeur interne de ligne du résutalt Sybase associé à `result_identifier` jusqu'à la ligne `row_number` . Le prochain appel à `sybase_fetch_row` sans préciser la ligne, retournera la ligne `row_number` .

Voir aussi `sybase_data_seek` .

6.99.11 sybase_fetch_array() : Retourne une ligne sous la forme d'un tableau.

`array` **sybase_fetch_array** (`resource result`)
[lien librairie]

`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 fourni un confort d'utilisation notable.

Pour plus de détails : `sybase_fetch_row` .

6.99.12 sybase_fetch_field() : Lit les informations d'un champs.

`object` **sybase_fetch_field** (`resource result` , *`int field_offset`*)
[lien librairie]

`sybase_fetch_field` retourne un objet contenant les informations du champs.

`sybase_fetch_field` sert à obtenir des informations à propos des champs dans le résultat `result` . Si l'offset du champs n'est pas précisé, le champs 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` .

6.99.13 sybase_fetch_object() : Retourne une ligne sous la forme d'un objet.

`int` **sybase_fetch_object** (`resource result`)
[lien librairie]

`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).

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` .

6.99.14 sybase_fetch_row() : Retourne une ligne sous la forme d'un tableau énuméré.

`array` **sybase_fetch_row** (`resource result`)
[lien librairie]

`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.

Voir aussi `sybase_fetch_array` , `sybase_fetch_object` , `sybase_data_seek` et `sybase_result` .

6.99.15 sybase_field_seek() : Modifie l'index d'un champs.

`int` **sybase_field_seek** (`resource result` , `int field_offset`)
[lien librairie]

sybase_field_seek modifie l'index d'un champs. Le prochain appel à la fonction sybase_fetch_field sans préciser l'index du champs retournera ce champs.

Voir aussi sybase_fetch_field .

6.99.16 sybase_free_result() : Libère un résultat de la mémoire.

boolean **sybase_free_result** (int result)
[lien librairie]

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.

6.99.17 sybase_get_last_message() : Retourne le dernier message du serveur

string **sybase_get_last_message** (void)
[lien librairie]

sybase_get_last_message retourne le dernier message rapporté par le serveur.

6.99.18 sybase_min_client_severity() : Fixe la sévérité minimale du client

void **sybase_min_client_severity** (int severity)
[lien librairie]

sybase_min_client_severity fixe la sévérité minimale du client.

Note
sybase_min_client_severity est disponible avec l'interface CT vers Sybase, mais pas avec la librairie DB.

Voir aussi sybase_min_server_severity .

6.99.19 sybase_min_error_severity() : Fixe la sévérité minimale du client pour les erreurs

void **sybase_min_error_severity** (int severity)
[lien librairie]

sybase_min_error_severity fixe la sévérité minimale du client pour les erreurs.

Voir aussi sybase_min_message_severity .

6.99.20 sybase_min_message_severity() : Fixe la sévérité minimale du client pour les messages

`void` **sybase_min_message_severity** (`int severity`)
[lien librairie]

`sybase_min_message_severity` fixe la sévérité minimale du client pour les messages.

Voir aussi `sybase_min_error_severity` .

6.99.21 sybase_min_server_severity() : Fixe la sévérité minimale du client pour le serveur

`void` **sybase_min_server_severity** (`int severity`)
[lien librairie]

`sybase_min_server_severity` fixe la sévérité minimale du client pour le serveur.

Note
`sybase_min_server_severity` est disponible avec l'interface CT vers Sybase, mais pas avec la librairie DB.

Voir aussi `sybase_min_client_severity` .

6.99.22 sybase_num_fields() : Retourne le nombre de champs dans un résultat.

`int` **sybase_num_fields** (`resource result`)
[lien librairie]

`sybase_num_fields` retourne le nombre de champs du résultat `result` .

Voir aussi `sybase_query` , `sybase_fetch_field` et `sybase_num_rows` .

6.99.23 sybase_num_rows() : Retourne le nombre de lignes dans un résultat.

`int` **sybase_num_rows** (`resource result`)
[lien librairie]

`sybase_num_rows` retourne le nombre de lignes du résultat `result` .

Voir aussi `sybase_query` et `sybase_fetch_row` .

6.99.24 sybase_pconnect() : Ouvre une connexion persistante à un serveur Sybase.

resource **sybase_pconnect** (string servername , string username , string password)
[lien librairie]

sybase_pconnect retourne un identifiant de connexion positif en cas de succès, et FALSE en cas d'erreur.

sybase_connect se comporte comme sybase_connect avec deux différence 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 crée par sybase_pconnect .

Ce type de liens est dit 'persistant'.

6.99.25 sybase_query() : Envoie une requête à une base Sybase.

int **sybase_query** (string query , int link_identifier)
[lien librairie]

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 .

6.99.26 sybase_result() : Lit une valeur dans un résultat.

string **sybase_result** (resource result , int i , mixed field)
[lien librairie]

sybase_result retourne le contenu d'une cellule. L'argument field peut être l'index du champs, ou bien le nom du champs, ou encore, le nom de la table " point " le nom du champs. 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 .

6.99.27 sybase_select_db() : Sélectionne une base de données Sybase.

`boolean` **sybase_select_db** (`string database_name` , `resource link_identifier`)
[lien librairie]

`sybase_select_db` retourne `TRUE` en cas de succès, et `FALSE` en cas d'erreur.

`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` .

Tous les prochains appels à `sybase_query` seront faits sur la bas de données courante et active.

Voir aussi `sybase_connect` , `sybase_pconnect` et `sybase_query` .

6.100 Analyseur de code PHP

6.100.1 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 utiliser ce module à vos risques et périls.

Reportez vous à l'annexe concernant les tokens .

6.100.2 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
tokenizer.global_value	"42"	PHP_INI_ALL
tokenizer.global_string	"foobar"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.100.3 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.

`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)

- Introduction
- Configuration à l'exécution
- Constantes Prédefinies
- token_get_all
- token_name

6.100.5 token_name() : Lit le nom d'un élément de code source

`string` **token_name** `( int type )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.101 URL

6.101.1 base64_decode() : Décode une chaîne en MIME base64

`string` **base64_decode** (`string encoded_data`)
[lien librairie]

`base64_decode` décode `encoded_data` et retourne les données décodées. Les informations initiales peuvent être binaires.

Voir aussi `base64_encode` et la RFC2045 section 6.8.

- base64_decode
- base64_encode
- parse_url
- rawurldecode
- rawurlencode
- urldecode
- urlencode

6.101.3 parse_url() : Analyse une URL et retourne ses composants.

`array` **parse_url** (`string url`)
[lien librairie]

`parse_url` retourne un tableau associatif contenant les composants de l'URL. Les composants recherchés sont : "scheme", "host", "port", "user", "pass", "path", "query", et "fragment".

6.101.4 rawurldecode() : Décode une chaîne URL.

`string` **rawurldecode** (`string str`)
[lien librairie]

`rawurldecode` retourne une chaîne dont les séquences de caractères %xy, avec xy deux valeurs hexadécimales, auront été remplacées par le caractère ASCII correspondant. Par exemple, la chaîne `foo%20bar%40baz` devient `foo bar@baz` .

Voir aussi `rawurlencode` , `urldecode` et `urlencode` .

6.101.5 rawurlencode() : Encode une chaîne en URL, selon la RFC1738.

`string` **rawurlencode** (`string str`)
[lien librairie]

`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

```
<?php
  echo '<A HREF="ftp://user:', rawurlencode ('foo @+%/'), '@ftp.my.com/x.txt">';
?>
```

Ou, si vous transmettez un chemin dans une URL

Exemple avec rawurlencode

```
<?php
  echo '<A HREF="http://x.com/department_list_script/', rawurlencode ('sales et marketing/Miami'), '
?>
```

Voir aussi rawurldecode , urldecode et urlencode .

6.101.6 urldecode() : Décode une chaîne encodée URL.

string **urldecode** (string str)
[lien librairie]

urldecode décode toutes les séquences % ## et les remplace par leur valeur. La chaîne ainsi décodée est retournée.

Exemple avec urldecode

```
<?php
  $a = split('&', $querystring);
  $i = 0;
  while ($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 .

6.101.7 urlencode() : Encode une chaîne en URL.

string **urlencode** (string str)
[lien librairie]

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), '">';
?>
```

Voir aussi `urldecode` .

Note: Faites bien attention aux variables qui ressemblent à des entités HTML, comme par exemple &amp, &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` .

6.102 Variables

6.102.1 Introduction

Pour plus de détails sur le comportement des variables, reportez vous à la section Variables du chapitre Référence du langage .

6.102.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.102.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.102.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable
unserialize_callback_func	""	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les 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` .

6.102.5 Types de ressources

Cette extension ne définit aucune ressource.

6.102.6 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

6.102.8 empty() : Détermine si une variable est affectée.

`int` **empty** `( mixed var )`
[lien librairie]

`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.

```
<?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.

Voir aussi `isset` et `unset` .

6.102.9 floatval() : Convertit une chaîne en nombre à virgule flottante

`float` **floatval** `( mixed var )`
[lien librairie]

`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.

```
<?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 .

6.102.10 get_defined_vars() : Liste toutes les variables définies

`array` **get_defined_vars** `( void )`
[lien librairie]

`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.

```
<?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` .

6.102.11 get_resource_type() : Retourne le type de ressource

`string` **get_resource_type** `( resource handle )`
[lien librairie]

`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.

```
<?php
$c = mysql_connect();
echo get_resource_type($c)."\n";
// affiche : mysql link
// (lien mysql)
$fp = fopen("foo","w");
echo get_resource_type($fp)."\n";
// affiche : file
// (fichier)
$doc = new_xmldoc("1.0");
echo get_resource_type($doc->doc)."\n";
// affiche : domxml document
// (document domxml)
?>
```

6.102.12 gettype() : Retourne le type de la variable.

`string` **gettype** `( mixed var )`
[lien librairie]

`gettype` retourne le type de la variable PHP `var` .

Les chaînes de caractères que peut retourner la fonction sont les suivantes :

- "boolean"
- "integer"
- "double"
- "string"
- "array"
- "object"
- "resource"
- "user function"
- "unknown type"

Voir aussi `settype` .

6.102.13 import_request_variables() : Importe les variables de GET/POST/Cookie dans l'environnement global

`bool` **import_request_variables** (`string types` , *`string prefix`*)
[lien librairie]

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.

```
<?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` .

6.102.14 intval() : Retourne la valeur numérique (integer) de la variable.

`int` **intval** (`mixed var` , *`int base`*)
[lien librairie]

`intval` retourne la valeur numérique entière (integer) de la variable `var` , en convertisant 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 .

6.102.15 is_array() : Détermine si une variable est un tableau.

bool **is_array** (mixed var)
[lien librairie]

is_array renvoie la valeur TRUE si la variable var est un tableau, FALSE sinon.

Voir aussi is_double , is_float , is_int , is_integer , is_real , is_string , is_long , et is_object .

6.102.16 is_bool() : Détermine si une variable est un tableau booléen

bool **is_bool** (mixed var)
[lien librairie]

is_bool retourne TRUE si var est un booléen.

Voir aussi is_array , is_double , is_float , is_int , is_integer , is_real , is_string , is_long , et is_object .

6.102.17 is_callable() : Détermine si l'argument peut être appelé comme fonction

bool **is_callable** (mixed var , *bool syntax_only* , *string callable_name*)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.102.18 is_double() : Détermine si une variable est de type double.

bool **is_double** (mixed var)
[lien librairie]

is_double renvoie TRUE si la variable var est du type "double", FALSE sinon.

Voir aussi is_array , is_bool , is_float , is_int , is_integer , is_real , is_string , is_long , et is_object .

6.102.19 is_float() : Détermine si une variable est de type float.

bool **is_float** (mixed var)
[lien librairie]

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 .

6.102.20 is_int() : Détermine si une variable est de type integer.

bool **is_int** (mixed var)
[lien librairie]

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 .

6.102.21 is_integer() : Détermine si une variable est de type int.

bool **is_integer** (mixed var)
[lien librairie]

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 .

6.102.22 is_long() : Détermine si une variable est de type integer.

bool **is_long** (mixed var)
[lien librairie]

is_long renvoie TRUE si la variable var est du type entier long (long), FALSE sinon.

Voir aussi is_bool , is_double , is_float , is_int , is_real , is_string , is_object , is_array , et is_integer .

6.102.23 is_null() : Indique si une variable est NULL

bool **is_null** (mixed var)
[lien librairie]

is_null retourne TRUE , si var est NULL , et FALSE .

Voir aussi : is_bool , is_double , is_numeric , is_float , is_int , is_real , is_string , is_object , is_array et is_integer .

6.102.24 is_numeric() : Détermine si une variable est un type numérique

bool **is_numeric** (mixed var)
[lien librairie]

is_numeric retourne TRUE si var est un nombre, ou une chaîne numérique, ou FALSE sinon.

Voir aussi is_bool , is_double , is_float , is_int , is_real , is_string , is_object , is_array , et is_integer .

6.102.25 is_object() : Détermine si une variable est de type object.

bool **is_object** (mixed var)
[lien librairie]

is_object renvoie TRUE si la variable var est un objet, FALSE sinon.

Voir aussi is_bool , is_long , is_int , is_integer , is_float , is_double , is_real , is_string , et is_array .

6.102.26 is_real() : Détermine si une variable est de type real.

bool **is_real** (mixed var)
[lien librairie]

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 .

6.102.27 is_resource() : Détermine si une variable est une ressource

bool **is_resource** (mixed var)
[lien librairie]

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 peut de nettoyage lorsqu'elle sont devenues inutiles, mais pas encore supprimées.

6.102.28 is_scalar() : Indique si une variable est un scalaire

bool **is_scalar** (mixed var)
[lien librairie]

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 virgules flottantes, des chaînes de caractères ou des booléens. Par exemple :

```
<?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"
// }
?>
```

Note
`is_scalar` a été ajoutée en version PHP 4.05.

Voir aussi : `is_bool` , `is_double` , `is_numeric` , `is_float` , `is_int` , `is_real` , `is_string` , `is_object` , `is_array` et `is_integer` .

6.102.29 is_string() : Détermine si une variable est de type string.

`bool` **is_string** (`mixed var`)
[lien librairie]

`is_string` renvoie TRUE si la variable `var` est du type chaine de caractères (string), FALSE sinon.

Voir aussi `is_long` , `is_int` , `is_integer` , `is_float` , `is_double` , `is_real` , `is_object` , et `is_array` .

6.102.30 isset() : Détermine si une variable est affectée.

`int` **isset** (`mixed var`)
[lien librairie]

`isset` renvoie TRUE si la variable `var` est définie, FALSE sinon.

Si une variable a été désaffectée avec la fonction `unset` , la fonction `isset` renverra FALSE .

```
<?php
$a = "test";
echo isset ($a); //
<TT>TRUE</TT>

unset($a);
echo isset ($a); //
<TT>FALSE</TT>

?>
```

Voir aussi `empty` et `unset` .

6.102.31 print_r() : Affiche des informations lisibles pour une variable.

`void` **print_r** (`mixed expression`)
[lien librairie]

Cette fonction 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.

Comparer `print_r` et `var_dump` .

```
<?php
$a = array (1, 2, array ("a", "b", "c"));
print_r ($a);
?>
```

6.102.32 serialize() : Linéarise une variable

`string` **serialize** (`mixed value`)
[lien librairie]

`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! */
    }
}
?>
```

6.102.33 settype() : Affecte un type à une variable.

`int` **settype** (`string var` , `string type`)
[lien librairie]

`settype` modifie le type de la variable `var` en `type` .

Les valeurs possibles pour le paramètre `type` sont :

- "integer"
- "double"
- "string"
- "array"
- "object"

`settype` renvoie TRUE en cas de succès, FALSE sinon.

Voir aussi `gettype` .

6.102.34 strval() : Retourne la valeur de la variable, au format chaîne.

`string` **strval** (`mixed var`)
[lien librairie]

`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 .

6.102.35 unserialize() : Crée une variable PHP à partir d'une valeur linéarisée

`mixed` **unserialize** (`string str`)
[lien librairie]

`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.

Exemple avec `unserialize`

```
<?php
// Ici, on utilise
<A HREF="function.unserialize.php"><TT>unserialize</TT></a>
 pour charger les données de sessions
// depuis la base de données, dans $session_data. Cet exemple complète
// celui fourni avec
<A HREF="function.serialize.php"><TT>serialize</TT></a>
.
$conn = odbc_connect ("webdb", "php", "chicken");
$stmt = odbc_prepare ($conn, "SELECT data FROM sessions WHERE id = ?");
```

```
$sqldata = array ($PHP_AUTH_USER);
if (!odbc_execute ($stmt, &$sqldata) || !odbc_fetch_into ($stmt, &$tmp)) {
    // si la préparation ou la lecture échoue, on crée un tableau vide
    $session_data = array();
} else {
    // les données sauvées sont dasn $tmp[0].
    $session_data = unserialize ($tmp[0]);
    if (!is_array ($session_data)) {
    // Erreur... initialisation à tableau vide
    $session_data = array();
    }
}
?>
```

6.102.36 unset() : Détruit une variable

`void` **unset** (`mixed var` , *`mixed var`* , `...`)
[lien librairie]

`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` .

```
<?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` .

```
<?php
function foo(&$bar) {
    unset($bar);
    $bar = "bla";
}
$bar = 'truc';
echo "$bar\n";
foo($bar);
echo "$bar\n";
?>
```

L'exemple ci dessus va afficher :

```
truc

truc
```

Si une variable statique est détruite à l'intérieure d'une fonction `unset` détruira la référence à la variable statique, plutôt que la variable statique elle même.

```
<?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` :

```
<?php
function foo() {
    unset($GLOBALS['bar']);
}
```

```
$bar = "truc";
foo();
?>
```

Note
`unset` est une structure du langage et non pas une fonction.

Voir aussi `isset` et `empty` .

6.102.37 var_dump() : Dumpe les informations d'une variable.

`void` **var_dump** (`mixed expression`)
[lien librairie]

Cette fonction retourne les informations structurées d'une variable, y compris son type et sa valeur. Les tableaux sont explorés recursivement, avec des indentations, pour mettre en valeur leur structure.

Comparez `var_dump` et `print_r` .

```
<pre>
<?php
    $a = array (1, 2, array ("a", "b", "c"));
    var_dump ($a);
?>
</pre>
```

6.102.38 var_export() : Retourne le code PHP utilisé pour générer une variable

`mixed` **var_export** (`mixed expression` , *`bool return`*)
[lien librairie]

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` .

```
<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>
```

6.103 vpopmail

- 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

6.103.2 vpopmail_add_alias_domain() : Ajout un alias pour un domaine virtuel

`bool` **vpopmail_add_alias_domain** (`string domain` , `string aliasdomain`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.3 vpopmail_add_domain_ex() : Ajoute un nouvaeu domaine virtuel

`bool` **vpopmail_add_domain_ex** (`string domain` , `string passwd` , *`string quota`* , *`string bounce`* , *`bool apop`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.4 vpopmail_add_domain() : Ajoute un nouveau domaine virtuel

`bool` **vpopmail_add_domain** (`string domain` , `string dir` , `int uid` , `int gid`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.5 vpopmail_add_user() : Ajoute un nouvel utilisateur à un domaine virtuel

`bool` **vpopmail_add_user** (`string user` , `string domain` , `string password` , *`string gecos`* , *`bool apop`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.6 vpopmail_alias_add() : Ajoute un alias virtuel

`bool` **vpopmail_alias_add** (`string user` , `string domain` , `string alias`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.7 vpopmail_alias_del_domain() : Efface tous les alias virtuels d'un domaine

`bool` **vpopmail_alias_del_domain** (`string domain`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.8 vpopmail_alias_del() : Efface tous les alias d'un utilisateur

`bool` **vpopmail_alias_del** (`string user` , `string domain`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.9 vpopmail_alias_get_all() : Lit toutes les lignes d'alias d'un domaine

`array` **vpopmail_alias_get_all** (`string domain`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.10 vpopmail_alias_get() : Lit toutes les lignes d'un alias de domaine

`array` **vpopmail_alias_get** (`string alias` , `string domain`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.11 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`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.12 vpopmail_del_domain_ex() : Efface un domaine virtuel

`bool` **vpopmail_del_domain_ex** (`string domain`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.13 vpopmail_del_domain() : Efface un domaine virtuel

`bool` **vpopmail_del_domain** (`string domain`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.14 vpopmail_del_user() : Efface un utilisateur d'un domaine virtuel

`bool` **vpopmail_del_user** (`string user` , `string domain`)

[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.15 vpopmail_error() : Lit le dernier message d'erreur vpopmail

`string` **vpopmail_error** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.16 vpopmail_passwd() : Remplace le mot de passe d'un utilisateur virtuel

`bool` **vpopmail_passwd** (`string user` , `string domain` , `string password`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.103.17 vpopmail_set_user_quota() : Modifie le quota d'un utilisateur virtuel

`bool` **vpopmail_set_user_quota** (`string user` , `string domain` , `string quota`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.104 API windows

6.104.1 Introduction

Cette extension est une API générique d'accès aux DLL. Elle a été originalemnt écrite pour accéder aux API Win32 depuis PHP, mais vous pouvez l'utiliser pour accéder à d'autres fonctions exportées par des DLL.

Actuellement, les types supportés sont les types PHP génériques (chaînes de caractères, booléens, nombres, entiers et NULL), et les types que vous définissez avec `w32api_deftype` .

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 utiliser ce module à vos risques et périls.

6.104.2 Pré−requis

Cette extension ne fonctionne qu'avec le système Windows.

6.104.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

6.104.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.104.5 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"` .

6.104.6 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)

6.104.7 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
$ticks = GetTickCount();

// Conversion en texte lisible
$secs  = floor($ticks / 1000);
$mins  = floor($secs / 60);
$hours = floor($mins / 60);

$str = sprintf("Vous utilisez cet ordinateur depuis  : ".
               "\r\n %d millisecondes, ou \r\n %d secondes".
               "ou \r\n %d minutes ou\r\n %d heures %d minutes.",
               $ticks,
               $secs,
               $mins,
               $hours,
               $mins - ($hours*60));

// Affiche le message dans une boîte de dialogue avec un bouton OK
MessageBoxA(NULL,
            $str,
            "Uptime Information",
            MB_OK);
?>
```


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- Exemples

- w32api_deftype
- w32api_init_dtype
- w32api_invoke_function
- w32api_register_function
- w32api_set_call_method

6.104.9 w32api_init_dtype() : Crée une instance de type de données et la remplit

`resource` **w32api_init_dtype** (`string typename` , `mixed val1` , `mixed val2`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.104.10 w32api_invoke_function() : Appelle une fonction windows 32

`mixed` **w32api_invoke_function** (`string funcname`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce 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` appelle la fonction `funcname` , en lui passant les arguments passés après le nom de la fonction.

6.104.11 w32api_register_function() : Enregistre une fonction win32 dans PHP

`bool` **w32api_register_function** (`string libary` , `string function_name`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.104.12 w32api_set_call_method() : Modifie le nom de la méthode appelée

`void` **w32api_set_call_method** `( int method )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.105 WDDX

Ces fonctions doivent fonctionner avec l'aide de WDDX .

Pour utiliser WDDX, you devez installer la librairie EXPAT (qui est fournie avec la distribution d'Apache 1.3.7 ou plus récent), et recompiler PHP avec `--with-xml` et `--enable-wddx` .

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='0.9'><header comment='Paquet PHP' ><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 */
$cities = array("Paris", "Marseilles", "Lyon");
wddx_add_vars($packet_id, " villes ");
$packet = wddx_packet_end($packet_id);
print $packet;
?>
```

Cet exemple donnera :

```
<wddxPacket version='0.9'><header comment='PHP' ><data><struct>
<var name='pi'><number>3.1415926</number></var><var name='cities'>
<array length='3'><string>Paris</string><string>Marseilles</string>
<string>Lyon</string></array></var></struct></data></wddxPacket>
```

- wddx_add_vars
- wddx_deserialize
- wddx_packet_end
- wddx_packet_start
- wddx_serialize_value
- wddx_serialize_vars

6.105.2 wddx_deserialize() : Lire un paquet WDDX.

`mixed` **wddx_deserialize** (`string packet`)
[lien librairie]

`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.

6.105.3 wddx_packet_end() : Clos un paquet WDDX.

`string` **wddx_packet_end** (`resource packet_id`)
[lien librairie]

`wddx_packet_end` clos un paquet WDDX repéré par son identifiant `packet_id` .

6.105.4 wddx_packet_start() : Commencer un nouveau paquet WDDX avec une structure

`resource` **wddx_packet_start** (*`string comment`*)
[lien librairie]

`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.

6.105.5 wddx_serialize_value() : Enregistrer une valeur dans un paquet WDDX

`string` **wddx_serialize_value** (`mixed var` , *`string comment`*)
[lien librairie]

`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.

6.105.6 wddx_serialize_vars() : Enregistrer plusieurs valeurs dans un paquet WDDX

`string` **wddx_serialize_vars** (`mixed var_name` , *`mixed ...`*)
[lien librairie]

`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='0.9'><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>
```

6.106 Analyseur syntaxique XML

6.106.1 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 .

6.106.2 Pré-requis

Cette extension de PHP utilise `expat` , disponible à http://www.jclark.com/xml/ . Le fichier Makefile livré avec `expat` ne construit pas par défaut de librairie : il faut utiliser la ligne suivante :

```
libexpat.a: $(OBJS)
    ar -rc $@ $(OBJS)
    ranlib $@
```

Les sources RPM de expat sont disponibles à http://sourceforge.net/projects/expat/ .

6.106.3 Installation

Notez que si vous utilisez Apache-1.3.7 ou plus récent, vous disposez déjà de la librairie expat. Configurez simplement PHP avec `--with-xml` (sans aucun autre information) et la librairie expat d'Apache sera automatiquement utilisée.

Sous UNIX, lancez la configuration de PHP avec l'option `--with-xml` , la librairie `expat` étant installée là où votre compilateur peut la trouver. Si vous compilez PHP comme module de PHP 1.3.9 ou plus récent, PHP utilisera automatiquement le module `expat` livré avec Apache. Il vous faudra peut être fixer les valeurs des variables d'environnement **`CPPFLAGS`** et **`LDFLAGS`** , si vous avez fait une installation exotique.

Compilez PHP. **Tada!** C'est fait !

6.106.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.106.5 Types de ressources

Cette extension ne définit aucune ressource.

6.106.6 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)

6.106.7 gestionnaires d'évènements

Les gestionnaires d'évènements XML sont:

Fonction PHP de configuration du gestionnaire	Description de l'événement
`xml_set_element_handler`	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.
`xml_set_character_data_handler`	"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.
xml_set_processing_instruction_handler	Les programmeurs PHP sont habitués aux instructions exécutables (processing instructions ou PIs). <?php ?> 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).
xml_set_default_handler	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.
xml_set_unparsed_entity_decl_handler	Ce gestionnaire est appelé pour gérer les déclaration des entités non analysés.
xml_set_notation_decl_handler	Ce gestionnaire est appelé pour gérer les notations.
xml_set_external_entity_ref_handler	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 entité externe pour un exemple.

6.106.8 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.

6.106.9 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

6.106.10 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 transfert 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).

6.106.11 Quelques exemples

Voici une liste d'exemple de code PHP qui analyse un document XML.

6.106.11.1 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("could not open XML input");
}

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);
?>
```

6.106.11.2 XML Transtypage XML −> HTML

XML 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();
// Utilison la gestion de casse, de manière à être sur 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);
?>
```

6.106.11.3 XML 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 (!eregi("^([a-z]+)://", $file)
        && fileowner($file) == getmyuid()) {
            return true;
    }
    return false;
}

function startElement($parser, $name, $attribs) {
    print "<<font color=\"#0000cc\">$name</font>";
    if (sizeof($attribs)) {
        while (list($k, $v) = each($attribs)) {
            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("XML error: %s at line %d while parsing entity %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("could not open XML input");
}

print "<pre>";
while ($data = fread($fp, 4096)) {
    if (!xml_parse($xml_parser, $data, feof($fp))) {
        die(sprintf("XML error: %s at line %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>
     <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 'Hi!  This is 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 Prédefinies
- gestionnaires d'évènements
- Problèmes de casse
- Codes d'erreurs
- Codage des caractères
- Quelques 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
- 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_external_entity_ref_handler
- xml_set_notation_decl_handler
- xml_set_object
- xml_set_processing_instruction_handler
- xml_set_unparsed_entity_decl_handler

6.106.13 utf8_encode() : Convertit une chaîne ISO−8859−1 en UTF−8.

`string` **utf8_encode** `( string data )`
[lien librairie]

`utf8_encode` code la chaîne data au format `UTF−8` , et retourne la version codée. `UTF−8` est un mécanisme standardisé utilisé par Unicode pour coder les caractère de grande taille dans des flots 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 flot 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	110bbbbb 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.

6.106.14 xml_error_string() : Lit le message d'erreur de l'analyseur XML.

`string` **xml_error_string** (`int code`)
[lien librairie]

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.

6.106.15 xml_get_current_byte_index() : Retourne l'index de l'octet courant d'un analyseur XML.

`int` **xml_get_current_byte_index** (`resource parser`)
[lien librairie]

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.

6.106.16 xml_get_current_column_number() : Retourne le nombre courant de colonne d'un analyseur XML.

`int` **xml_get_current_column_number** (`resource parser`)
[lien librairie]

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.

6.106.17 xml_get_current_line_number() : Retourne le numéro de ligne courant d'un analyseur XML.

int **xml_get_current_line_number** (resource parser)
[lien librairie]

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.

6.106.18 xml_get_error_code() : Retourne le nombre courant de colonne d'un analyseur XML.

int **xml_get_error_code** (resource parser)
[lien librairie]

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.

6.106.19 xml_parse_into_struct() : Analyse une structure XML

int **xml_parse_into_struct** (resource parser , string data , array &values , array &index)
[lien librairie]

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 :

```
<?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'éxécution du code, l'affichage sera :

```
Tableau d'index

Array

(
```

```
    [PARA] => Array
        (
            [0] => 0
            [1] => 2
        )
    [NOTE] => Array
        (
            [0] => 1
        )
)
Tableau de valeurs
Array
(
    [0] => Array
        (
            [tag] => PARA
            [type] => open
            [level] => 1
        )
    [1] => Array
        (
            [tag] => NOTE
            [type] => complete
            [level] => 2
            [value] => simple note
        )
    [2] => Array
        (
            [tag] => PARA
            [type] => close
            [level] => 1
        )
)
```

L'analyse événementielle (comme celle de expat), peut se révéler complexe lorsque le document XML est complexe. `xml_parse_into_struct` ne génère pas d'objet de type DOM, mais il génère plutôt des structures qui peuvent être parcourues à la façon d'un arbre. Considérons le fichier suivant, qui représente une petite base de données XML :

moldb.xml − Petite base de données moléculaire

```
<?xml version="1.0"?>
<moldb>
    <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>
</moldb>
```

Et maitenant, un code qui analyse le document, et génère les objet ad hoc :

parsemoldb.php − analyse moldb.xml et crée un tableau d'objet moléculaires

```
<?php
class AminoAcid {
    var $name;  // nom de l'amino acide
    var $symbol;    // symbole en trois lettres
    var $code;  // code en une lettre
    var $type;  // hydrophobe, chargé ou neutre
    function AminoAcid ($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);
    // parcourt les structures
    foreach ($tags as $key->$val) {
        if ($key == "molecule") {
            $molranges = $val;
            // chaque paire contigue sont les définitions supérieures
            // et inférieures de la molécule
            for ($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);
}
$db = readDatabase("moldb.xml");
echo "** Database of AminoAcid objects:\n";
print_r($db);
?>
```

Après exécution de `parsemoldb.php` , la variable `$db` contient un tableau d'objets `AminoAcid` , et l'affichage le confirme :

```
** Database of AminoAcid objects:

Array

(

    [0] => aminoacid Object

        (

            [name] => Alanine

            [symbol] => ala

            [code] => A

            [type] => hydrophobic

        )

    [1] => aminoacid Object

        (

            [name] => Lysine

            [symbol] => lys

            [code] => K

            [type] => charged

        )

)
```

6.106.20 xml_parse() : Commence l'analyse d'un fichier XML.

`int` **xml_parse** (`resource parser` , `string data` , *`int isFinal`*)
[lien librairie]

`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 `isFinal` est mis à TRUE pour le dernier morceau.

`isFinal` (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` .

6.106.21 xml_parser_create() : Création d'un analyseur XML.

`resource` **xml_parser_create** (*`string encoding`*)
[lien librairie]

`encoding` (optional)

Le codage de caractère de l'analyseur : les codages suivants sont supportés :

`ISO-8859-1` (par défaut)
`US-ASCII`
`UTF-8`

`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.

6.106.22 xml_parser_free() : Détruit un analyseur XML.

`boolean` **xml_parser_free** (`resource parser`)
[lien librairie]

`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 .

6.106.23 xml_parser_get_option() : Lit les options d'un analyseur XML.

`mixed` **xml_parser_get_option** (`resource parser` , `int option`)
[lien librairie]

`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.

6.106.24 xml_parser_set_option() : Affecte les options d'un analyseur XML.

int **xml_parser_set_option** (resource parser , int option , mixed value)
[lien librairie]

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
XML_OPTION_CASE_FOLDING	entier	Contrôle la gestion de la casse des balises de cet analyseur XML. Par défaut, activé.
XML_OPTION_TARGET_ENCODING	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 xml_parser_create . Les codages supportés sont ISO−8859−1 , US−ASCII et UTF−8 .

6.106.25 xml_set_character_data_handler() : Affecte les gestionnaires de caractère bruts.

int **xml_set_character_data_handler** (resource parser , string handler)
[lien librairie]

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.

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.

6.106.26 xml_set_default_handler() : Affecte le gestionnaire par défaut.

int **xml_set_default_handler** (resource parser , string handler)
[lien librairie]

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.

6.106.27 xml_set_element_handler() : Affecte les gestionnaires de début et de fin.

int **xml_set_element_handler** (resource parser , string startElementHandler , string endElementHandler)
[lien librairie]

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: **startElementHandler** 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 `endElementHandler` doit accepter deux paramètres: **`endElementHandler`** `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.

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.

6.106.28 xml_set_external_entity_ref_handler() : Modifie le gestionnaire de référence externes.

`int` **`xml_set_external_entity_ref_handler`** `( resource parser , string handler )`
[lien librairie]

Fixe le gestionnaire d'entité externe de l'analyseur XML `parser` . `handler` et `endElementHandler` sont des chaînes qui contiennent les noms de fonction qui existent lorsque `xml_parse` est appelé pour créer le `parser` .

La fonction `handler` doit accepter 5 paramètres, et retourner un entier. Si la valeur retourné 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` . `int` **`handler`** `( resource parser , string openEntityNames , string base , string systemId , string publicId )`

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 referencé).

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.

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.

6.106.29 xml_set_notation_decl_handler() : Affecte les gestionnaires de notation.

`int` **xml_set_notation_decl_handler** (`resource parser` , `string handler`)
[lien librairie]

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
<TT>name</TT>

{
<TT>systemId</TT>
 |
<TT>publicId</TT>
?>
```

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 notationName` , `string base` , `string systemId` , `string publicId`)

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.

Il n'est pas pour l'instant possible d'utiliser des objets pour servir de gestionnaires. Reportez−vous à `xml_set_object` pour utiliser l'analyseur XML depuis un objet.

6.106.30 xml_set_object() : Utilise un analyseur XML à l'intérieur d'un objet.

`void` **xml_set_object** (`resource parser` , `object &object`)
[lien librairie]

`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);
}
} // Fin de la classe xml
$xml_parser = new xml();
$xml_parser->parse("<A ID=\"bonjour\">PHP</?>");
?>
```

6.106.31 xml_set_processing_instruction_handler() : Affecte les gestionnaires d'instructions exécutables.

`int` **xml_set_processing_instruction_handler** (`resource parser` , `string handler`)

[lien librairie]

Affecte les gestionnaires 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 :

```
<?

<TT>target</TT>

<TT>data</TT>
```

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.

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.

6.106.32 xml_set_unparsed_entity_decl_handler() : Affecte les gestionnaires d'entité non déclaré.

`int` **xml_set_unparsed_entity_decl_handler** `( resource parser , string handler )`
[lien librairie]

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
<TT>name</TT>
 {
<TT>publicId</TT>
 |
<TT>systemId</TT>
}
        NDATA
<TT>notationName</TT>
```

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 entityName` , `string base` , `string systemId` , `string publicId` , `string notationName`)

`parser`
: Le premier paramètre, `parser` , est une référence sur l'analyseur XML qui appelle cette fonction.

`entityName`
: Le nom de l'entité qui va être définie

`base`
: La meilleure base de résolution de l'identifiant système de cette entité externe. Actuellement, ce paramètre est toujours une chaîne vide.

`systemId`
: Identifiant système pour cet entité externe.

`publicId`
: Identifiant public pour cet entité externe.

`notationName`
: Nom de la notation de cette entité. (Voir `xml_set_notation_decl_handler`).

Si un gestionnaire reçoit une chaîne vide ou `FALSE` , c'est qu'il est en train d'être désactivé.

`xml_set_unparsed_entity_decl_handler` retourne `TRUE` si le gestionnaire est actif, et `FALSE` sinon, ou si `parser` n'est pas un analyseur.

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.

6.107 XML−RPC

6.107.1 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 documenation 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 utiliser ce module à vos risques et périls.

6.107.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

6.107.3 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 . Cette extension est fournie avec PHP depuis la version 4.1.0.

6.107.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modiable
xmlrpc_errors	"0"	PHP_INI_SYSTEM
xmlrpc_error_number	"0"	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

6.107.5 Types de ressources

Cette extension ne définit aucune ressource.

6.107.6 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

6.107.8 xmlrpc_decode() : Décode le code XML en types PHP natifs

`array` **xmlrpc_decode** (`string xml` , *`string encoding`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.9 xmlrpc_encode_request() : Génère le code XML d'une requête de méthode

`string` **xmlrpc_encode_request** (`string method` , `mixed params`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.10 xmlrpc_encode() : Génère le code XML pour une valeur PHP

`string` **xmlrpc_encode** (`mixed value`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.11 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 )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.12 xmlrpc_parse_method_descriptions() : Décode le code XML en une liste de descriptions de méthodes

`array` **xmlrpc_parse_method_descriptions** `( string xml )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.13 xmlrpc_server_add_introspection_data() : Ajoute des données d'introspection

`int` **xmlrpc_server_add_introspection_data** `( resource server , array desc )`
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.14 xmlrpc_server_call_method() : Analyse une requête XML est appelle les méthodes associées

`mixed` **xmlrpc_server_call_method** (`resource server` , `string xml` , `mixed user_data` , *`array output_options`*)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.15 xmlrpc_server_create() : Crée un serveur XMLRPC

`resource` **xmlrpc_server_create** (`void`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.16 xmlrpc_server_destroy() : Détruit un serveur XMLRPC

`void` **xmlrpc_server_destroy** (`resource server`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.17 xmlrpc_server_register_introspection_callback() : Enregsitre une fonction PHP pour générer la documentation

`bool` **xmlrpc_server_register_introspection_callback** (`resource server` , `string function`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.18 xmlrpc_server_register_method() : Enregistre une fonction PHP avec une méthode

`bool` **xmlrpc_server_register_method** (`resource server` , `string method_name` , `string function`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.107.19 xmlrpc_set_type() : Modifie le type XMLRPC, base64 ou datetime, pour une chaînes de caractères PHP

`bool` **xmlrpc_set_type** (`string value` , `string type`)
[lien librairie]

Attention
Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.108 XSLT

6.108.1 Introduction

Cette extension PHP fourni une API indépendante pour les transformations XSLT. Actuellement, seul Sablotron est supporté. Le support d'autres librairies 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 XLST 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.

6.108.2 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.

6.108.3 Installation

Sous UNIX, lancez `configure` avec l'option `−−with−sablot` . La librairie `Sablotron` doit être installée là où le compilateur peut la trouver.

Assurez vous que vous avez la même librairie, compilée avec `Sablotron` que celle qui est fournie avec PHP. Les options de configuration : −−with−expat−dir −−with−iconv−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 librairies 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 .

6.108.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.108.5 Types de ressources

Cette extension ne définit aucune ressource.

6.108.6 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.

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_sax_handler

6.108.8 xslt_errno() : Retourne le numéro d'erreur courant

`int` **xslt_errno** (*`resource xh`*)
[lien librairie]

`xslt_errno` le numéro courant d'erreur, pour l'analyseur `xh` . Si `xh` n'est pas fourni, le dernier numéro d'erreur est retourné.

6.108.9 xslt_error() : Retourne le message d'erreur courant

`mixed` **xslt_error** (*`resource xh`*)
[lien librairie]

`xslt_errno` le message d'erreur courant, pour l'analyseur `xh` . Si `xh` n'est pas fourni, le dernier numéro d'erreur est retourné.

6.108.10 xslt_free() : Détruit un analyseur XSLT

`void` **xslt_free** (`resource xh`)
[lien librairie]

`xslt_free` détruit l'analyseur XSLT `xh` .

6.108.11 xslt_process() : Transforme des données XML

`boolean` **xslt_process** (`string xsl_data` , `string xml_data` , `string result`)
[lien librairie]

`xslt_process` prend la chaîne string `xsl_data` comme feuille de style XSLT, et des données XML dans `xml_data` . Le résultat de la transformation sera placé dans `result` . `xslt_process` retourne TRUE en cas de succès, et FALSE sinon. Vous pourrez lire les erreurs survenues grâce aux fonctions `xslt_errno` et `xslt_error` .

Utilisation de `xslt_process` pour transformer trois

```
<?php
$xslData = '<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="article">
    <table border="1" cellpadding="2" cellspacing="1">
        <tr>
            <td width="20%">
              
            </title>
            <td width="80%">
                <h2><xsl:value-of select="title"></h2>
                <h3><xsl:value-of select="author"></h3>
                <br>
                <xsl:value-of select="body">
            </td>
        </tr>
    </table>
</xsl:template>
</xsl:stylesheet>';
$xmlData = '
<?xml version="1.0">
<article>
    <title>Learning German</title>
    <author>Sterling Hughes</author>
    <body>
      Essential phrases:
      <br>
      <br>
      K&#246;nnen Sie mir sagen, wo die Toilette ist?<br>
      Ein grosses Bier, bitte!<br>
      Noch eins, bitte.<br>
    </body>
</article>';
if (xslt_process($xslData, $xmlData, $result))
{
    echo "Voici un brillant article sur l'apprentissage du ";
    echo " français: ";
    echo "<br>\n<br>";
    echo $result;
}
```

```
else
{
    echo "Une erreur est survenue durant le traitement XSL...\n";
    echo "\tErreur numéro : " . xslt_errno() . "\n";
    echo "\tMessage d'erreur : " . xslt_error() . "\n";
    exit;
}
?>
```

6.108.12 xslt_set_sax_handler() : Modifie les gestionnaires SAX de l'analyseur XSLT

`boolean` **xslt_set_sax_handler** `( resource xh , array handlers )`
[lien librairie]

`xslt_set_sax_handler` remplace les gestionnaires SAX de l'analyseur XSLT `xh` par `handlers` .

6.109 YAZ

6.109.1 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 paralléle.

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 celle supportés 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 à http://www.indexdata.dk/yaz/ . Vous pouvez trouvez des informations, des scripts d'exemples, etc... pour cette extension à http://www.indexdata.dk/phpyaz/ .

6.109.2 Installation

Compilez YAZ et installez−le. Compilez PHP avec vos modules et ajoutez l'option `−−with−yaz` . Les instructions sont :

Installation de YAZ

```
gunzip −c yaz−1.6.tar.gz|tar xf −
gunzip −c php−4.0.X.tar.gz|tar xf −
cd yaz−1.6
./configure −−prefix=/usr
make
make install
cd ../php−4.0.X
./configure −−with−yaz=/usr/bin
make
make install
```

6.109.3 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` .

6.109.4 Types de ressources

Cette extension ne définit aucune ressource.

6.109.5 Constantes Prédefinies

Cette extension ne définit aucune constante.

6.109.6 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 paralléle. Lorsqu'il est appelé sans paramètre, ce script affiche la requête. Sinon, il effectue la recherche sur les serveurs.

Recherche paralléle 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($term).'<br>';
    for ($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].":";
        $error = yaz_error($id[$i]);
        if (!empty($error)) {
            echo "Erreur: $error";
        } else {
            $hits = yaz_hits($id[$i]);
            echo "Nombre de résultats : $hits";
        }
        echo '<dl>';
        for ($p = 1; $p <= 10; $p++) {
            $rec = yaz_record($id[$i],$p,"string");
            if (empty($rec)) continue;
            echo "<dt><B>$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_hits
- yaz_itemorder
- yaz_present
- yaz_range
- yaz_record
- yaz_scan_result
- yaz_scan
- yaz_search
- yaz_sort
- yaz_syntax
- yaz_wait

6.109.8 yaz_ccl_conf() : Configure l'analyseur CCL

`int` **yaz_ccl_conf** (`int id` , `array config`)
[lien librairie]

`yaz_ccl_conf` configure l'analyseur CCL de requete 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 champs CCL et la valeur correspondante contient une chaîne spécifiant le code RPN. Ce code est une séquence de paires "attribut−type, attribute−value". Les "attribut−type" et "attribut−value" sont séparé par le signe égal (=). Chaque paire est séparé 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 champs correspond à leur équivalent équivalent BIB−1. On suppose que chaque variable `$id` est un hôte de destination.

6.109.9 yaz_ccl_parse() : Appelle l'analyseur CCL

`int` **yaz_ccl_parse** (`int id` , `string query` , `array & result`)
[lien librairie]

yaz_ccl_parse appelle l'analyseyr 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 champs 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, champs 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).

6.109.10 yaz_close() : Ferme une connexion YAZ

`int` **yaz_close** (`int id`)
[lien librairie]

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 .

6.109.11 yaz_connect() : Prépare une connexion à un hôte YAZ

`int` **yaz_connect** (`string zurl` , *`string authentication`*)
[lien librairie]

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 .

6.109.12 yaz_database() : Spécifie la base d'une session

`int` **yaz_database** (`int id` , `string databases`)
[lien librairie]

yaz_database spécifie databases , la ou les bases utilisées lors des recherches, lectures, etc, en remplaçant celles spécifiées lors de la fonction 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.

yaz_database retourne TRUE en cas de succès, ou FALSE en cas d'erreur.

6.109.13 yaz_element() : Spécifie le type d'éléments à lire

`int` **yaz_element** (`int id` , `string elementset`)
[lien librairie]

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).

yaz_element retourne TRUE en cas de succès, et FALSE sinon.

6.109.14 yaz_errno() : Retourne le numéro d'erreur

int **yaz_errno** (int id)
[lien librairie]

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.

6.109.15 yaz_error() : Retourne une description de l'erreur

string **yaz_error** (int id)
[lien librairie]

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 .

6.109.16 yaz_hits() : Retourne le nombre de résultat de la dernière recherche

int **yaz_hits** (int id)
[lien librairie]

yaz_hits retourne le nombre de résultat de la dernière recherche.

6.109.17 yaz_itemorder() : Prépare une requête Z39.50 Item Order avec le package ILL−Request

int **yaz_itemorder** (array args)
[lien librairie]

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)" (Note du Traducteur : mailez moi de l'aide!). Reportez−vous ici ou aux spécification . 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-addreess
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-addreess
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-complicance
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
```

6.109.18 yaz_present() : Prépare à la lecture (Z39.50 present).

`int` **yaz_present** (`void`)
[lien librairie]

`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 résultat à lire.

6.109.19 yaz_range() : Spécifie le nombre maximal de résultat à lire

`int` **yaz_range** (`int id` , `int start` , `int number`)
[lien librairie]

`yaz_range` est utilisée conjointement à `yaz_search` , pour spécifier le nombre maximal `number` de résultat 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.

`yaz_range` retourne TRUE en cas de succès; FALSE en cas d'erreur.

6.109.20 yaz_record() : Retourne un résultat

`int` **yaz_record** (`int id` , `int pos` , `string type`)
[lien librairie]

`yaz_record` retourne un résultatà la position `pos` , ou une chaîicirc;ne 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 chaîicirc;ne 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 chaîicirc;ne, prête à l'affichage. Si type vaut "array", la ligne sera retournée sous la forme d'un tableau structuré.

6.109.21 yaz_scan_result() : Retourne le résultat d'un scan

array **yaz_scan_result** (int id , *array & result*)
[lien librairie]

yaz_scan_result retourne un tableau contenant les termes recu 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 (numbre d'entrée retournées), stepsize (taille du pas), position (position du terme), status (Statut du Scan).

6.109.22 yaz_scan() : Prépare un scan

int **yaz_scan** (int id , string type , string startterm , *array flags*)
[lien librairie]

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 reéllement 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

```
function scan_titles($id, $starterm) {
  yaz_scan($id,"rpn", "&#64;attr 1=4 " . $starterm);
  yaz_wait();
  $errno = yaz_errno($id);
  if ($errno == 0) {
    $ar = yaz_scan_result($id,&$options);
    echo 'Scan ok; ';
    $ar = yaz_scan_result($id, &$options);
    while(list($key,$val)=each($options)) {
       echo "$key = $val  ";
    }
    echo '<br><table><tr><td>';
    while(list($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>";
  }
```

```
    }
```

6.109.23 yaz_search() : Prépares une recherche

`int` **yaz_search** (`int id` , `string type` , `string query`)
[lien librairie]

`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` .

6.109.24 yaz_sort() : Configure les critères de tri

`int` **yaz_sort** (`int id` , `string criteria`)
[lien librairie]

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 ser 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 champs spécifie soit une combinaison d'attributs numériques, constituée de paire type=valeur, séparée par des virgules (e.g. `1=4,2=1`) , soit le champs spécifie un critère en chaîne de caractère (e.g. `title`). Les options (flags) sont des séquences de caractères, choisis parmi les valeurs suivantes. Ces caractères ne peuvent être séparé 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`

6.109.25 yaz_syntax() : Spécifie la syntaxe de lecture des lignes

`int` **yaz_syntax** (`int id` , `string syntax`)

[lien librairie]

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.

6.109.26 yaz_wait() : Attend l'exécution d'une requête

int **yaz_wait** (void)
[lien librairie]

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).

6.110 NIS

6.110.1 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.

6.110.2 Installation

Pour ajouter ces fonctionnalitées, vous devez compiler PHP avec l'option `--with-yp` (PHP 3) ou `--enable-yp` (PHP 4).

6.110.3 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.

YPERR_BADARGS (entier)
YPERR_BADDB (entier)
YPERR_BUSY (entier)
YPERR_DOMAIN (entier)
YPERR_KEY (entier)
YPERR_MAP (entier)
YPERR_NODOM (entier)
YPERR_NOMORE (entier)
YPERR_PMAP (entier)
YPERR_RESRC (entier)
YPERR_RPC (entier)
YPERR_YPBIND (entier)
YPERR_YPERR (entier)
YPERR_YPSERV (entier)
YPERR_VERS (entier)

- Introduction
- Installation
- 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

6.110.5 yp_cat() : Retourne un tableau contenant toute la carte

`array` **yp_cat** (`string domain` , `string map`)
[lien librairie]

Attention
Cette fonction n'est pas documentée, et seule la liste des arguments est disponible.

6.110.6 yp_err_string() : Retourne la chaîne d'erreur associée à la dernière opération.

`string` **yp_err_string** (`void`)
[lien librairie]

`yp_err_string` Retourne la chaîne d'erreur associée à la dernière opération. Très pratique pour savoir exactement ce qui n'est pas allé.

Exemple

```
<?php
    echo "Error: " . yp_err_string();
?>
```

Voir aussi yp_errno .

6.110.7 yp_errno() : Retourne le code d'erreur de la dernière opération.

`int` **yp_errno** (`void`)
[lien librairie]

`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 .

6.110.8 yp_first() : Retourne le premier couple (clé ; valeur) d'une carte donnée.

`array` **yp_first** (`string domain` , `string map`)
[lien librairie]

`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` .

6.110.9 yp_get_default_domain() : Retourne le domaine NIS par défaut.

`int` **yp_get_default_domain** (`void`)
[lien librairie]

`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;
?>
```

6.110.10 yp_master() : Retourne le nom de la machine maître pour une carte.

`string` **yp_master** (`string domain` , `string map`)
[lien librairie]

`yp_master` retourne le nom de la machine maître d'une carte.

Exemple de maître NIS

```php
<?php
    $number = yp_master($domain, $mapname);
    echo "Master for this map is: " . $master;
?>
```

Voir aussi `yp-get-default-domain` .

6.110.11 yp_match() : Retourne la ligne associée.

`string` **yp_match** (`string domain` , `string map` , `string key`)
[lien librairie]

`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
<?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` .

6.110.12 yp_next() : Retourne le couple (clé ; valeur) suivant d'une carte donnée.

`array` **yp_next** (`string domain` , `string map` , `string key`)
[lien librairie]

`yp_next` retourne le couple (clé ; valeur) suivant la clé donnée d'une carte donnée ou `FALSE` .

Exemple avec `yp_next`

```php
<?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` .

6.110.13 yp_order() : Retourne le numéro d'ordre d'une carte.

`int` **yp_order** (`string domain` , `string map`)
[lien librairie]

`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` .

6.111 Accès aux fichiers Zip (Lecture seule)

6.111.1 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.

6.111.2 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 pourcréer l'archive elle−même.

6.111.3 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.

6.111.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

6.111.5 Types de ressources

Cette extension ne définit aucune ressource.

6.111.6 Constantes Prédefinies

Cette extension ne définit aucune constante.

6.111.7 Exemples

Cet exemple ouvre un fichier ZIP, lire chaque fichier de l'archive, et en affiche le contenu. Le fichier d'archive `test2.zip` utilisé dans cet exemple est fourni dans la distribution source de ZZIPlib.

Exemple d'utilisation de ZIP

```php
<?php

$zip = zip_open("/tmp/test2.zip");

if ($zip) {

    while ($zip_entry = zip_read($zip)) {
        echo "Nom :               " . zip_entry_name($zip_entry) . "\n";
        echo "Taille réelle du fichier : "  . zip_entry_filesize($zip_entry) . "\n";
        echo "Taille compressée du fichier : " . zip_entry_compressedsize($zip_entry) . "\n";
        echo "Méthode de compression : " . zip_entry_compressionmethod($zip_entry) . "\n";

        if (zip_entry_open($zip, $zip_entry, "r")) {
            echo "Contenu du fichier : \n";
            $buf = zip_entry_read($zip_entry, zip_entry_filesize($zip_entry));
            echo "$buf\n";

            zip_entry_close($zip_entry);
        }
        echo "\n";

    }

    zip_close($zip);

}

?>
```

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes 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

6.111.9 zip_entry_close() : Ferme un dossier d'archive

`void` **zip_entry_close** (`resource zip_entry`)
[lien librairie]

`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 .

6.111.10 zip_entry_compressedsize() : Lit la taille compressée d'un dossier d'archive

int **zip_entry_compressedsize** (resource zip_entry)
[lien librairie]

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 .

6.111.11 zip_entry_compressionmethod() : Lit la méthode de compression utilisée sur un dossier d'archive

string **zip_entry_compressionmethod** (resource zip_entry)
[lien librairie]

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 .

6.111.12 zip_entry_filesize() : Lit la taille décompressée d'un dossier d'archive

int **zip_entry_filesize** (resource zip_entry)
[lien librairie]

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 .

6.111.13 zip_entry_name() : Lit le nom du dossier d'archive

string **zip_entry_name** (resource zip_entry)
[lien librairie]

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 .

6.111.14 zip_entry_open() : Ouvre un fichier d'archive en lecture

`bool` **zip_entry_open** (`resource zip` , `resource zip_entry` , *`string mode`*)
[lien librairie]

`zip_entry_open` ouvre un fichier dans une archive 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` .

6.111.15 zip_entry_read() : Lit le contenu d'un fichier dans un dossier

`string` **zip_entry_read** (`resource zip_entry` , *`int length`*)
[lien librairie]

`zip_entry_read` lit jusqu'à `length` octets dans le fichier représentée 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
The `length` parameter should be the uncompressed length you wish to read.

`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` .

6.111.16 zip_open() : Ouvre une archive ZIP

`resource` **zip_open** (`string filename`)
[lien librairie]

`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` .

6.111.17 zip_read() : Lit la prochaine entrée dans une archive ZIP

`resource` **zip_read** (`resource zip`)
[lien librairie]

`zip_read` lit la prochaine entrée dans l'archive ZIP. Le paramètre `zip` est une archive valide, retourné par `zip_open` .

`zip_read` retourne une ressource de dossier d'archive, à utiliser plus tard avec les autres fonctions de la librairie.

Voir aussi `zip_open` , `zip_close` , `zip_entry_open` et `zip_entry_read` .

6.112 Compression Zlib

6.112.1 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 fichiesr 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 ambiguités avec les fichiers contenant des deux points `:` . Cette fonctionnalité requiert une librarie C qui fournisse la fonction `fopencookie()` . A ma connaissance, la librairie GNU libc est la seule à fournir cette fonctionnalité.

6.112.2 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.

6.112.3 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 comprésser 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 fichier de configuration `php.ini` .

Nom	Par défaut	Modifiable
zlib.output_compression	"Off"	PHP_INI_ALL
zlib.output_compression_level	"−1"	PHP_INI_ALL
zlib.output_handler	""	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à `ini_set` .

Voici de détails sur les directives de configuration.

`zlib.output_compression` *`booléen / entier`*
: Active ou pas la compression transparente des pages. Si cette option est mise à "On" dans ou dans la configuration Apache, les pages sont compressées si le navigateur envoie un entête "Accept−Encoding: gzip" ou "deflate". Les entêtes "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 entêtes n'ont pas encore été envoyé. Si vous émettez un entête 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 entêtes ait 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.

6.112.4 Types de ressources

Cette extension ne définit aucune ressource.

6.112.5 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.

`FORCE_GZIP` (entier)
`FORCE_DEFLATE` (entier)

6.112.6 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
- Configuration à l'exécution
- Types de ressources
- Constantes Prédefinies
- Exemples
- gzclose
- gzcompress
- gzdeflate
- gzencode
- gzeof
- gzfile
- gzgetc
- gzgets
- gzgetss
- gzinflate
- gzopen
- gzpassthru
- gzputs
- gzread
- gzrewind
- gzseek
- gztell
- gzuncompress
- gzwrite
- readgzfile

6.112.8 gzcompress() : Compresse une chaîne

`string` **gzcompress** (`string data` , *`int level`*)
[lien librairie]

`gzcompress` compresse 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 de 0 jusqu'à 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 entêtes. Voyez la fonction `gzencode` pour cela.

Voir aussi `gzdeflate` , `gzinflate` , `gzuncompress` et `gzencode` .

6.112.9 gzdeflate() : Compresse une chaîne avec la méthode DEFLATE

`string` **gzdeflate** (`string data` , *`int level`*)
[lien librairie]

`gzdeflate` compresse la chaîne `data` avec la méthode DEFLATE, ou `FALSE` si une erreur survient. Le paramètre optionel `level` peut prendre les valeurs de 0 (pas de compression) jusqu'à 9 (compression maximale). Par défaut, le niveau de compression est le niveau de la librairie Zlib. Zlib.

Pour plus de détails sur l'algorithme de compression ZLIB, reportez−vous à la RFC " ZLIB Compressed Data Format Specification version 4.3 " (RFC 1950).

Voir aussi `gzinflate` , `gzcompress` , `gzuncompress` et `gzencode` .

6.112.10 gzencode() : Crée une chaîne compressée

`string` **gzencode** (`string data` , *`int level`* , *`int encoding_mode`*)
[lien librairie]

`gzencode` retourne la version compressée de la chaîne `data` , compatible avec le résultat de la commande `gzip` program. Cette fonction retourne `FALSE` si une erreur survient. Le paramètre optionel `level` peut prendre les valeurs de 0 (pas de compression) jusqu'à 9 (compression maximale). Par défaut, le niveau de compression est le niveau de la librairie Zlib.

Vous pouvez aussu 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 entêtes 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 résultant contient les entêtes et la structure de données appropriée pour en faire un fichier .gz standard, c'est à dire :

Créer un fichier gzip

```php
<?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 l'algorithme de compression ZLIB, reportez−vous à la RFC " ZLIB Compressed Data Format Specification version 4.3 " (RFC 1950).

Voir aussi `gzcompress` `gzuncompress`, `gzdeflate` et `gzinflate` .

6.112.11 gzeof() : Indique si la fin d'un fichier compressé est atteinte

`int` **gzeof** (`int zp`)
[lien librairie]

`gzeof` retourne TRUE si le pointeur de fichier du fichier compressé `zp` a été atteinte, et retourne FALSE sinon.

`zp` doit être un fichier compressé valide, retourné par la fonction `gzopen` .

6.112.12 gzfile() : Lit la totalité d'un fichier compressé

`array` **gzfile** (`string filename` , *`int use_include_path`*)
[lien librairie]

`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` .

6.112.13 gzgetc() : Lit un caractère dans un fichier compressé

`string` **gzgetc** (`int zp`)
[lien librairie]

`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 (tout comme `gzeof`).

`zp` doit être un fichier compressé valide, retourné par la fonction `gzopen` .

Voir aussi `gzopen` et `gzgets` .

6.112.14 gzgets() : Lit une ligne dans un fichier compressé

`string` **gzgets** (`int zp` , `int length`)
[lien librairie]

`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 ausi `gzopen` , `gzgetc` et `fgets` .

6.112.15 gzgetss() : Lit une ligne dans un ficheir compressé, et supprime les balises

`string` **gzgetss** (`int zp` , `int length` , *`string allowable_tags`*)
[lien librairie]

`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
`Allowable_tags` a été ajouté en PHP 3.0.13, et PHP 4.0b3.

Voir aussi `gzgets` , `gzopen` et `strip_tags` .

6.112.16 gzinflate() : Décompresse une chaîne

`string` **gzinflate** (`string data` , *`int length`*)
[lien librairie]

`gzinflate` retourne 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` .

6.112.17 gzopen() : Ouvre un fichier compressé avec gzip

`int` **gzopen** (`string filename` , `string mode` , *`int use_include_path`*)
[lien librairie]

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 transaparente.

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 .

gzopen Example

```
<?php
  $fp = gzopen ("/tmp/file.gz", "r");
?>
```

Voir aussi gzclose .

6.112.18 gzpassthru() : Affiche le contenu d'un fichier en une seule fois

int **gzpassthru** (int zp)
[lien librairie]

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.

6.112.19 gzputs() : Ecrit une chaîne dans un fichier compressé

int **gzputs** (int zp , string str , *int length*)
[lien librairie]

gzputs est un alias de la fonction gzwrite , et lui est identique en tous points.

6.112.20 gzread() : Lecture de fichier compressé binaire

`string` **gzread** (`int zp` , `int length`)
[lien librairie]

`gzread` lit jusqu'à `length` octets dans le fichier compressé gzip, représenté par `zp` . La lecture s'arrête lorsque `length` octets (décompressé) on 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");
  $contents = gzread($zd, 10000);
  gzclose($zd);
?>
```

Voir aussi `gzwrite` , `gzopen` , `gzgets` , `gzgetss` , `gzfile` et `gzpassthru` .

6.112.21 gzrewind() : Recommence la lecture du fichier

`int` **gzrewind** (`int zp`)
[lien librairie]

`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` .

6.112.22 gzseek() : Déplace le pointeur de lecture

`int` **gzseek** (`int zp` , `int offset`)
[lien librairie]

`gzseek` place le pointeur de lecture du fichier `zp` à l'offset `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 fonction est ouvert en lecture, cette fonction est alors émulée, et se révèle extrêment lente. Si le fichier est ouvert en lecture, 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, `gzseek` retourne −1. Notez que placer le pointeur au dela de la fin du fichier n'est pas une erreur.

Voir aussi `gztell` et `gzrewind` .

6.112.23 gztell() : Lit la position courante du pointeur de lecture

`int` **gztell** (`int zp`)
[lien librairie]

`gztell` retourne la position du pointeur de lecture dans le fichier `zp` , c'est à dire son offset 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` .

6.112.24 gzuncompress() : Décompresse une chaîne compressée

`string` **gzuncompress** (`string data` , *`int length`*)
[lien librairie]

`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` .

6.112.25 gzwrite() : Ecrit dans un fichier compressé, en gzip

`int` **gzwrite** (`int zp` , `string string` , *`int length`*)
[lien librairie]

`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 on été écrit, ou à la fin de la chaîne `string` : le premier qui arrive.

Notez que si `length` est donné, et que magic_quotes_runtime sera ignoré, et les slashs ne seront pas supprimés de la chaîne `string` .

Voir aussi `gzread` , `gzopen` et `gzputs` .

6.112.26 readgzfile() : Lit tout le fichier compressé

`int` **readgzfile** (`string filename` , *`int use_include_path`*)
[lien librairie]

`readgzfile` lit le fichier `filename` , le décompresse et affiche le résultat.

`readgzfile` peut aussi service à lire un fichier qui n'est pas compressé : dans ce cas, `readgzfile` va lire le fichier sans le décompressé.

`readgzfile` retourne le nombre d'octets décompressé 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` .

fr/ should have been there

7 Streams API for PHP Extension Authors

7.1 Overview

The PHP Streams API introduces a unified approach to the handling of files and sockets in PHP extension. Using a single API with standard functions for common operations, the streams API allows your extension to access files, sockets, URLs, memory and script−defined objects. Streams is a run−time extensible API that allows dynamically loaded modules (and scripts!) to register new streams.

The aim of the Streams API is to make it comfortable for developers to open files, urls and other streamable data sources with a unified API that is easy to understand. The API is more or less based on the ANSI C stdio family of functions (with identical semantics for most of the main functions), so C programmers will have a feeling of familiarity with streams.

The streams API operates on a couple of different levels: at the base level, the API defines php_stream objects to represent streamable data sources. On a slightly higher level, the API defines php_stream_wrapper objects which "wrap" around the lower level API to provide support for retrieving data and meta−data from URLs.

Streams can be cast (converted) into other types of file−handles, so that they can be used with third−party libraries without a great deal of trouble. This allows those libraries to access data directly from URL sources. If your system has the `fopencookie` function, you can even pass any PHP stream to any library that uses ANSI stdio!

7.2 Streams Basics

Using streams is very much like using ANSI stdio functions. The main difference is in how you obtain the stream handle to begin with. In most cases, you will use `php_stream_open_wrapper` to obtain the stream handle. This function works very much like fopen, as can be seen from the example below:

simple stream example that displays the PHP home page

```
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);
}
```

The table below shows the Streams equivalents of the more common ANSI stdio functions. Unless noted otherwise, the semantics of the functions are identical.

ANSI Stdio Function	PHP Streams Function	Notes
fopen	php_stream_open_wrapper	Streams includes additional parameters
fclose	php_stream_close	
fgets	php_stream_gets	
fread	php_stream_read	The nmemb parameter is assumed to have a value of 1, so the prototype looks more like read(2)
fwrite	php_stream_write	The nmemb parameter is assumed to have a value of 1, so the prototype looks more like 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	Same semantics as puts, NOT fputs
fstat	php_stream_stat	Streams has a richer stat structure

8 FAQ : Foire Aux Questions

8.1 General Information

This section holds the most general questions about PHP: what it is and what it does.

8.1.1 faq.general.what

This section holds questions about how to get in touch with the PHP community. The best way is the mailing lists.

8.2.1 faq.mailinglist.isthere

This section has details about PHP download locations, and OS issues.

8.3.1 faq.obtaining.where

This section holds common questions about relation between PHP and databases. Yes, PHP can access virtually any database available today.

8.4.1 faq.databases.mssql

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 .

8.5.1 faq.installation.phpinior' −−!>

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 NTLanMan permissions on your script file.

8.5.10 faq.install.phpandiis

This section gathers most common errors that occur at build time.

8.6.1 faq.build.configurescript 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` .

8.6.13 faq.build.microtime

This section gathers most common errors you can face, while writing PHP scripts.

8.7.1 faq.using.anyform

PHP and HTML interact a lot: PHP can generate HTML, and HTML can pass information to PHP.

8.8.1 faq.html.encoding 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[]'];
```

8.9 PHP and COM

PHP can be used to access COM and DCOM objects on Win32 platforms.

8.9.1 faq.com.q1

PHP is the best language for web programing, but what about other languages?

8.10.1 faq.languages.asp

PHP has already a long history behind him: Legendary PHP 1.0, PHP/FI, PHP 3.0 and PHP 4.0.

8.11.1 faq.migrating.phpfi

PHP has already a long history behind him : Legendary PHP 1.0, PHP/FI, PHP 3.0 and PHP 4.0.

8.12.1 faq.migration4.php34

There can be some questions we can't put into other categories. Here you can find them.

8.13.1 faq.misc.popup

9 Appendices

9.1 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.

9.1.1 Histoire de PHP

9.1.1.1 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 'Personnal 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, Interpreteur 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 publié.

9.1.1.2 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.

9.1.1.3 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.

9.1.2 Quelques projets liés à PHP

9.1.2.1 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 crée 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 inclus une vaste sélection de classes de bases pour réaliser une couche d'abstraction de bases de données, du cache, des calculs mathématiques et des transactiosn de commerce électronique.

9.1.2.2 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 constitué de 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.

9.1.2.3 PHP−GTK

PHP−GTK est la solution PHP pour écrire des applications avec interface, coté 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 traités à la main, dans le fichier .overrides.

Travailler sur le générateur de code et sur l'infrastructure pris 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 concues.

9.1.3 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 paralellement.

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 témoingnage du succès planétaire de PHP.

9.1.4 Publications about PHP

Autant que nous le sachions, le premier article consacré à PHP dans un magasine papier fut publié dans un magazine Francais, 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 librarie MSDN de Microsoft.

9.2 Migration de PHP 3.0 à PHP 4.0

9.2.1 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'effecteur le changement de cadre de production. Les indications suivantes vous mettront sur la voie.

9.2.2 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 :

```
$ ./configure \
  --with-apxs=/apache/bin/apxs \
  --enable-versioning \
  --with-mysql \
  --enable-track-vars
```

9.2.3 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.

```
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 :

```
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 :

```
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 des 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 . A 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 :

```
~/php4/scripts/apache:#  ./apconf−conv.sh /usr/local/apache/conf/httpd.conf
```

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 :

```
~/php4/scripts/apache:#  find / −name .htaccess −exec ./aphtaccess−conv.sh {} \;
```

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.

9.2.4 Comportement de l'analyseur

L'analyse et l'éxecution sont désormais deux étapes complètement dissociées, et l'éxécution 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.

9.2.5 Rapport d'erreur

9.2.5.1 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.

9.2.5.2 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 : `$HTTP_SERVER_VARS[REQUEST_METHOD]` . Modifier ce code en `$HTTP_SERVER_VARS['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.

9.2.6 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.

9.2.7 empty("0")

L'évolution la plus polémique est celle de `empty` . Une chaîne contenant seulement le caractère `'0'` (zéro) est maintenant considérée comme vide, alors qu'elle ne l'était pas en PHP 3.0.

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.

9.2.8 Fonctions manquantes

Bien que PHP 4.0 dispose de nombreuses nouvelles fonctionnalités fonctions et extensions, vous vous rencontrer 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'éxécution ont été séparées. D'autres fonctions, et mêmes 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.

9.2.8.1 Fonctions manquantes pour des raisons de structure

Comme PHP 4.0 sépare l'analyse et l'éxécution, il n'est plus possible de modifier le comportement de l'analyseur (intégré dans le moteur Zend) durant l'éxécution, 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 a disparu est le débuggeur de PHP 3.0, comme décrit dans un autre appendice. Un nouveau débuggeur est promis par Zend, mais il n'a pas encore montré le bout de son nez.

9.2.8.2 Fonctions et extensions obsolètes

Les extensions Adabas et Solid n'existent plus. Elles sont intégrées dans les fonctions ODBC Unifié .

9.2.8.3 N unset

`unset` , bient 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.

9.2.9 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).

9.2.10 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 recherche les séquences `{$` dans votre code, et en les remplaçant par `\{$` avec votre outil de remplacement habituel.

9.2.11 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.

9.3 Migration de PHP/FI 2.0 à PHP 3.0

9.3.1 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 possible 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.

9.3.2 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

```
<?php
  echo "Ceci est du code PHP/FI 2.0.\n";?
?>
```

Comme en version 2.0, PHP/FI accepte aussi cette variante :

Migration: premières nouvelles balises PHP

```
<?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>
```

9.3.3 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 à changée :

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).

9.3.4 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 !

9.3.5 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 != "") {
?>
```

9.3.6 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.

9.3.7 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.

9.3.8 L TRUE / FALSE

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");
?>
```

9.3.9 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";
?>
```

9.4 Débuggeur PHP

9.4.1 A propos du débuggeur

PHP 3 inclut le support d'un serveur de débuggage.

PHP 4 n'inclut aucun support de débuggage.

9.4.2 Utiliser le débuggeur PHP

Le débuggeur PHP sert à repérer les bugs récalcitrants. Le débuggeur 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 débuggeur :

1. Réservez un port TCP pour le débuggeur dans le fichier de configuration (debugger.port) et activez−le (debugger.enabled).
2. Configurer un client TCP sur ce port (par exemple `socket −l −s 1400` sous UNIX).
3. Dans votre code, placez la ligne "debugger_on(`host`)", où `host` est l'IP ou le nom de l'hôte qui supporte le client TCP .

Désormais, toutes les alertes, notes, ... seront envoyées sur la socket client, **même si vous avez inactivé le rapport d'erreur avec `error_reporting`** .

9.4.3 Protocole du débuggeur

Le protocole de débugage PHP 3 fonctionne ligne par ligne. Chaque ligne a un type **type** et plusieurs lignes composent un message Chaque message commence avec une ligne du type `start` et se termine avec une ligne de type `end` . PHP peut envoyer des lignes de plusieurs messages simultanément.

Voici un exemple de ligne à ce format :

```
date

time

host
(
pid
)

type
:

message−data
```

`date`
: Les dates sont au format ISO 8601 (`yyyy − mm − dd`)

`time`

Les heures, y compris les micro−secondes: hh : mm : uuuuuu

host

Le nom DNS ou adresse IP de l'hôte qui a généré l'erreur.

pid

PID (process id) sur l'hôte host , qui a généré l'erreur.

type

Type de la ligne. Indique au programme client comment traiter les données suivantes :

Nom	Signification
start	Indique au programme client que le message du débuggeur commence ici. Le contenu de data sera un type d'erreur, comme listé ci−dessous.
message	Le message d'erreur PHP 3.
location	Nom du fichier et numéro de ligne, où l'erreur est survenue. La première occurrence de location contiendra toujours la localisation générale. data contiendra : file : line . Il y a toujours une indication de location après un message et après chaque function .
frames	Nombre de frames dans le dump de la pile. S'il y a 4 frames, attendez vous à des informations sur 4 niveaux de fonctions. Si la ligne "frame" n'existe pas, la profondeur doit être 0 (une erreur est survenue au niveau général).
function	Nom de la fonction qui a généré l'erreur. Elle sera répétée à chaque niveau de la pile d'appel.
end	Indique au client que le message du débuggeur se termine ici.

data

Ligne de données.

Débuggeur	Interne PHP 3
alerte (warning)	E_WARNING
erreur	E_ERROR
analyse (parse)	E_PARSE
note (notice)	E_NOTICE
core−error	E_CORE_ERROR
core−warning	E_CORE_WARNING
inconnue	(toutes les autres)

Exemple de message du débuggeur

```
1998−04−05 23:27:400966 lucifer.guardian.no(20481) start: notice
1998−04−05 23:27:400966 lucifer.guardian.no(20481) message: Uninitialized variable
1998−04−05 23:27:400966 lucifer.guardian.no(20481) location: (
NULL
):7
1998−04−05 23:27:400966 lucifer.guardian.no(20481) frames: 1
1998−04−05 23:27:400966 lucifer.guardian.no(20481) function: display
1998−04−05 23:27:400966 lucifer.guardian.no(20481) location: /home/ssb/public_html/test.php3:10
1998−04−05 23:27:400966 lucifer.guardian.no(20481) end: notice
```

9.5 Développement PHP

9.5.1 Créer une fonction PHP 3

9.5.1.1 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.

9.5.1.2 Arguments de fonctions

Les arguments sont toujours de type val. Ce type contient un membre de type union, qui indique le type reél 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 un nouvelle chaîne avec estrdup() et une nouvelle longueur de chaîne. Seulement après, vous pouvez modifier le type en IS_STRING. Si vous modifiez une chaîne en IS_STRING ou IS_ARRAY vous devez d'abord appeler le destructeur pval_destructor.

9.5.1.3 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;
}
```

9.5.1.4 Utiliser les arguments d'une fonction

De type de chaque argument est stocké dans le champs pval. Ce champs peut prendre les valeurs suivantes :

IS_STRING	Chaîne de caractères
IS_DOUBLE	Nombre à virgule flottante, en précision double
IS_LONG	Entier long
IS_ARRAY	Tableau
IS_EMPTY	Aucune
IS_USER_FUNCTION	??
IS_INTERNAL_FUNCTION	?? (Si ce type ne peut pas être passé à une fonction, effacez−le)
IS_CLASS	??
IS_OBJECT	??

Si vous recevez un argument d'un type, et que vous voulez l'utiliser avec un autre type, ou si vous voulez simplement forcer le type, vous pouvez utiliser l'une des fonctions de conversion suivantes :

```
convert_to_long(arg1);
convert_to_double(arg1);
convert_to_string(arg1);
convert_to_boolean_long(arg1);
/* Si la chaîne est "" ou "0" elle devient 0, 1 sinon */
convert_string_to_number(arg1);
/* Convertit une chaîne en LONG ou DOUBLE suivant la chaîne */
```

Ces fonctions convertissent sur place : elles ne retournent aucune valeur.

La valeur de l'argument est enregistrée dans une union. Les membres sont :

- IS_STRING: arg1−>value.str.val
- IS_LONG: arg1−>value.lval
- IS_DOUBLE: arg1−>value.dval

9.5.1.5 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 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/permanante 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 au crash ("segmentation fault"). Soyez alors extrêmement prudent, et libérez toute votre mémoire inutilisée.

Si vous compilez avec "−DDEBUG", PHP 3 affichera la liste de tous les appels à emalloc() et estrdup() mais jamais à efree() lorsque celui−ci intervient dans un script spécifié.

9.5.1.6 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)

`Gestion de la mémoire`

Soyez prudent. La valeur doit être placée dans une portion de mémoire créée avec malloc(), sinon le gestionnaire de mémoire essayera de libérer le pointeur plus tard. Ne passez aucune mémoire allouée statiquement à SET_VAR_STRING.

Les tables des symboles de PHP 3 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 grâces aux macros:

```
add_next_index_long(return_value,long_value);
add_next_index_double(return_value,double_value);
add_next_index_string(return_value,estrdup(string_value));
```

Bien sûr, si l'ajout n'est pas fait juste après l'initialisation, vous devrez d'abord rechercher le tableau :

```
pval *arr;
```

```
if (hash_find(active_symbol_table,"foo",sizeof("foo"),(void **)&arr)==FAILURE)
{ introuvable... }
else
{ utilisez arr->value.ht... }
```

Notez que hash_find reçoit un pointeur sur un pointeur sur pval, et pas un pointeur sur pval.

Toutes les fonctions d'accès aux hash retourne TRUE (SUCCES) ou FALSE (FAILURE), excepté hash_exists(), qui retourne un booléen.

9.5.1.7 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.

9.5.1.8 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 reél 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

9.5.1.9 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.

9.5.1.10 Utiliser la table des ressources persistantes.

PHP 3.0 dispose d'une lieu de stockage des ressources persistantes (i.e., les ressources qui doivent être conservées d'un hit à l'autre). Le premier module a 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'implémentez un module persistant. Typiquement, on utiliser le fait que la liste de ressources persistantes est une table de hash. Par exemple, dans les modules MySQL/mSQL, lors d'un appel à pconnect(), la fonction construit une chaîne avec l'hôte/utilisateur/mot_de_passe, et l'utilise pour enregistrer dans la table de hash. Au prochain appel, avec les mêmes hôte/utilisateur/mot_de_passe, la même clé sera générée, et la ressource associée sera retrouvée.

Jusqu'à ce que la documentation s'étoffe, jetez un oeil aux fichiers mysql.c ou msql.c pour voir comment implémentez vos accès aux ressources persistantes.

Une chose importante à noter : les ressources qui sont enregistrées dans la liste de ressource persistante ne DOIVENT PAS être allouée avec le gestionnaire de mémoire PHP, c'est−à−dire qu'elles ne doivent pas être créée avec emalloc(), estrdup(), etc. Au contraire, il faut utiliser les fonctions standard malloc(), strdup(), etc. La raison est for simple : à la fin de la requête, la mémoire sera supprimée par le gestionnaire. 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 placé dans la liste de ressources persistantes, il faut ajouter les destructeurs dans les deux listes de ressources, persistantes ou pas. Le destructeur de la liste de ressources non persistantes ne doit rien faire du tout, tandis que celui de la liste de ressources persistantes doit libérer proprement toutes les ressources acquises (mémoire, lien SQL...). Commep pour les ressources non persistantes vous DEVEZ ajouter un destructeur, même s'il ne fait rien. N'oubliez pas que emalloc() et compagnie ne doivent pas être utilisé en conjonction avec la liste de ressources persistantes, et donc, vous ne devez pas utiliser efree() non plus.

9.5.1.11 Ajouter des directives de configuration à l'exécution

De nombreuses caractéristiques de PHP 3 peuvent être configurée à l'éxécution. 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 aproprié à

cfg_get_string() ou cfg_get_long().
3. Ajoutez la directive, ses restrictions et un commentaire dans la structure php3_commands du fichier mod_php3.c. Notez la partie restrictions RSRC_CONF sont des directives qui ne peuvent être disponibles que dans le fichier de configuration Apache. Toutes les directives OR_OPTIONS peuvent être placées n'importe où, y compris dans un fichier `.htaccess` .
4. Soit dans php3take1handler(), soit dans php3flaghandler(), ajoutez l'entrée appropriée pour votre directive.
5. Dans la section de configuration, de _php3_info(), dans le fichier functions/info.c, vous devez ajouter votre configuration.
6. Finalement, vous devez utiliser votre configuration quelque part. Elle sera accessible par php3_ini.directive.

9.5.2 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 :

9.5.2.1 HashTable *function_table

La table de hash dans laquelle le fonction doit être recherchée.

9.5.2.2 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.

9.5.2.3 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.

9.5.2.4 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.

9.5.2.5 int param_count

Le nombre de paramètre passé à la fonction.

9.5.2.6 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.

9.5.3 Rapport d'erreurs

Pour signaler les erreurs d'une fonction interne, vous devez appelez 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 :

9.5.3.1 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.

9.5.3.2 E_WARNING

Les alertes sont affichées par défaut, mais n'interrompent pas l'éxécution du script. Elles indiquent un problème qui doit être intercepté par le script avant que l'appel. Par exemple, appeler `ereg` avec une regex invalide.

9.5.3.3 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.

9.5.3.4 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.

9.5.3.5 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.

9.5.3.6 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.

9.5.3.7 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.

9.5.3.8 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.

9.5.3.9 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.

9.5.3.10 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.

9.5.3.11 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.

9.6 Liste d'alias

Voici la liste des aliases. 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.

Alias	Fonction principale	Extension mère
_	`gettext`	Gettext

add	swfmovie_add	Ming (flash)
add	swfsprite_add	Ming (flash)
add_root	domxml_add_root	DOM XML
addaction	swfbutton_addaction	Ming (flash)
addcolor	swfdisplayitem_addcolor	Ming (flash)
addentry	swfgradient_addentry	Ming (flash)
addfill	swfshape_addfill	Ming (flash)
addshape	swfbutton_addshape	Ming (flash)
addstring	swftext_addstring	Ming (flash)
addstring	swftextfield_addstring	Ming (flash)
align	swftextfield_align	Ming (flash)
attributes	domxml_attributes	DOM XML
children	domxml_children	DOM XML
chop	rtrim	Base syntax
close	closedir	Base syntax
com_get	com_propget	COM
com_propset	com_propput	COM
com_set	com_propput	COM
cv_add	ccvs_add	CCVS
cv_auth	ccvs_auth	CCVS
cv_command	ccvs_command	CCVS
cv_count	ccvs_count	CCVS
cv_delete	ccvs_delete	CCVS
cv_done	ccvs_done	CCVS
cv_init	ccvs_init	CCVS
cv_lookup	ccvs_lookup	CCVS
cv_new	ccvs_new	CCVS
cv_report	ccvs_report	CCVS
cv_return	ccvs_return	CCVS
cv_reverse	ccvs_reverse	CCVS
cv_sale	ccvs_sale	CCVS
cv_status	ccvs_status	CCVS
cv_textvalue	ccvs_textvalue	CCVS
cv_void	ccvs_void	CCVS
die	exit	Miscellaneous functions
dir	getdir	Base syntax
diskfreespace	disk_free_space	Filesystem
domxml_getattr	domxml_get_attribute	DOM XML
domxml_setattr	domxml_set_attribute	DOM XML
doubleval	floatval	Base syntax
drawarc	swfshape_drawarc	Ming (flash)
drawcircle	swfshape_drawcircle	Ming (flash)
drawcubic	swfshape_drawcubic	Ming (flash)
drawcubicto	swfshape_drawcubicto	Ming (flash)
drawcurve	swfshape_drawcurve	Ming (flash)

drawcurveto	swfshape_drawcurveto	Ming (flash)
drawglyph	swfshape_drawglyph	Ming (flash)
drawline	swfshape_drawline	Ming (flash)
drawlineto	swfshape_drawlineto	Ming (flash)
dtd	domxml_intdtd	DOM XML
dumpmem	domxml_dumpmem	DOM XML
fbsql	fbsql_db_query	FrontBase
fputs	fwrite	Base syntax
get_attribute	domxml_get_attribute	DOM XML
getascent	swffont_getascent	Ming (flash)
getascent	swftext_getascent	Ming (flash)
getattr	domxml_get_attribute	DOM XML
getdescent	swffont_getdescent	Ming (flash)
getdescent	swftext_getdescent	Ming (flash)
getheight	swfbitmap_getheight	Ming (flash)
getleading	swffont_getleading	Ming (flash)
getleading	swftext_getleading	Ming (flash)
getshape1	swfmorph_getshape1	Ming (flash)
getshape2	swfmorph_getshape2	Ming (flash)
getwidth	swfbitmap_getwidth	Ming (flash)
getwidth	swffont_getwidth	Ming (flash)
getwidth	swftext_getwidth	Ming (flash)
gzputs	gzwrite	Zlib
i18n_convert	mb_convert_encoding	Multi−bytes Strings
i18n_discover_encoding	mb_detect_encoding	Multi−bytes Strings
i18n_http_input	mb_http_input	Multi−bytes Strings
i18n_http_output	mb_http_output	Multi−bytes Strings
i18n_internal_encoding	mb_internal_encoding	Multi−bytes Strings
i18n_ja_jp_hantozen	mb_convert_kana	Multi−bytes Strings
i18n_mime_header_decode	mb_decode_mimeheader	Multi−bytes Strings
i18n_mime_header_encode	mb_encode_mimeheader	Multi−bytes Strings
imap_create	imap_createmailbox	IMAP
imap_fetchtext	imap_body	IMAP
imap_getmailboxes	imap_list_full	IMAP
imap_getsubscribed	imap_lsub_full	IMAP
imap_header	imap_headerinfo	IMAP
imap_listmailbox	imap_list	IMAP
imap_listsubscribed	imap_lsub	IMAP
imap_rename	imap_renamemailbox	IMAP
imap_scan	imap_listscan	IMAP
imap_scanmailbox	imap_listscan	IMAP
ini_alter	ini_set	Base syntax
is_double	is_float	Base syntax
is_integer	is_int	Base syntax
is_long	is_int	Base syntax

is_real	is_float	Base syntax
is_writeable	is_writable	Base syntax
join	implode	Base syntax
labelframe	swfmovie_labelframe	Ming (flash)
labelframe	swfsprite_labelframe	Ming (flash)
last_child	domxml_last_child	DOM XML
lastchild	domxml_last_child	DOM XML
ldap_close	ldap_unbind	LDAP
magic_quotes_runtime	set_magic_quotes_runtime	Base syntax
mbstrcut	mb_strcut	Multi−bytes Strings
mbstrlen	mb_strlen	Multi−bytes Strings
mbstrpos	mb_strpos	Multi−bytes Strings
mbstrrpos	mb_strrpos	Multi−bytes Strings
mbsubstr	mb_substr	Multi−bytes Strings
ming_setcubicthreshold	ming_setcubicthreshold	Ming (flash)
ming_setscale	ming_setscale	Ming (flash)
move	swfdisplayitem_move	Ming (flash)
movepen	swfshape_movepen	Ming (flash)
movepento	swfshape_movepento	Ming (flash)
moveto	swfdisplayitem_moveto	Ming (flash)
moveto	swffill_moveto	Ming (flash)
moveto	swftext_moveto	Ming (flash)
msql	msql_db_query	mSQL
msql_createdb	msql_create_db	mSQL
msql_dbname	msql_result	mSQL
msql_dropdb	msql_drop_db	mSQL
msql_fieldflags	msql_field_flags	mSQL
msql_fieldlen	msql_field_len	mSQL
msql_fieldname	msql_field_name	mSQL
msql_fieldtable	msql_field_table	mSQL
msql_fieldtype	msql_field_type	mSQL
msql_freeresult	msql_free_result	mSQL
msql_listdbs	msql_list_dbs	mSQL
msql_listfields	msql_list_fields	mSQL
msql_listtables	msql_list_tables	mSQL
msql_numfields	msql_num_fields	mSQL
msql_numrows	msql_num_rows	mSQL
msql_regcase	sql_regcase	mSQL
msql_selectdb	msql_select_db	mSQL
msql_tablename	msql_result	mSQL
mssql_affected_rows	sybase_affected_rows	Sybase
mssql_affected_rows	sybase_affected_rows	Sybase
mssql_close	sybase_close	Sybase
mssql_close	sybase_close	Sybase
mssql_connect	sybase_connect	Sybase

mssql_connect	sybase_connect	Sybase
mssql_data_seek	sybase_data_seek	Sybase
mssql_data_seek	sybase_data_seek	Sybase
mssql_fetch_array	sybase_fetch_array	Sybase
mssql_fetch_array	sybase_fetch_array	Sybase
mssql_fetch_field	sybase_fetch_field	Sybase
mssql_fetch_field	sybase_fetch_field	Sybase
mssql_fetch_object	sybase_fetch_object	Sybase
mssql_fetch_object	sybase_fetch_object	Sybase
mssql_fetch_row	sybase_fetch_row	Sybase
mssql_fetch_row	sybase_fetch_row	Sybase
mssql_field_seek	sybase_field_seek	Sybase
mssql_field_seek	sybase_field_seek	Sybase
mssql_free_result	sybase_free_result	Sybase
mssql_free_result	sybase_free_result	Sybase
mssql_get_last_message	sybase_get_last_message	Sybase
mssql_get_last_message	sybase_get_last_message	Sybase
mssql_min_client_severity	sybase_min_client_severity	Sybase
mssql_min_error_severity	sybase_min_error_severity	Sybase
mssql_min_message_severity	sybase_min_message_severity	Sybase
mssql_min_server_severity	sybase_min_server_severity	Sybase
mssql_num_fields	sybase_num_fields	Sybase
mssql_num_fields	sybase_num_fields	Sybase
mssql_num_rows	sybase_num_rows	Sybase
mssql_num_rows	sybase_num_rows	Sybase
mssql_pconnect	sybase_pconnect	Sybase
mssql_pconnect	sybase_pconnect	Sybase
mssql_query	sybase_query	Sybase
mssql_query	sybase_query	Sybase
mssql_result	sybase_result	Sybase
mssql_result	sybase_result	Sybase
mssql_select_db	sybase_select_db	Sybase
mssql_select_db	sybase_select_db	Sybase
multcolor	swfdisplayitem_multcolor	Ming (flash)
mysql	mysql_db_query	MySQL
mysql_createdb	mysql_create_db	MySQL
mysql_db_name	mysql_result	MySQL
mysql_dbname	mysql_result	MySQL
mysql_dropdb	mysql_drop_db	MySQL
mysql_fieldflags	mysql_field_flags	MySQL
mysql_fieldlen	mysql_field_len	MySQL
mysql_fieldname	mysql_field_name	MySQL
mysql_fieldtable	mysql_field_table	MySQL
mysql_fieldtype	mysql_field_type	MySQL
mysql_freeresult	mysql_free_result	MySQL

mysql_listdbs	mysql_list_dbs	MySQL
mysql_listfields	mysql_list_fields	MySQL
mysql_listtables	mysql_list_tables	MySQL
mysql_numfields	mysql_num_fields	MySQL
mysql_numrows	mysql_num_rows	MySQL
mysql_selectdb	mysql_select_db	MySQL
mysql_tablename	mysql_result	MySQL
name	domxml_attrname	DOM XML
new_child	domxml_new_child	DOM XML
new_xmldoc	domxml_new_xmldoc	DOM XML
nextframe	swfmovie_nextframe	Ming (flash)
nextframe	swfsprite_nextframe	Ming (flash)
node	domxml_node	DOM XML
oci8append	ocicollappend	OCI8
oci8assign	ocicollassign	OCI8
oci8assignelem	ocicollassignelem	OCI8
oci8close	ocicloselob	OCI8
oci8free	ocifreecoll	OCI8
oci8free	ocifreedesc	OCI8
oci8getelem	ocicollgetelem	OCI8
oci8load	ociloadlob	OCI8
oci8max	ocicollmax	OCI8
oci8ocifreecursor	ocifreestatement	OCI8
oci8save	ocisavelob	OCI8
oci8savefile	ocisavelobfile	OCI8
oci8size	ocicollsize	OCI8
oci8trim	ocicolltrim	OCI8
oci8writetemporary	ociwritetemporarylob	OCI8
oci8writetofile	ociwritelobtofile	OCI8
odbc_do	odbc_exec	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

9.7 Mot réservés en PHP

Cette annexes est une liste d'identifiant 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 es variables prédéfinies. Ces listes ne sont pas complètes ou exhaustives.

9.7.1 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 contantes, 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.

- and .
- as .
- $argv .
- $argc .
- break .
- case .
- cfunction .
- class .
- continue .
- declare .
- default .
- do .
- die .

- `echo` .
- else .
- elseif .
- `empty` .
- enddeclare .
- endfor .
- endforeach .
- endif .
- endswitch .
- endwhile .
- eval .
- E_ALL .
- E_PARSE .
- E_ERROR .
- E_WARNING .
- `exit` .
- extends .
- FALSE .
- for .
- foreach .
- function .
- $HTTP_COOKIE_VARS .
- $HTTP_GET_VARS .
- $HTTP_POST_VARS .
- $HTTP_POST_FILES .
- $HTTP_ENV_VARS .
- $HTTP_SERVER_VARS .
- if .
- `include` .
- `include_once` .
- global .
- `list` .
- new .
- not .
- NULL .
- old_function .
- or .
- parent .
- PHP_OS .
- $PHP_SELF .
- PHP_VERSION .
- `print` .
- `require` .
- `require_once` .
- return .
- static .
- switch .
- stdClass .
- $this .
- TRUE .
- var .
- xor .
- `virtual` .
- while .

- __FILE__ .
- __LINE__ .
- __sleep .
- __wakeup .

9.7.2 Variables prédéfinies

9.7.2.1 V $_SERVER

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_SERVER_VARS .

$_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 accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_SERVER et $HTTP_SERVER_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

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 cour d'éxécution, 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 passées 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 éxécute 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_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`'.

9.7.2.2 V $_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, indépendemment 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.

9.7.2.3 H $_COOKIE

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait `$HTTP_COOKIE_VARS` .

Un tableau associatif des valeurs passées au script courant via les cookies HTTP. Elle est automatiquement 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 accessible 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.

9.7.2.4 H $_GET

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_GET_VARS .

Un tableau associatif des valeurs passées au script courant via le protocole HTTP et la méthode GET. Elle est automatiquement globale dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire `global $_GET;` pour y accéder, comme vous le faisiez avec $HTTP_GET_VARS .

$HTTP_GET_VARS contient les mêmes informations, mais n'est pas autoglobale.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_GET et $HTTP_GET_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

9.7.2.5 H $_POST

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_POST_VARS .

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 accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_POST et $HTTP_POST_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

9.7.2.6 V $_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 accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_FILES` et `$HTTP_POST_FILES` . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

9.7.2.7 V $_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 accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_REQUEST` . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

9.7.2.8 S $_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 accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_SESSION` et `$HTTP_SESSION_VARS` . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

9.7.2.9 V $GLOBALS

Note
$GLOBALS est disponible depuis PHP 3.0.0.

Un tableau associatif contenant les références sur toutes les variables globales actuellement définies dans le contexte d'exécution global du script. Les noms des variables sont les index du tableau.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire `global $GLOBALS;` pour y accéder.

9.7.2.10 L $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).

9.7.3 Classes prédéfinies

9.7.3.1 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'instantier `dir` .

`stdClass`

9.7.3.2 C 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`

9.7.3.3 C 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

9.7.3.4 C 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

9.8 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 p
aspell	`aspell_new`	`aspell_check` , `aspell_check_raw` , `aspell_suggest`	None
bzip2	`bzopen`	`bzerrno` , `bzerror` , `bzerrstr` , `bzflush` , `bzread` , `bzwrite`	`bzclose`
COM	`com_load`	`com_invoke` , `com_propget` , `com_get` , `com_propput` , `com_set` , `com_propput`	None
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_cl`

		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_jpeg , cpdf_place_inline_image , cpdf_add_annotation	
cpdf outline			
curl	curl_init	curl_init , curl_exec	curl_cl
dbm	dbmopen	dbmexists , dbmfetch , dbminsert , dbmreplace , dbmdelete , dbmfirstkey , dbmnextkey	dbmclos
dba	dba_popen	dba_delete , dba_exists , dba_fetch , dba_firstkey , dba_insert , dba_nextkey , dba_optimize , dba_replace , dba_sync	dba_clos
dba persistent	dba_open	dba_delete , dba_exists , dba_fetch , dba_firstkey , dba_insert , dba_nextkey , dba_optimize , dba_replace , dba_sync	None
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_c
dbx_link_object	dbx_connect	dbx_query	dbx_clos
dbx_result_object	dbx_query		None
domxml attribute			
domxml document			
domxml node			
xpath context			
xpath object			
fbsql database	fbsql_select_db		None
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_c

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
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_f
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_clos
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_size , ftp_mdtm , ftp_rename , ftp_delete , ftp_site	ftp_qui
gd	imagecreate , imagecreatefromgif , imagecreatefromjpeg , imagecreatefrompng , imagecreatefromwbmp , imagecreatefromstring , imagecreatetruecolor	imagearc , imagechar , imagecharup , imagecolorallocate , imagecolorat , imagecolorclosest , imagecolorexact , imagecolorresolve , imagegammacorrect , imagegammacorrect , imagecolorset , imagecolorsforindex , imagecolorstotal , imagecolortransparent , imagecopy , imagecopyresized , imagedashedline , imagefill , imagefilledpolygon , imagefilledrectangle , imagefilltoborder , imagegif , imagepng , imagejpeg , imagewbmp , imageinterlace , imageline , imagepolygon , imagepstext , imagerectangle , imagesetpixel , imagestring , imagestringup , imagesx , imagesy , imagettftext , imagefilledarc , imageellipse , imagefilledellipse , imagecolorclosestalpha , imagecolorexactalpha , imagecolorresolvealpha , imagecopymerge , imagecopymergegray , imagecopyresampled , imagetruecolortopalette , imagesetbrush	imagedes

		, imagesettile , imagesetthickness	
gd font	imageloadfont	imagechar , imagecharup , imagefontheight	None
gd PS encoding			
gd PS font	imagepsloadfont	imagepstext , imagepsslantfont , imagepsextendfont , imagepsencodefont , imagepsbbox	imageps
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_powm , 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
hyperwave document	hw_cp , hw_docbyanchor , hw_getremote , hw_getremotechildren	hw_children , hw_childrenobj , hw_getparents , hw_getparentsobj , hw_getchildcoll , hw_getchildcollobj , hw_getremote , hw_getsrcbydestobj , hw_getandlock , hw_gettext , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , hw_getanchorsobj , hw_inscoll , hw_pipedocument , hw_unlock	hw_dele
hyperwave link	hw_connect	hw_children , hw_childrenobj , hw_cp , hw_deleteobject , hw_docbyanchor , hw_docbyanchorobj , hw_errormsg , hw_edittext , hw_error , hw_getparents , hw_getparentsobj , hw_getchildcoll , hw_getchildcollobj , hw_getremote , hw_getremotechildren , hw_getsrcbydestobj , hw_getobject , hw_getandlock , hw_gettext , hw_getobjectbyquery , hw_getobjectbyqueryobj , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , hw_getanchorsobj , hw_mv , hw_incollections , hw_info , hw_inscoll , hw_insdoc , hw_insertdocument , hw_insertobject , hw_mapid , hw_modifyobject , hw_pipedocument , hw_unlock , hw_who , hw_getusername	hw_clos hw_free
hyperwave link persistent	hw_pconnect	hw_children , hw_childrenobj , hw_cp , hw_deleteobject , hw_docbyanchor ,	None

		hw_docbyanchorobj , hw_errormsg , hw_edittext , hw_error , hw_getparents , hw_getparentsobj , hw_getchildcoll , hw_getchildcollobj , hw_getremote , hw_getremotechildren , hw_getsrcbydestobj , hw_getobject , hw_getandlock , hw_gettext , hw_getobjectbyquery , hw_getobjectbyqueryobj , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , hw_getanchorsobj , hw_mv , hw_incollections , hw_info , hw_inscoll , hw_insdoc , hw_insertdocument , hw_insertobject , hw_mapid , hw_modifyobject , hw_pipedocument , hw_unlock , hw_who , hw_getusername	
icap	icap_open	icap_fetch_event , icap_list_events , icap_store_event , icap_snooze , icap_list_alarms , icap_delete_event	icap_cl
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_msgno , imap_search , imap_fetch_overview	imap_cl
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_

ingres persistent	ingres_pconnect	ingres_query , ingres_num_rows , ingres_num_fields , ingres_field_name , ingres_field_type , ingres_field_nullable , ingres_field_length , ingres_field_precision , ingres_field_scale , ingres_fetch_array , ingres_fetch_row , ingres_fetch_object , ingres_rollback , ingres_commit , ingres_autocommit	None
interbase blob			
interbase link	ibase_connect	ibase_query , ibase_prepare , ibase_trans	ibase_c
interbase link persistent	ibase_pconnect	ibase_query , ibase_prepare , ibase_trans	None
interbase query	ibase_prepare	ibase_execute	ibase_f
interbase result	ibase_query	ibase_fetch_row , ibase_fetch_object , ibase_field_info , ibase_num_fields	ibase_f
interbase transaction	ibase_trans	ibase_commit	ibase_r
java			
ldap link	ldap_connect , ldap_search	ldap_count_entries , ldap_first_attribute , ldap_first_entry , ldap_get_attributes , ldap_get_dn , ldap_get_entries , ldap_get_values , ldap_get_values_len , ldap_next_attribute , ldap_next_entry	ldap_cl
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_fr
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_cl

		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	
SWFAction			
SWFBitmap			
SWFButton			
SWFDisplayItem			
SWFFill			
SWFFont			
SWFGradient			
SWFMorph			
SWFMovie			
SWFShape			
SWFSprite			
SWFText			
SWFTextField			
mnogosearch agent			
mnogosearch result			
msql link	msql_connect	msql , msql_create_db , msql_createdb , msql_drop_db , msql_drop_db , msql_select_db , msql_select_db	msql_cl
msql link persistent	msql_pconnect	msql , msql_create_db , msql_createdb , msql_drop_db , msql_drop_db , msql_select_db , msql_select_db	None
msql query	msql_query	msql , msql_affected_rows , msql_data_seek , msql_dbname , msql_fetch_array , msql_fetch_field , msql_fetch_object , msql_fetch_row , msql_fieldname , msql_field_seek , msql_fieldtable , msql_fieldtype ,	msql_fr msql_fr

		msql_fieldflags , msql_fieldlen , msql_num_fields , msql_num_rows , msql_numfields , msql_numrows , msql_result	
mssql link	mssql_connect	mssql_query , mssql_select_db	mssql_c
mssql link persistent	mssql_pconnect	mssql_query , mssql_select_db	None
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_f
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_c
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
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_f

		mysql_tablename	
oci8 collection			
oci8 connection	ocilogon , ociplogon , ocinlogon	ocicommit , ociserverversion , ocinewcursor , ociparse , ocierror	ocilogo
oci8 descriptor			
oci8 server			
oci8 session			
oci8 statement	ocinewdescriptor	ocirollback , ocinewdescriptor , ocirowcount , ocidefinebyname , ocibindbyname , ociexecute , ocinumcols , ociresult , ocifetch , ocifetchinto , ocifetchstatement , ocicolumnisnull , ocicolumnname , ocicolumnsize , ocicolumntype , ocistatementtype , ocierror	ocifree
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_cl
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
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_fr
velocis link			
velocis result			
OpenSSL key	openssl_get_privatekey , openssl_get_publickey	openssl_sign , openssl_seal , openssl_open , openssl_verify	openssl
OpenSSL X.509	openssl_x509_read	openssl_x509_parse ,	openssl

		openssl_x509_checkpurpose	
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_clos
oracle link	ora_logon	ora_do , ora_error , ora_errorcode , ora_rollback , ora_commitoff , ora_commiton , ora_open , ora_commit	ora_log
oracle link persistent	ora_plogon	ora_do , ora_error , ora_errorcode , ora_rollback , ora_commitoff , ora_commiton , ora_open , ora_commit	None
pdf document	pdf_new	pdf_add_bookmark , pdf_add_launchlink , pdf_add_locallink , 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_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_clos pdf_del

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_clos
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_locl
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_clos
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
pgsql result	pg_exec	pg_fetch_array , pg_fetch_object , pg_fieldisnull , pg_fetch_row , pg_fieldname , pg_fieldnum , pg_fieldprtlen , pg_fieldsize , pg_fieldtype , pg_getlastoid , pg_numfields , pg_result , pg_numrows	pg_free
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
pspell config	pspell_config_create	pspell_new_config	None
Sablotron XSLT	xslt_create	xslt_closelog , xslt_openlog , xslt_run , xslt_set_sax_handler , xslt_errno , xslt_error , xslt_fetch_result , xslt_free	xslt_fr

shmop	shmop_open	shmop_read , shmop_write , shmop_size , shmop_delete	shmop_c
sockets file descriptor set	socket	accept_connect , bind , connect , listen , read , write	close
sockets i/o vector			
dir	dir	readdir , rewinddir	closedi
file	fopen	feof , fflush , fgetc , fgetcsv , fgets , fgetss , flock , fpassthru , fputs , fwrite , fread , fseek , ftell , fstat , ftruncate , set_file_buffer , rewind	fclose
pipe	popen	feof , fflush , fgetc , fgetcsv , fgets , fgetss , fpassthru , fputs , fwrite , fread	pclose
socket	fsockopen	fflush , fgetc , fgetcsv , fgets , fgetss , fpassthru , fputs , fwrite , fread	fclose
stream			
sybase−db link	sybase_connect	sybase_query , sybase_select_db	sybase_
sybase−db link persistent	sybase_pconnect	sybase_query , sybase_select_db	None
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_
sybase−ct link	sybase_connect	sybase_affected_rows , sybase_query , sybase_select_db	sybase_
sybase−ct link persistent	sybase_pconnect	sybase_affected_rows , sybase_query , sybase_select_db	None
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_
sysvsem	sem_get	sem_acquire	sem_rel

sysvshm	shm_attach	shm_remove , shm_put_var , shm_get_var , shm_remove_var	shm_deta
wddx	wddx_packet_start	wddx_add_vars	wddx_pa
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_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_pars
zlib	gzopen	gzeof , gzgetc , gzgets , gzgetss , gzpassthru , gzputs , gzread , gzrewind , gzseek , gztell , gzwrite	gzclose
ZZIP Directory			
ZZIP Entry			

9.9 List of Supported Protocols/Wrappers

The following is a list of the various URL style protocols that PHP has built−in for use with the filesystem functions such as `fopen` and `copy` . In addition to these wrappers, as of PHP 4.3, you can write your own wrappers using PHP script and `stream_register_wrapper` .

9.9.1 HTTP and HTTPS

PHP 3, PHP 4. `https://` since PHP 4.3

- `http://example.com`
- `http://user:password@example.com`
- `https://example.com`
- `https://user:password@example.com`

Allows read−only access to files/resources via HTTP 1.0, using the HTTP GET method. A `Host:` header is sent with the request to handle name−based virtual hosts. If you have configured a user_agent string using your ini file or the stream context, it will also be included in the request.

Redirects have been supported since PHP 4.0.5; if you are using an earlier version you will need to include trailing slashes in your URLs.

The stream allows access to the **body** of the resource; the headers are stored in the `$http_response_header` variable. Since PHP 4.3, the headers are available using

stream_get_meta_data .

HTTP connections are read−only; you cannot write data or copy files to an HTTP resource.

Note
HTTPS is supported starting from PHP 4.3, if you have compiled in support for OpenSSL.

9.9.2 FTP and FTPS

PHP 3, PHP 4. ftps:// since 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

Allows read access to existing files and creation of new files via FTP. If the server does not support passive mode ftp, the connection will fail.

You can open files for either reading or writing, but not both simultaneously. If the remote file already exists on the ftp server and you attempt to open it for writing, the connection will fail. If you need to update existing files over ftp, use ftp_connect .

ftps:// was introduced in PHP 4.3. It is the same as ftp:// , but attempts to negotiate a secure connection with the ftp server. If the server does not support SSL, then the connection falls back to regular unencrypted ftp.

Note
FTPS is supported starting from PHP 4.3, if you have compiled in support for OpenSSL.

9.9.3 PHP input/output streams

PHP 3.0.13 and up, php://output and php://input since PHP 4.3

- php://stdin
- php://stdout
- php://stderr
- php://output
- php://input

php://stdin , php://stdout and php://stderr allow access to the corresponding input or output stream of the PHP process.

php://output allows you to write to the output buffer mechanism in the same way as print and echo .

php://input allows you to read raw POST data. It is a less memory intensive alternative to $HTTP_RAW_POST_DATA and does not need any special php.ini directives.

php://stdin and php://input are read−only, whereas php://stdout , php://stderr and php://output are write−only.

9.9.4 Compression Streams

`zlib:` PHP 4.0.4 − PHP 4.2.3 (systems with fopencookie only)

`compress.zlib://` and `compress.bzip2://` PHP 4.3 and up

- `zlib:`
- `compress.zlib://`
- `compress.bzip2://`

`zlib:` works like `gzopen` , except that the stream can be used with `fread` and the other filesystem functions. This is deprecated as of PHP 4.3 due to ambiguities with filenames containing ':' characters; use `compress.zlib://` instead.

`compress.zlib://` and `compress.bzip2://` are equivalent to `gzopen` and `bzopen` respectively, and operate even on systems that do not support fopencookie.

9.10 Liste des token de l'analyseur

De nombreuses parties du langage PHP sont représentées en interne par des trucs (NDT : litéralement) 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 suppposé 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

9.11 A propos du manuel

9.11.1 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 mirroires. Pour un confort d'utilisation accrue, il est recommandé d'uitiliser un mirroir 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 plus part 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 faire plus de 2000 pages, et qu'il est constamment révisé.

Note
Si vous n'avez pas de programme capable de lire des fichiers PDF, téléchargez le Adobe Acrobat Reader .

Pour les propriétaires de PDA Palm, deux formats Palm document et iSilo sont optimisés pour cette plateforme. Vous pouvez emporter votre PDA avec vous dans vos déplacement quotidiens, et utiliser les programmes DOC et iSilo pour améliorer vos connaissances de PHP, ou bien profiter de ses références rapides.

Pour les plate−formes Windowsm, le format Windows HTML Help fourni 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.

Note
Un projet de Visual Basic pour Linux a débuté, et il incluera un créateur de format CHM Creator et Viewer for Linux. Voyez leur page SourceForge.net si ce projet vous interesse.

9.11.2 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.

9.11.3 Où trouver plus d'information sur PHP?

Ce manuel n'a pas pour objectif de fournir des présentations sur les pratiques de programmation. Si vous être 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.

9.11.4 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 .

9.11.5 Comment sont générées les documentations

Ce manuel est écrit en XML , en utilisant la DTD de DocBook XM , et DSSSL (Document Style and Semantics Specification Language) pour le formatage, et aussi, expérimentalement, XSLT (Extensible Stylesheet Language Transformations) pour les maintenance et les formatages.

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 Microsoft HTML Help Workshop pour générer le format Windows HTML Help format du manuel, et bien sur, PHP lui−même pour réaliser certaines conversions et formatages.

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 .

9.11.6 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 doc 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 interessé 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 : Portugais brésilien, Tchèque, Hollandais, Français, Allemand, Hongrois, Italien, Japonais, Coréen, Polonais et Espagnol.

Ils peuvent tous être téléchargés ici : http://www.php.net/docs.php .

10 Index

10.1 Index des fonctions

- **A**
- ♦ abs
  ♦ acos
  ♦ acosh
  ♦ addcslashes
  ♦ addslashes
  ♦ aggregate
  ♦ 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_count_values
  ♦ array_diff
  ♦ array_fill
  ♦ array_filter
  ♦ array_flip
  ♦ array_intersect
  ♦ 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
- bcdiv
- bcmod
- bcmul
- bcpow
- bcscale
- bcsqrt
- bcsub
- bin2hex
- 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
- com_addref
- 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_restore
- cpdf_rlineto
- cpdf_rmoveto
- cpdf_rotate
- cpdf_save
- cpdf_save_to_file
- cpdf_scale
- cpdf_set_char_spacing
- cpdf_set_creator
- cpdf_set_current_page
- 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_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_setopt
- curl_version
- current
- cybercash_base64_decode
- cybercash_base64_encode
- cybercash_decr
- cybercash_encr
- cybermut_creerformulairecm
- cybermut_creerreponsecm
- cybermut_testmac
- 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_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_rcrtexact

- dbplus_rcrtlike
- dbplus_resolve
- dbplus_restorepos
- dbplus_rkeys
- dbplus_ropen
- dbplus_rquery
- dbplus_rrename
- dbplus_rsecindex
- dbplus_runlink
- dbplus_rzap
- dbplus_savepos
- dbplus_setindex
- dbplus_setindexbynumber
- dbplus_sql
- dbplus_tcl
- dbplus_tremove
- dbplus_undo
- dbplus_undoprepare
- dbplus_unlockrel
- dbplus_unselect
- dbplus_update
- dbplus_xlockrel
- dbplus_xunlockrel
- dbx_close
- dbx_compare
- dbx_connect
- dbx_error
- dbx_escape_string
- dbx_query
- dbx_sort
- dcgettext
- deaggregate
- 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
- domdocument−>add_root [deprecated]
- domxml_attributes
- domxml_children
- domxml_dumpmem
- domxml_get_attribute
- domxml_new_child
- domxml_new_xmldoc
- domxml_root
- domxml_set_attribute
- 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
- exit
- exp
- explode
- expm1
- extension_loaded
- extract
- ezmlm_hash

F

- 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_db_query

- fbsql_db_status
- fbsql_drop_db
- fbsql_errno
- 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_insert_id
- fbsql_list_dbs
- fbsql_list_fields
- fbsql_list_tables
- fbsql_next_result
- fbsql_next_result
- fbsql_num_fields
- fbsql_num_rows
- fbsql_pconnect
- fbsql_query
- fbsql_read_blob
- fbsql_read_clob
- fbsql_result
- fbsql_rollback
- fbsql_select_db
- fbsql_set_lob_mode
- fbsql_start_db
- fbsql_stop_db
- fbsql_tablename
- fbsql_warnings
- fclose
- fdf_close
- fdf_create
- fdf_get_file
- fdf_get_status
- fdf_get_value
- fdf_next_field_name
- fdf_open
- 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_value
- feof
- fflush
- fgetc
- fgetcsv
- fgets
- fgetss
- file
- file_exists
- file_get_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
- fputs
- fread
- frenchtojd
- fribidi_log2vis
- fscanf
- fseek
- fsockopen
- fstat
- ftell
- ftp_cdup
- ftp_chdir
- 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_rawlist
- ftp_rename
- ftp_rmdir
- ftp_set_option
- ftp_site
- ftp_size
- ftp_systype
- ftruncate
- func_get_arg
- func_get_args
- func_num_args
- function_exists
- fwrite

G

- 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_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
- gethostbynamel

- 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
- hebrev
- hebrevc
- hexdec
- highlight_file
- highlight_string
- htmlentities
- htmlspecialchars
- hw_api−>checkin
- hw_api−>checkout
- hw_api−>children
- hw_api−>content
- hw_api−>copy
- hw_api−>dbstat
- hw_api−>dcstat
- hw_api−>dstanchors
- hw_api−>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−>setcommitedversion
- hw_api−>srcanchors
- hw_api−>srcsofdst
- hw_api−>unlock
- hw_api−>user
- hw_api−>userlist
- hw_api_attribute
- hw_api_attribute−>key
- hw_api_attribute−>langdepvalue
- hw_api_attribute−>value
- hw_api_attribute−>values
- hw_api_content
- hw_api_content−>mimetype
- hw_api_content−>read
- hw_api_error−>count
- hw_api_error−>reason
- hw_api_object
- hw_api_object−>assign
- hw_api_object−>attreditable
- hw_api_object−>count
- hw_api_object−>insert
- hw_api_object−>remove
- hw_api_object−>title
- hw_api_object−>value
- hw_api_reason−>description
- hw_api_reason−>type
- hw_array2objrec
- hw_children
- hw_childrenobj
- hw_close
- hw_connect
- hw_cp
- hw_deleteobject
- hw_docbyanchor
- hw_docbyanchorobj
- hw_document_attributes
- hw_document_bodytag
- hw_document_content
- hw_document_setcontent
- hw_document_size
- hw_edittext
- hw_error
- hw_errormsg
- hw_free_document
- hw_getanchors
- hw_getanchorsobj

- hw_getandlock
- hw_getchildcoll
- hw_getchildcollobj
- hw_getchilddoccoll
- hw_getchilddoccollobj
- hw_getobject
- hw_getobjectbyquery
- hw_getobjectbyquerycoll
- hw_getobjectbyquerycollobj
- hw_getobjectbyqueryobj
- hw_getparents
- hw_getparentsobj
- hw_getremote
- hw_getremotechildren
- hw_getsrcbydestobj
- hw_gettext
- hw_identify
- hw_incollections
- hw_info
- hw_inscoll
- hw_insdoc
- 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_unlock
- hw_username
- hw_who
- hwapi_hgcsp
- hypot

I

- ibase_close
- ibase_commit
- ibase_connect
- ibase_errmsg
- ibase_execute
- ibase_fetch_object
- ibase_fetch_row
- ibase_field_info
- ibase_free_query
- ibase_free_result
- ibase_num_fields
- ibase_pconnect
- ibase_prepare
- ibase_query
- ibase_rollback
- ibase_timefmt

- ibase_trans
- 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_free_slob
- 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_open_slob
- ifxus_read_slob
- ifxus_seek_slob
- ifxus_tell_slob
- ifxus_write_slob
- ignore_user_abort
- image2wbmp
- imagealphablending
- imagearc
- imagechar
- imagecharup
- imagecolorallocate
- imagecolorat
- imagecolorclosest
- imagecolorclosestalpha
- imagecolordeallocate
- imagecolorexact

- imagecolorexactalpha
- imagecolorresolve
- imagecolorresolvealpha
- imagecolorset
- imagecolorsforindex
- imagecolorstotal
- imagecolortransparent
- imagecopy
- imagecopymerge
- imagecopymergegray
- imagecopyresampled
- imagecopyresized
- imagecreate
- imagecreatefromgif
- imagecreatefromjpeg
- imagecreatefrompng
- imagecreatefromstring
- imagecreatefromwbmp
- imagecreatefromxbm
- imagecreatefromxpm
- imagecreatetruecolor
- imagedashedline
- imagedestroy
- imageellipse
- imagefill
- imagefilledarc
- imagefilledellipse
- imagefilledpolygon
- imagefilledrectangle
- imagefilltoborder
- imagefontheight
- imagefontwidth
- imagegammacorrect
- imagegif
- imageinterlace
- imagejpeg
- imageline
- imageloadfont
- imagepalettecopy
- imagepng
- imagepolygon
- imagepsbbox
- imagepscopyfont
- imagepsencodefont
- imagepsextendfont
- imagepsfreefont
- imagepsloadfont
- imagepsslantfont
- imagepstext
- imagerectangle
- imagesetbrush
- imagesetpixel
- imagesetthickness
- imagesettile

- imagestring
- imagestringup
- imagesx
- imagesy
- imagetruecolortopalette
- imagettfbbox
- imagettftext
- imagetypes
- imagewbmp
- imap_8bit
- imap_alerts
- imap_append
- imap_base64
- imap_binary
- imap_body
- 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_getmailboxes
- imap_getsubscribed
- imap_header
- imap_headerinfo
- imap_headers
- imap_last_error
- imap_listmailbox
- imap_listsubscribed
- 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_setflag_full
- imap_sort
- imap_status
- imap_subscribe
- 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
- iptcparse
- ircg_channel_mode
- ircg_disconnect
- 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_notice
- ircg_part

- ircg_pconnect
- ircg_register_format_messages
- ircg_set_current
- 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
- jdtofrench
- jdtogregorian
- jdtojewish
- jdtojulian
- jdtounix
- 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
- leak
- 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
- max
- 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_http_input
- mb_http_output
- mb_internal_encoding
- mb_language
- mb_output_handler
- mb_parse_str
- mb_preferred_mime_name
- mb_send_mail
- mb_strcut
- mb_strimwidth
- mb_strlen
- mb_strpos
- mb_strrpos
- mb_strwidth
- mb_substitute_character
- mb_substr
- mcal_append_event
- mcal_close
- mcal_create_calendar
- mcal_date_compare
- mcal_date_valid
- mcal_day_of_week
- mcal_day_of_year
- mcal_days_in_month
- mcal_delete_calendar
- mcal_delete_event
- mcal_event_add_attribute
- mcal_event_init

- mcal_event_set_alarm
- mcal_event_set_category
- mcal_event_set_class
- mcal_event_set_description
- mcal_event_set_end
- mcal_event_set_recur_daily
- mcal_event_set_recur_monthly_mday
- mcal_event_set_recur_monthly_wday
- mcal_event_set_recur_none
- mcal_event_set_recur_weekly
- mcal_event_set_recur_yearly
- mcal_event_set_start
- mcal_event_set_title
- mcal_expunge
- mcal_fetch_current_stream_event
- mcal_fetch_event
- mcal_is_leap_year
- mcal_list_alarms
- mcal_list_events
- mcal_next_recurrence
- mcal_open
- mcal_popen
- mcal_rename_calendar
- mcal_reopen
- mcal_snooze
- mcal_store_event
- mcal_time_valid
- mcal_week_of_year
- mcrypt_cbc
- mcrypt_cfb
- mcrypt_create_iv
- mcrypt_decrypt
- mcrypt_ecb
- mcrypt_enc_get_algorithms_name
- mcrypt_enc_get_block_size
- mcrypt_enc_get_iv_size
- mcrypt_enc_get_key_size
- mcrypt_enc_get_modes_name
- mcrypt_enc_get_supported_key_sizes
- mcrypt_enc_is_block_algorithm
- mcrypt_enc_is_block_algorithm_mode
- mcrypt_enc_is_block_mode
- mcrypt_enc_self_test
- mcrypt_encrypt
- mcrypt_generic
- mcrypt_generic_deinit
- mcrypt_generic_end
- mcrypt_generic_init
- 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
- 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_deleteresponse
- 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_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
- metaphone
- method_exists
- mhash
- mhash_count
- mhash_get_block_size
- mhash_get_hash_name
- mhash_keygen_s2k
- microtime
- mime_content_type
- min
- mkdir
- mktime
- 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
- msql
- msql_affected_rows
- msql_close
- msql_connect
- msql_create_db
- msql_createdb
- msql_data_seek
- msql_dbname
- msql_drop_db
- msql_dropdb
- msql_error
- msql_fetch_array
- msql_fetch_field
- msql_fetch_object
- msql_fetch_row
- msql_field_seek
- msql_fieldflags
- msql_fieldlen
- msql_fieldname
- msql_fieldtable
- msql_fieldtype
- msql_free_result
- msql_freeresult
- msql_list_dbs
- msql_list_fields
- msql_list_tables
- msql_listdbs
- msql_listfields
- msql_listtables
- msql_num_fields
- msql_num_rows
- msql_numfields
- msql_numrows
- msql_pconnect
- msql_query
- msql_regcase
- msql_result
- msql_select_db
- msql_selectdb
- msql_tablename

- mssql_close
- mssql_connect
- 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_free_result
- mssql_get_last_message
- mssql_min_error_severity
- mssql_min_message_severity
- mssql_num_fields
- mssql_num_rows
- mssql_pconnect
- mssql_query
- mssql_result
- 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

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_can_change_color
- ncurses_cbreak
- ncurses_clear
- ncurses_clrtobot
- ncurses_clrtoeol
- ncurses_color_set
- ncurses_curs_set
- ncurses_def_prog_mode
- ncurses_def_shell_mode
- ncurses_define_key
- 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_getmouse
- ncurses_halfdelay
- ncurses_has_colors
- ncurses_has_ic
- ncurses_has_il
- ncurses_has_key
- 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_killchar
- ncurses_longname
- ncurses_mouseinterval
- ncurses_mousemask
- 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_newwin
- ncurses_nl
- ncurses_nocbreak
- ncurses_noecho
- ncurses_nonl
- ncurses_noqiflush
- ncurses_noraw
- ncurses_putp

- ncurses_qiflush
- ncurses_raw
- ncurses_refresh
- ncurses_resetty
- ncurses_savetty
- ncurses_scr_dump
- ncurses_scr_init
- ncurses_scr_restore
- ncurses_scr_set
- ncurses_scrl
- 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_touch
- ncurses_standend
- ncurses_standout
- ncurses_start_color
- ncurses_termattrs
- ncurses_termname
- ncurses_timeout
- ncurses_typeahead
- ncurses_ungetch
- ncurses_ungetmouse
- ncurses_use_default_colors
- ncurses_use_env
- ncurses_use_extended_names
- ncurses_vidattr
- ncurses_vline
- ncurses_wrefresh
- next
- 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
- number_format

- **O**
- ♦ ob_clean
  ♦ ob_end_clean
  ♦ ob_end_flush
  ♦ ob_flush
  ♦ ob_get_contents
  ♦ ob_get_length
  ♦ ob_get_level
  ♦ ob_get_status
  ♦ ob_gzhandler
  ♦ ob_iconv_handler
  ♦ ob_implicit_flush
  ♦ ob_start
  ♦ ocibindbyname
  ♦ ocicancel
  ♦ 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
  ♦ ociresult
  ♦ ocirollback
  ♦ ocirowcount

- ocisavelob
- ocisavelobfile
- ociserverversion
- ocisetprefetch
- ocistatementtype
- ociwritelobtofile
- octdec
- odbc_autocommit
- odbc_binmode
- odbc_close
- odbc_close_all
- odbc_columnprivileges
- odbc_columns
- odbc_commit
- odbc_connect
- odbc_cursor
- odbc_do
- odbc_error
- odbc_errormsg
- odbc_exec
- odbc_execute
- odbc_fetch_into
- 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_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_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_seal
- openssl_sign
- openssl_verify
- openssl_x509_checkpurpose
- 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_weblink
- pdf_arc
- pdf_attach_file
- pdf_begin_page
- pdf_circle
- pdf_clip
- pdf_close
- pdf_close_image
- pdf_closepath
- pdf_closepath_fill_stroke
- pdf_closepath_stroke
- pdf_concat
- pdf_continue_text
- pdf_curveto
- pdf_delete
- pdf_end_page
- 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_parameter
- pdf_get_value
- pdf_lineto
- 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_png
- pdf_open_tiff
- pdf_place_image
- 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_leading
- pdf_set_parameter
- pdf_set_text_matrix
- pdf_set_text_pos
- pdf_set_text_rendering
- pdf_set_value
- pdf_set_word_spacing
- pdf_setdash
- pdf_setflat
- pdf_setfont
- pdf_setgray
- pdf_setgray_fill
- pdf_setgray_stroke
- pdf_setlinecap
- pdf_setlinejoin
- pdf_setlinewidth
- 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_client_encoding
- pg_close
- pg_connect
- pg_dbname
- pg_end_copy
- pg_fetch_array
- pg_fetch_object
- 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_host
- pg_lo_close
- pg_lo_create
- pg_lo_export
- pg_lo_import
- pg_lo_open
- pg_lo_read
- pg_lo_read_all
- pg_lo_unlink
- pg_lo_write
- pg_num_fields
- pg_num_rows
- pg_options
- pg_pconnect
- pg_port
- pg_put_line
- pg_set_client_encoding
- pg_trace
- pg_tty
- pg_untrace
- php_ini_scanned_files
- php_logo_guid
- php_sapi_name
- php_uname
- phpcredits
- phpinfo
- phpversion
- pi
- png2wbmp

- popen
- pos
- posix_ctermid
- 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_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_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
- printf
- proc_close
- proc_open
- 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
- rewind
- rewinddir
- rmdir
- round
- rsort
- rtrim

S

- sem_acquire
- sem_get
- sem_release
- 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_readonly
- 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_magic_quotes_runtime
- set_time_limit
- setcookie
- setlocale
- settype
- 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_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
- sqrt
- srand
- sscanf
- stat
- str_pad
- str_repeat
- str_replace
- str_rot13
- str_word_count
- strcasecmp
- strchr
- strcmp
- strcoll
- strcspn
- stream_get_meta_data
- stream_register_wrapper
- stream_set_blocking

- stream_set_timeout
- stream_set_write_buffer
- strftime
- strip_tags
- stripcslashes
- stripslashes
- stristr
- strlen
- strnatcasecmp
- strnatcmp
- strncasecmp
- strncmp
- strpos
- strrchr
- strrev
- 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
- swfdisplayitem
- swfdisplayitem−>addcolor

- swfdisplayitem−>move
- swfdisplayitem−>moveto
- swfdisplayitem−>multcolor
- swfdisplayitem−>remove
- swfdisplayitem−>rotate
- swfdisplayitem−>rotateto
- swfdisplayitem−>scale
- swfdisplayitem−>scaleto
- swfdisplayitem−>setdepth
- swfdisplayitem−>setname
- swfdisplayitem−>setratio
- swfdisplayitem−>skewx
- swfdisplayitem−>skewxto
- swfdisplayitem−>skewy
- swfdisplayitem−>skewyto
- swffill
- swffill−>moveto
- swffill−>rotateto
- swffill−>scaleto
- 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_fetch_array
- 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
- 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_clear_search_limits
- 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_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
- variant
- 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_free
- xml_parser_get_option
- xml_parser_set_option
- xml_set_character_data_handler
- xml_set_default_handler
- xml_set_element_handler
- xml_set_external_entity_ref_handler
- xml_set_notation_decl_handler
- xml_set_object
- xml_set_processing_instruction_handler
- xml_set_unparsed_entity_decl_handler
- xmldoc
- xmldocfile
- 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
- xmltree
- xpath_eval
- xpath_new_context
- xslt_create
- xslt_errno
- xslt_error
- xslt_free
- xslt_process
- xslt_set_sax_handler

Y

- yaz_addinfo
- yaz_ccl_conf
- yaz_ccl_parse
- yaz_close
- yaz_connect
- yaz_database
- yaz_element
- yaz_errno
- yaz_error
- yaz_hits
- yaz_itemorder
- yaz_present
- yaz_range
- yaz_record
- yaz_scan
- yaz_scan_result
- yaz_search
- 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

10.2 Index des concepts

- **A**
-
  - Accepte une connexion sur une socket
  - Activate keyboard input echo
  - Active l'approximation des translation d'objets.
  - Active l'option décidant si, lors de la déconnexion du client, le script doit poursuivre son exécution ou non.
  - Active la couche de contrôle des membres et méthodes
  - Active la sauvegarde des paires de remplacement
  - Active la validation automatique.
  - Active le debugger interne de php.
  - Active le mode auto−validation
  - Active le suivi d'une connexion postgresql
  - Active ou désactive l'affichage des données de debuggage.
  - Active ou désactive l'entrelacement.
  - Active ou désactive la validation automatique.
  - Active ou désactive le mode autocommit.
  - Active ou désactive le mode passif.
  - Active ou désactive les alertes frontbase
  - 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.
  - Add attributed string at current position
  - Add attributed string with specified length at current position
  - Add character at current position and advance cursor
  - Add data to a transaction
  - Add string at new position in window
  - Add string with specified length at current position
  - Addition de 2 nombres gmp
  - Additionne deux nombres de grande taille.
  - Affecte le gestionnaire par défaut.
  - Affecte les gestionnaires d'entité non déclaré.
  - Affecte les gestionnaires d'instructions exécutables.
  - Affecte les gestionnaires de caractère bruts.
  - Affecte les gestionnaires de début et de fin.
  - Affecte les gestionnaires de notation.
  - Affecte les options d'un analyseur xml.
  - Affecte un type à une variable.
  - Affecte une nouvelle date de modification à un fichier.
  - Affiche de nombreuses informations sur le php.
  - Affiche des informations lisibles pour une variable.
  - Affiche hw_document.
  - Affiche la partie du fichier située après le pointeur du fichier.
  - Affiche le contenu d'un fichier en une seule fois
  - Affiche le frame courant.

♦ Affiche le frame nommé.
♦ Affiche le résultat sous la forme d'une table html.
♦ Affiche ou affecte le paramètre "apache request notes".
♦ Affiche un fichier.
♦ Affiche un résultat sous forme de table html
♦ Affiche un texte à la position courante.
♦ Affiche un texte à une position donnée.
♦ Affiche un texte à une position.
♦ Affiche un texte dans un rectangle.
♦ Affiche un texte sur une nouvelle ligne.
♦ 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 dynamiquemetn 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−slashs dans une chaîne
♦ Ajoute des données d'introspection
♦ Ajoute des slashes dans une chaîne, à la mode du langage c
♦ Ajoute différentes limitations de recherche
♦ Ajoute du texte
♦ Ajoute le mot au dictionnaire personnel de la session courante
♦ Ajoute le mot au dictionnaire personnel.
♦ 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 enregistrement dans une base dbase.
♦ Ajoute un lien hypertexte dans la page courante
♦ Ajoute un lien sur un fichier dans la page courante
♦ Ajoute un lien sur une annotation dans la page courante
♦ Ajoute un nouvaeu domaine virtuel
♦ Ajoute un nouveau domaine virtuel
♦ Ajoute un nouveau fils
♦ Ajoute un nouveau vecteur dans le tableau
♦ Ajoute un nouvel utilisateur à un domaine virtuel
♦ Ajoute un objet à un sprite
♦ Ajoute un objet dans une animation.
♦ Ajoute un paramètre à une transaction
♦ Ajoute un remplissage plein à la forme.
♦ Ajoute un signet à la page courante
♦ Ajoute un signet à la page courante.
♦ 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 éxécutable dans la page courante

- Ajoute une annotation.
- Ajoute une autre racine
- Ajoute une chaîne dans une boîte aux lettres.
- Ajoute une contrainte sur une relation
- Ajoute une couleur à la liste du gradient.
- Ajoute une couleur à une transoformation.
- 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 note d'annotation dans la page courante
- Ajoute une valeur dans une structure de configuration d'utilisateur
- Ajouter des variables à un paquet wddx.
- Alias de `disk_free_space`
- Alias de `odbc_field_len`
- Alias de `rtrim`
- 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.
- Aligne sur le chemin courant.
- Allocate a color pair
- Alloue une couleur pour une image.
- 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 fichier en fonction d'un format
- Analyse un fichier et retourne une ressource représentant la structure du fichier
- Analyse une au format csv pour que mcve_getcell et consorts fonctionnent
- Analyse une chaîne à l'aide d'un format
- Analyse une chaîne d'adresse.
- Analyse une en−tête mail
- Analyse une requête http
- Analyse une requête sql.
- Analyse une requête xml est appelle les méthodes associées
- Analyse une requête.
- Analyse une structure xml
- Analyse une url et retourne ses composants.
- Analyse, exécute et lit une requête
- Annule (rend void) une transaction dans la queue d'attente
- Annule la fonction exécutée à chaque tick
- Annule les transactions en cours.
- Annule toutes les limitations de recherche
- Annule un bit
- Annule une transaction
- Annule une transaction
- Annule une transaction

- Annule une transaction
- Annule une transaction sesam
- Annule une transaction.
- Annule une transaction.
- Appele une fonction d'échappement dans les plugins de personnalité msession
- Appelle l'analyseur ccl
- Appelle une fonction utilisateur
- Appelle une fonction utilisateur avec les paramètres rassemblés en tableau
- Appelle une fonction windows 32
- Appelle une méthode d'un composant
- 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
- Arc cosinus
- Arc cosinus hyperbolique
- Arc sinus
- Arc sinus hyperbolique
- Arc tangeant hyperbolique
- Arc tangent
- Arc tangent de deux variables
- Arrête l'animation flash.
- Arrondi à l'entier inférieur
- Arrondi.
- Arrondit au nombre supérieur
- Assign terminal default colors to color id −1
- Assigne l'action du bouton
- Attache un fichier à la page courante
- Attend l'exécution d'une requête
- Attend la fin de l'exécution d'un processus fils
- Attend une connexion (listen) sur une socket
- Attend x microsecondes
- Attributs d'un objet.
- 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
- Authentification d'une connexion ftp
- Avance d'un frame.
- Avance le pointeur interne d'un tableau

B

- Balise de corps d'un document.
- Bientôt documenté....
- Bientôt documenté....
- Bientôt documenté....
- Bientôt documenté....
- Bientôt documenté....
- Bientôt documenté....
- Bientôt documenté....

- ♦ Bientôt documenté....
- ♦ Bientôt documenté....
- ♦ Bientôt documenté....
- ♦ Bientôt documenté....
- ♦ Bientôt documenté....
- ♦ Bientôt documenté....
- ♦ Bientôt documenté....
- ♦ Bientôt documenté....
- ♦ Bind to ldap directory

• C

- ♦ Calculate la somme de vérification crc32
- ♦ Calcule des statistiques sur une table
- ♦ Calcule l'intersection de tableaux
- ♦ Calcule la clé métaphone
- ♦ Calcule la clé soundex
- ♦ 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 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 un hash.
- ♦ Carré parfait gmp
- ♦ 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 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 racine
- ♦ Change la taille de la police.
- ♦ Change la valeur d'une option de configuration
- ♦ Change le "umask" courant.
- ♦ Change le codage vectoriel d'un caractère dans une police.
- ♦ Change le dossier courant.
- ♦ Change le groupe possesseur du 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.
- ♦ Charge les données ispell
- ♦ Charge un fichier ouvert sur un serveur ftp.
- ♦ Charge un fichier sur un serveur ftp.
- ♦ Charge un module dotnet
- ♦ Charge un nouveau dictionnaire

♦ Charge un nouveau dictionnaire
♦ Charge un nouveau dictionnaire [obsolète]
♦ Charge un nouveau dictionnaire avec un dictionnaire personnel
♦ Charge 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 insert− and delete−capabilities
♦ Check for line insert− and delete−capabilities
♦ Check for presence of a function key on terminal keyboard
♦ Check if terminal has colors
♦ Check if we can change terminals colors
♦ Check the status of an invoice
♦ Checks in an object
♦ Checks out an object
♦ Checks whether an attribute is editable
♦ Chemin du dossier courant.
♦ Cherche les notes par mots clés dans database_name
♦ Chiffrage indéchiffrable (hashing)
♦ Chiffre
♦ 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/déchriffre des données en mode cbc
♦ Choisi la durée de transition entre deux pages.
♦ Choisi la rotation.
♦ Choisi une fonction utilisateur comme gestionnaire d'erreurs
♦ Choisit l'encodage du client
♦ Choisit le fichier qui contient le dictionnaire personnel
♦ Choisit le fichier qui contient les paires de remplacement.
♦ Choisit le mode de remplissage par texture repétée.
♦ 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 un niveau de gris comme couleur de dessin.
♦ Choisit une couleur rgb comme couleur de dessin et de remplissage.
♦ Choisit une couleur rgb comme couleur de dessin.
♦ Choisit une couleur rgb comme couleur de remplissage.
♦ Classe com
♦ Classe dossier
♦ Classe variant
♦ Clear screen
♦ Clear screen from current position to bottom
♦ Clear screen from current position to end of line
♦ Clears soft labels from screen
♦ Clones object
♦ Clos un paquet wddx.
♦ Close active page
♦ Close an open printer connection

- Close document
- Colorisation de la syntaxe d'un fichier
- Colorisation de la syntaxe d'un fichier
- Combine plusieurs tableaux ensembles, récursivement
- Commence l'analyse d'un fichier xml.
- Commence la définition d'un bouton.
- Commence la description d'une liste d'action pour la frame courante.
- Commence un nouveau chemin
- Commence une forme complexe.
- Commence une nouvelle page.
- Commencer un nouveau paquet wddx avec une structure
- Commits version other than last version
- 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'occurences de segments dans une chaîne
- Compte de population
- Compte le nombre d'éléments d'un tableau
- Compte le nombre d'entrées après une recherche
- 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
- Compter le nombre d'enregistrements dans une base dbase.
- Concatène une matrice avec ctm
- Configure l'analyseur ccl
- Configure la bufferisation de fichier pour un flôt
- 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 durée d'expiration d'un flôt
- Configure le mode bloquant d'un flôt

- Configure le temps maximum d'une transaction (par transaction)
- Configure les critères de tri
- Configure les fonctions de stockage de sessions
- 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 the printer connection
- Configure une fonction de callback pour refaire des liaisons, lors de recherche de référants.
- Connecte à un serveur irc
- Connection persistante à un serveur oracle.
- Connexion à un serveur
- Connexion à une source
- Considère deux mots accolés comme un composé.
- Construit un 'struct iovec'
- Contenu d'un document.
- Contrôle la situation, l'aparance et la zone active du bouton courant.
- Control screen background
- Control use of environment information about terminal size
- Control use of extended names in terminfo descriptions
- 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 en utf8
- Convertit entre les différents "kana"
- Convertit l'encodage de variables
- 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 tous les caractères éligibles en entités html
- Convertit un calendrier en nombre de jours julien
- 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 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 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 objets.
- 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 palette d'une image à l'autre
- Copie les messages spécifiés dans une boîte aux lettres.
- Copie un fichier.
- Copie une partie d'une image
- Copie, redimensionne, rééchantillonne une image
- Copies physically
- Copies soft label keys to screen
- Copies soft label keys to virtual screen
- Cosinus
- Cosinus hyperbolic
- Coupe une chaîne en segments
- Coupe une chaîne en segments
- Coupe une partie de chaîne
- Création d'un analyseur xml.
- Crée et initialise une structure mcve_conn
- Crée ou ouvre un bloc de mémoire partagée
- Crée ou ouvre un segment de mémoire partagée.
- Crée un arbre d'objet php, à partir d'un document xml.
- Crée un blob
- Crée un clob
- Crée un document xml vide
- Crée un dossier.
- Crée un dossier.
- 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 symbolique.
- Crée un lien.
- 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 document fdf.
- Crée un nouveau document.
- 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.
- 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 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 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 msql.
- Crée une base de données msql.
- 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 form_name
- Crée une note à partir du formulaire form_name
- Crée une nouvelle action.

- Crée une nouvelle boîte aux lettres.
- Crée une nouvelle forme.
- Crée une nouvelle image jpeg à partir d'un fichier ou d'une url
- Crée une nouvelle image png à partir d'un fichier ou d'une url
- Crée une nouvelle image à palette.
- Crée une nouvelle image à partir d'un fichier ou d'une url.
- Crée une nouvelle image en vraies couleurs
- 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 variable php à partir d'une valeur linéarisée
- Create a new brush
- Create a new device context
- Create a new font
- Create a new pen
- Create a new window
- Create a new, blank transaction
- Create new instance of class hw_api_content
- Creates a link to an object
- Creates a new instance of class hw_api_object
- Creates instance of class hw_api_attribute

D

- Décharge le module de chiffrement
- Déchiffre
- Déchiffre un message s/mime
- Déchiffre un texte
- Déclenche une erreur utilisateur
- 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.
- Déconnecte d'un serveur ldap
- Déconnexion 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.
- Définit la couleur transparente.
- 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
- 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 l'origine de l'objet swffill
- Déplace le point courant.
- Déplace le pointeur de lecture
- Déplace le pointeur de résultat
- Déplace le pointeur de résultat
- Déplace le pointeur interne de fichier
- Déplace le pointeur interne de ligne.
- Déplace le pointeur interne de lignes.
- Déplace le pointeur interne de résultat
- Déplace le pointeur interne de résultat.
- Déplace le pointeur interne vers le résultat suivant
- Déplace le pointeur interne vers le résultat suivant
- Déplace le pointeur interne.
- Déplace le stylo de texte
- Déplace le stylo relativement.
- Déplace le stylo.
- Déplace les messages spécifiés dans une boîte aux lettres.
- Déplace un curseur à défilement
- Déplace un fichier téléchargé.
- Déplace un objet en coordonnées globales.
- Déplace un objet en coordonnées relatives.
- Déplace un objet.
- Désalloue une couleur pour une image
- Détecte l'encodage mime
- Détecte le type d'encodage d'un caractère http
- Détecte un encodage
- Détermine le meilleur encodoage pour un fichier
- Détermine le nombre de décimales par défaut
- Détermine le rendu du texte.

- Détermine le rendu du texte.
- Détermine si l'argument peut être appelé comme fonction
- 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 variable est affectée.
- Détermine si une variable est affectée.
- Détermine si une variable est de type double.
- 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 integer.
- Détermine si une variable est de type object.
- Détermine si une variable est de type real.
- Détermine si une variable est de type string.
- Détermine si une variable est un tableau 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 specifié par iovec_id
- Détruit les données du buffer de sortie, et éteint la bufferisation de sortie
- Détruit toutes les variables de session
- Détruit un analyseur créé par mailparse_msg_crea()
- Détruit un analyseur xml.
- Détruit un analyseur xslt
- Détruit un bloc de mémoire partagée
- Détruit un document.
- Détruit un serveur xmlrpc
- Détruit une image.
- Détruit une relation
- Détruit une session
- Détruit une session mnogosearch
- Détruit une variable
- Déverrouille un objet.
- Déverrouille une session
- Define a keycode
- Define default colors for color 0
- Delay output on terminal using padding characters
- Delete a brush
- Delete a device context
- Delete a font
- Delete a ncurses window
- Delete a pen
- Delete a transaction
- Delete an object
- Delete character at current position, move rest of line left
- Delete line at current position, move rest of screen up
- Deletes the printer's spool file
- Demande d'informations d'arbre sur une entité du réseau.
- Desaggrège un objet
- Dessine le long du chemin.
- Dessine un arc de cercle.
- Dessine un arc.

- Dessine un caractère horizontalement.
- Dessine un caractère verticalement.
- Dessine un cercle.
- Dessine un cercle.
- Dessine un pixel.
- Dessine un polygone rempli.
- Dessine un polygone.
- Dessine un rectangle rempli.
- Dessine un rectangle.
- Dessine un rectangle.
- 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 bézier cubique.
- Dessine une courbe de bézier quadratique entre deux points.
- Dessine une courbe relativement.
- Dessine une courbe.
- Dessine une courbe.
- Dessine une courbe.
- Dessine une ellipse
- Dessine une ellipse partielle et la remplit
- Dessine une ellipse partielle.
- Dessine une ellipse pleine
- Dessine une ligne le long du chemin.
- Dessine une ligne pointillée.
- Dessine une ligne relativement.
- Dessine une ligne, relativement.
- Dessine une ligne.
- Dessine une ligne.
- Dessine une ligne.
- Dessine une ligne.
- Dessine une ligne.
- 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
- Do not flush on signal characters
- Do not translate newline and carriage return / line feed
- Draw a bmp
- Draw a border around the screen using attributed characters
- Draw a chord
- Draw a horizontal line at current position using an attributed character and max. n characters long
- Draw a line
- Draw a pie
- Draw a rectangle
- Draw a rectangle with rounded corners
- Draw a vertical line at current position using an attributed character and max. n characters long
- Draw an ellipse

- Draw text
- Dump screen content to file
- Dumpe les informations d'une variable.
- Duplique un objet blob.
- Détecte le type mime d'un fichier

- **E**
-
  - Echappe les méta−caractères
  - Echappe les méta−caractères shell.
  - Echappe une chaîne de caractères pour qu'elle soit utilisée en ligne de commande.
  - Echappement des caractères spéciaux des expressions régulières.
  - Eclatement d'une chaîne par expression régulière.
  - Ecrire dans un bloc de mémoire partagée
  - Ecrit dans l'historique
  - Ecrit dans un fichier compressé, en gzip
  - Ecrit dans un fichier.
  - 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 la valeur courante de l'option quick_print de la librairie ucd.
  - Ecrit le document xml interne dans une chaîne
  - Ecrit les données de session et ferme la session
  - Ecrit toutes les modifications apportées à une relation
  - Ecrit un document pdf dans un fichier.
  - Ecrit un fichier sur un serveur ftp, et le lit depuis un fichier (non bloquant)
  - Ecrit un objet de grande taille
  - Ecrit une chaîne dans un fichier compressé
  - Ecrit une chaîne dans un objet slob.
  - Ecriture binaire dans un fichier bzip2
  - Ecriture du fichier en mode binaire.
  - Efface des objets.
  - Efface et remplace une portion de tableau
  - Efface l'historique
  - Efface la dernière exception java
  - Efface le buffer de sortie
  - Efface le cache de `stat`
  - Efface le résultat de la mémoire.
  - 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 domaine virtuel
  - Efface un domaine virtuel
  - Efface un dossier.
  - Efface un dossier.
  - Efface un enregistrement dans une base dbase.
  - Efface un fichier
  - Efface un fichier sur un serveur ftp.
  - Efface un objet de grande taille
  - Efface un objet pdf

- Efface un utilisateur d'un domaine virtuel
- 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 entrée dans un dossier
- Efface une entrée.
- 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 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 sous−requête apache
- 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 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
- Enable or disable a keycode
- Enclenche la bufferisation de sortie
- Encode des entités html
- Encode les données de session
- Encode une chaîne en mime base64.
- Encode une chaîne en url, selon la rfc1738.
- Encode une chaîne en url.
- Encode une chaîne pour une en−tête mime
- Encore non implémentée.
- Enlève la marque d'effacement d'un message.
- Enlève un objet.
- Enregistre l'environnement courant.
- Enregistre un nouveau format d'affichage des messages irc
- Enregistre un nouvel événement dans un calendrier mcal.
- Enregistre une enveloppe url, implémenté comme classe php
- 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 win32 dans php
- Enregistre une paire de remplacement pour un mot
- Enregistre une variable dans une session
- Enregistrer plusieurs valeurs dans un paquet wddx
- Enregistrer une valeur dans un paquet wddx
- Enregsitre 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 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 d'erreur quelque part
- 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 override à mcve
- Envoie une commande preauthorization à mcve
- Envoie une en−tête http.
- 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 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.
- Envoie une transaction brute à payflow pro
- Envoie votre animation au navigateur.
- Envoye une requête à un serveur cyrus imap
- Erase terminal screen
- Establit la connexion avec le système mcve
- Et logique
- Etablit une connexion à un serveur oracle.
- Etablit une connexion persistante.
- Eteind une socket en lecture et/ou écriture
- Eteint l'alarme d'un événement.
- Eteint la librairie payflow pro
- Etend ou condense une police de caractères
- Etire un objet en coordonnées globales.
- Etire un objet relativement.
- Evalue une chaîne comme un script php.
- Evalue une expression xpath
- Exécute du code tcl coté serveur
- Exécute immédiatement une requête sql

- Exécute l'appel système select() un tableau de socket avec une durée d'expiration
- Exécute le programmee indiqué dans l'espace courant de processus
- Exécute un programme externe et affiche le résultat brut.
- Exécute un programme externe et affiche le résultat.
- Exécute un programme externe.
- Exécute un programme sur le serveur ftp
- Exécute une commande analysée sur un pointeur oracle.
- Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties
- Exécute une commande via le shell et retourne le résultat sous forme de chaîne.
- Exécute une commande.
- Exécute une fonction fichier de langage c
- Exécute une fonction sur chacun des membres d'un tableau.
- Exécute une requête aql
- Exécute une requête msql.
- Exécute une requête préparée
- Exécute une requête préparée.
- Exécute une requête sesam
- Exécute une requête sql
- Exécute une requête sql
- Exécute une requête sql déjà préparée.
- Exécute une requête sql préparée.
- Exécute une requête sql sans mobiliser les résultats
- Exécute une requête sur un serveur frontbase
- Exécute une requête sur une base interbase
- Exécute une session curl
- Exponentielle
- Exporte un objet de grande taille vers un fichier
- Expression régulière globale.
- 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 une clé publique d'un certificat
- Extrait une portion de tableau

F

- Factorielle gmp
- Fait du processus courant un chef de session.
- Fait tourner une forme relativement.
- Fait une copie d'une police chargée pour modification ultérieure.
- Fait une liste détaillée de fichiers dans un dossier.
- Ferme et clos le chemin.
- Ferme l'accès au fichier
- Ferme la connexion à l'historique système.
- Ferme la connexion à muscat
- 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 mysql.
- Ferme le chemin courant
- Ferme le chemin et dessine le long du chemin.
- Ferme le chemin.
- Ferme le fichier courant shockwave flash.
- Ferme le fichier et dessine une ligne le long du chemin.
- Ferme le pointeur sur le dossier.
- Ferme toutes les connexions odbc
- Ferme un bloc de mémoire partagée
- Ferme un document fdf.
- Ferme un document pdf.
- Ferme un dossier d'archive
- Ferme un fichier bzip2
- Ferme un fichier compressé
- Ferme un fichier pdf.
- Ferme un fichier.
- Ferme un objet de grande taille.
- 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 dbase.
- Ferme une base de données dbm.
- Ferme une base.
- 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 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 sybase.
- Ferme une connexion yaz
- Ferme une image
- Ferme une relation
- Ferme une session curl
- Ferme une socket
- Filtre les éléments d'un tableau
- Finalise et envoie une transaction
- Find out how many transactions of a given type are stored in the system
- Fixe d'offset d'un champs.
- 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 champs.
♦ Fixe l'echelle horizontale du texte.
♦ Fixe l'espacement des caractères.
♦ Fixe l'espacement des caractères.
♦ Fixe l'espacement des mots.
♦ Fixe l'espacement des mots.
♦ Fixe l'identifiant de group de processus.
♦ Fixe l'offset du pointeur de champs.
♦ Fixe l'uid effective du processus courant.
♦ Fixe la bufferisation de fichier
♦ 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 fin de la structure globale.
♦ Fixe la description de la structure globale.
♦ Fixe la distance entre deux lignes.
♦ Fixe la durée de vie de la socket
♦ 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 position du texte.
♦ Fixe la récurrence annuelle.
♦ Fixe la récurrence hebdomadaire.
♦ Fixe la récurrence mensuelle.
♦ Fixe la récurrence quotidienne.
♦ Fixe la récurrence.
♦ Fixe la sévérité minimale du client
♦ Fixe la sévérité minimale du client pour le serveur
♦ Fixe la sévérité minimale du client pour les erreurs
♦ Fixe la sévérité minimale du client pour les messages
♦ Fixe la valeur d'un champs.
♦ Fixe la valeur d'une variable d'environnement.
♦ Fixe la valeur de la clé /f.
♦ 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.
♦ Fixe le niveau de sévérité des messages d'erreurs.
♦ 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 relativement.
♦ Fixe le point courant.
♦ 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 dates de début et de fin de la structure globale.
♦ Fixe les limites d'un document pdf.
♦ Fixe les mot clés d'un document pdf.
♦ Flash terminal screen (visual bell)
♦ Flush keyboard input buffer
♦ Flush on signal characters
♦ Fonction de callback pour la compression automatique des buffers
♦ Fonction de traitement des affichages web
♦ Fonction générique pour les images
♦ Fonctionnement des expressions régulières.
♦ Force l'écriture de toutes les données compressées
♦ Forks the currently running process
♦ Formate un nombre pour l'affichage.
♦ Formate une date/heure gmt/cut en fonction des paramétrages locaux.
♦ Formate une date/heure gmt/cut.
♦ Formate une date/heure locale
♦ Formate une date/heure locale avec les options locales.
♦ Fources output when ncurses_slk_noutrefresh is performed
♦ Free result memory
♦ Fuite de mémoire.

• **G**

♦ Générateur de congruence combinée linéaire
♦ Génère le code xml d'une requête de méthode
♦ Génère le code xml pour une valeur php
♦ Génère un accusé de réception de confirmation de paiement
♦ Génère un formulaire html de paiement
♦ Génère un identifiant unique.
♦ Génère un message d'erreur utilisateur
♦ Génère un message dans l'historique système.
♦ Génère une clé
♦ Génère une meilleure valeur aléatoire.
♦ Génère une valeur aléatoire.
♦ Gestion des colonnes de type long.
♦ Gestionnaire de sortie pour maitriser le jeu de caractères de sortie
♦ Get character and attribute at current position
♦ Get error string for given errorcode or last error
♦ Get logical font height
♦ Get text return value for previous function call

• **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.
♦ 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.
- Inactive la validation automatique.
- Inactive le debugger interne de php.
- Inactive le remplissage.
- Inactive un utilisateur mcve actif
- Incline (abscisses)
- Incline (ordonnées)
- Incline suivant les x relativement.
- Incline suivant les x.
- Incline suivant les y relativement.
- Incline suivant les y.
- Incline une police de caractères
- Incrémente le compteur de références
- Incrémente une valeur dans une session
- Indique de quoi est capable le navigateur client.
- Indique la taille des données à lire dans une colonne de grande taille
- Indique le nombre de lignes qui doivent être pré−lues
- Indique si la fin d'un fichier compressé est atteinte
- Indique si le client a abandonné la connexion.
- 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 les entêtes 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 `null`
- Indique si une variable est un scalaire
- Indique un fichier est autorisé en lecture
- Informations à propos d'une connexion.
- Inherit screen from file dump
- Initialise la librairie payflow pro
- Initialise la structure globale d'un flot.
- Initialise le générateur de nombres aléatoires
- Initialise tous les buffers nécessaires
- Initialise toutes les constantes liées au syslog

- Initialise un nouveau pointeur vide de lob/file
- Initialise une meilleure valeur aléatoire
- Initialise une session
- Initialise une session curl
- Initialise une session en mode lecture
- Initialise une structure pour stocker des données utilisateurs
- Initialize ccvs for use
- Initialize ncurses
- Initialize screen from file dump
- Initializes soft label key functions
- Insère des retours à la ligne html à chaque nouvelle ligne
- Insère ou modifie une variable de la mémoire partagée.
- Insère un document dans un groupe.
- Insère un document.
- Insère un groupe.
- Insère un object record.
- Insère une entrée.
- Insère une valeur.
- Insert a line, move rest of screen down
- Insert character moving rest of line including character at current position
- Insert lines before current line scrolling down (negative numbers delete and scroll up)
- Insert string at current position, moving rest of line right
- Inserts a new object
- Inserts a new object of type anchor
- Inserts a new object of type collection
- Inserts a new object of type document
- Inserts new attribute
- Installe un gestionnaire de signaux
- Interroge l'autre extrémité de la communication
- Interroge la socket locale
- 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
- Let the terminal beep
- Libère la mémoire
- Libère la mémoire allouée pour ispell
- Libère la mémoire associée à la connectivité ip/ssl

♦ Libère la mémoire occupée par une police postscript type 1.
♦ Libère la mémoire réservée par une requête préparée.
♦ Libère la mémoire.
♦ Libère le résultat de la mémoire
♦ Libère le résultat de la mémoire.
♦ Libère le résultat de la mémoire.
♦ Libère les ressouces
♦ Libère les ressources
♦ 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 utilisées par un résultat
♦ Libère tous les verrous posés par le client
♦ Libère toutes les ressources occupées par un pointeur.
♦ Libère toutes les ressources occupées par une commande.
♦ Libère un résultat de la mémoire.
♦ Libère un résultat mnogosearch
♦ Libère un résultat.
♦ Libère un sémaphore.
♦ Libère un segment de mémoire partagée.
♦ Libère un verrou en écriture posé sur une relation
♦ Lie un nom à une socket
♦ Lie une variable php à un paramètre 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
♦ Liste les bases de données disponibles sur le serveur mysql.
♦ Liste les bases de données msql sur un serveur.
♦ Liste les bases de données msql sur un serveur.
♦ Liste les boîtes aux lettres souscrites.
♦ Liste les boîtes aux lettres, et retourne le détail pour chacune.
♦ Liste les boîtes aux lettres.
♦ 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 fonctions d'une extension
♦ Liste les gestionnaires dba 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 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 des données d'une socket
- Lit des données dans un fd, en utilisant le tableau scatter−gather définit par iovec_id
- Lit des données dans un résultat
- Lit des informations a propos de la dernière requête
- Lit des informations concernant un transfert particulier
- 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 options passés dans la ligne de commande
- Lit différenctes options pour la connexion ftp courante
- 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
- Lit et/ou modifie le module de session courant
- Lit et/ou modifie le nom de la session
- Lit l'échelle d'un champs
- Lit l'adresse de vérification (address verification) dans le statut retourné
- 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 système unique d'une transaction
- Lit la clé suivante.
- Lit la clé suivante.
- Lit la configuration du cookie de session
- 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, et y recherche une chaîne.
♦ Lit la longueur d'un champs.
♦ Lit la longueur d'un champs.
♦ Lit la méthode de compression utilisée sur un dossier d'archive
♦ Lit la position courante du pointeur de lecture
♦ Lit la première clé.
♦ Lit la première clé.
♦ Lit la première ligne d'une relation
♦ Lit la prochaine entrée
♦ Lit la prochaine entrée dans une archive zip
♦ Lit la référence suivante
♦ Lit la réponse de l'api muscat
♦ Lit la structure d'un message.
♦ Lit la taille compressée d'un dossier d'archive
♦ Lit la taille décompressée d'un dossier d'archive
♦ Lit la taille d'une colonne
♦ Lit la taille de chaque colonne d'un résultat
♦ 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 champs.
♦ Lit la valeur d'un paramètre pdflib chaîne
♦ Lit la valeur d'un paramètre pdflib numérique
♦ 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 option de configuration.
♦ Lit la valeur de la clé /f.
♦ Lit la valeur de la clé /status.
♦ Lit la version courante du moteur zend.
♦ Lit la version de lotus notes
♦ Lit la version des api mnogosearch.
♦ 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 d'un fichier dans un dossier
♦ Lit le contenu d'une colonne
♦ 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 vpopmail
♦ Lit le dn d'une entrée
♦ Lit le fichier et renvoie le résultat dans un tableau.
♦ Lit le jeu de caractères courant
♦ Lit le message d'erreur de l'analyseur xml.
♦ Lit le message retourné par mcve ou par l'institution de traitement
♦ Lit le nom d'un élément de code source
♦ Lit le nom d'un champs
♦ Lit le nom d'un champs.
♦ Lit le nom d'un champs.
♦ Lit le nom d'un champs.
♦ Lit le nom d'une colonne dans une réponse au format csv
♦ 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 champs

- Lit le nom de la table d'un champs
- Lit le nom de la table qui contient le champs spécifié.
- Lit le nom du champs 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 immédiate
- Lit le nombre de lignes dans un résultat
- Lit le nombre total de documents dans les bases.
- 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 payflow pro
- Lit le statut courant d'une base de données
- Lit le statut de cvc2/cvv2/cid
- Lit le statut du buffer de sortie
- Lit le total des paiements non finalisés en mode batch
- Lit le type d'une colonne
- 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 destfp
- Lit les données de résultat.
- Lit les données du vecteur specifié par iovec_id[iovec_position]
- Lit les données liées à une clé.
- Lit les en−têtes exif d'une image jpeg
- Lit les inforamtions sur le champs.
- Lit les informations à propos de la boîte aux lettres courante.
- Lit les informations d'un champs.
- Lit les informations sur l'hôte mysql
- 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 un champs
- 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 des boîtes aux lettres
- Lit n bytes d'un objet slob.
- Lit tous les entêtes de réponse http
- Lit tous les entêtes http de la requête
- 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 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
- Lit toutes les valeurs de configuration
- Lit un blob dans une base de données
- Lit un bloc
- Lit un buffer contenant des données pdf
- Lit un caractère dans un fichier compressé
- Lit un champs de résultat mnogosearch
- Lit un clob dans une base de données
- 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 en totalité.
- Lit un objet de grande taille.
- Lit un paramètre de réponse personnalisé
- Lit un paramètre de réponse utilisateur
- Lit un sommaire des en−têtes de messages
- Lit un tablaeu 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 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 comme un tableau.
- Lit une ligne dans un ficheir compressé, et supprime les balises
- Lit une ligne dans un fichier compressé
- Lit une ligne dans un objet.
- Lit une ligne dans un résultat
- Lit une ligne dans un résultat
- Lit une ligne dans un tableau
- Lit une ligne dans un tableau
- Lit une ligne dans un tableau associatif
- Lit une ligne dans un tableau.
- Lit une ligne dans un tableau.
- Lit une ligne dans un tableau.
- Lit une ligne dans une base interbase
- Lit une ligne dans une base interbase dans un objet.
- Lit une ligne de résultat sous forme d'objet
- Lit une ligne de résultat sous forme de tableau numérique
- Lit une ligne de résultat, et la place dans un tableau.
- Lit une ligne de résultat.
- 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 textuelle de eturn_code
- Lit une représentation textuelle de return_avs
- Lit une représentation textuelle de return_cv
- Lit une sous−chaîne
- Lit une valeur dans un résultat.
- Lit une valeur dans une session
- Lit une valeur dans une structure usersetup
- Lit une valeur.
- 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
- Locks an object
- Log into hyperwave server
- 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
- Make a string lowercase
- Make a string uppercase
- Marque la note note_id comme lu par l'utilisateur user_name
- Marque la note note_id comme non plus 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 le premier caractère en majuscule
- Modifie certains paramètre numériques
- Modifie certains paramètres.
- Modifie l'échelle de la forme
- Modifie l'échelle.
- Modifie l'échelle.
- Modifie l'épaisseur d'un trait
- Modifie l'action d'un formulaire
- Modifie l'action javascript d'un champs
- Modifie l'alignement du texte
- Modifie l'encodage des caractères
- 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 champs.
- Modifie l'origine du système de coordonnées.
- Modifie l'origine du système de coordonnées.
- Modifie la "miter limit".
- Modifie la brosse pour le dessin des lignes
- Modifie la couleur de dessin à un niveau de gris.
- Modifie la couleur de fond.
- Modifie la couleur de la police

- Modifie la couleur de rastériation de droite.
- Modifie la couleur de rastérisation de gauche.
- Modifie la couleur des liens et annotations
- Modifie la couleur du champs texte
- Modifie la couleur grise comme couleur de remplissage et de dessin.
- Modifie la couleur grise comme couleur de remplissage.
- Modifie la couleur rgb comme couleur de dessin et de remplissage.
- Modifie la couleur rgb comme couleur de dessin.
- Modifie la couleur rgb comme couleur de remplissage.
- 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 largeur de ligne.
- Modifie la marge de droite.
- Modifie la marge de gauche.
- Modifie la place en profondeur (z−order)
- Modifie la platitude (flatness).
- Modifie la police courante
- Modifie la police du champs
- Modifie la prochaine ligne dans le résultat.
- Modifie la structure usersetup d'un utilisateur
- Modifie la valeur d'une option ldap
- Modifie la vitesse de l'animation.
- Modifie le bord des liens et annotations
- Modifie le contenu d'un objet blob.
- Modifie le contenu d'un objet char.
- Modifie le gestionnaire de référence externes.
- 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'une entrée
- Modifie le nom de la méthode appelée
- Modifie le nombre total d'images dans l'animation.
- Modifie le paramètre linecap.
- Modifie le paramètre linejoin.
- 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 système de coordonnées.
- Modifie le type xmlrpc, base64 ou datetime, pour une chaînes de caractères php
- Modifie les attributs d'objet record.
- Modifie les attributs du terminal et le taux de baud du port série
- Modifie les caractères de remplissage.
- Modifie les dimensions de l'animation.
- Modifie les données (??)
- Modifie les données du vecteur iovec_id[iovec_position]
- 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 marges du champs texte.
- Modifie les options de la connexion ftp
- Modifie les options de socket
- Modifie les paramètres de l'agent mnogosearch
- Modifie les paramètres de transaction sesam
- Modifie les paramètres du cookie de session
- Modifie les paramètres odbc.
- Modifie les pointillés compliqués
- Modifie les pointillés des liens et annotations
- Modifie un événement dans un calendrier mcal.
- Modifie un attribut
- Modifie un bit
- Modifie un niveau de gris comme couleur de dessin et de remplissage.
- Modifie un objet.
- Modifie un tableau de ... (??)
- Modifie une entrée ldap
- Modifie une ligne dans une relation
- Modifie une option d'un champs
- Modifie une option d'un champs
- Modifie une option de transfert curl
- 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
- Mot la valeur d'un champs.
- Move current position and add character
- Move cursor immediately
- Move output position
- Move position and add attributed string
- Move position and add attrributed string with specified length
- Move position and add string
- Move position and add string with specified length
- Move position and delete character, shift rest of line left
- Move position and get attributed character at new position
- Move position and get character at new position
- Moves object between collections
- Multiplication de 2 nombres gmp
- Multiplie deux nombres de grande taille.
- Multiplie la couleur de transformation.

N

- Ncurses is in endwin mode, normal screen output may be performed
- 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 ligne 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
- Non documenté
- 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 : gestion de nom de police
- Obsolète : gestion de police
- Obsolète : gestion de taille de police
- Obsolète : modifie la distance entre les lignes du texte
- Obsolète: ajoute un signet dans la page courante
- Obsolète: ajoute une annotation
- Obsolète: modifie la transition des pages
- 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
- Open connection to a printer
- Opposé de nombre gmp
- Optimise une base.
- Options disponibles pour les expressions régulières.
- Ou exclusif logique
- Ou logique
- Output text at current position
- Ouverture d'un fichier ou d'une url.
- Ouverture d'une base dbase.
- 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 document fdf.
- 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 flot imap vers une boîte aux lettres.
- Ouvre un flot imap vers une nouvelle boîte aux lettres.
- Ouvre un nouveau dictionnaire cracklib dictionary
- Ouvre un nouveau document pdf.
- Ouvre un nouveau fichier shockwave flash
- Ouvre un nouvel objet pdf
- Ouvre un objet de grande taille.

- Ouvre un objet slob.
- Ouvre un pointeur oracle.
- Ouvre une archive zip
- Ouvre une base de données dbm
- Ouvre une base de données.
- 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 sybase.
- Ouvre une connexion à une base de données
- Ouvre une connexion à une base de données interbase.
- Ouvre une connexion ftp
- Ouvre une connexion hyperwave.
- Ouvre une connexion mcal.
- 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 sybase.
- Ouvre une connexion persistante à une base de données interbase.
- Ouvre une connexion persistante à une base de données.
- Ouvre une connexion persistante à une source de données.
- Ouvre une connexion persistante mcal.
- Ouvre une connexion sesam
- Ouvre une connexion.
- Ouvre une image créée par les fonctions images php.
- Ouvre une image jpeg.
- Ouvre une nouvelle image à partir de données ccitt
- Ouvre une socket de connexion internet ou unix persistante.
- Ouvre une socket de connexion internet ou unix.
- Ouvre une socket sur un port pour accepter les connexions
- Ovre un nouveau fichier

P

- Pas encore implémenté.
- Pas encore implémentée
- 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
- 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
- Performs a command which is peculiar to a single protocol, and thus is not available in the general ccvs api

- Pgcd
- Pgcd étendu
- Pingue le serveur, et se reconnecte au besoin
- Place un objet sur la scène.
- Place une image dans la page.
- 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.
- Prépare une police pour utilisation ultérieure
- 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 déchargement
- 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 requête pour lier les paramètres et l'éxécuter ultérieurement.
- Prépare une requête sql
- Prépare une requête sql pour l'exécution.
- Prépare une requête z39.50 item order avec le package ill−request
- Prépare une transaction
- Prépares une recherche
- Prend une ou plusieurs valeurs, au hasard dans un tableau
- Protège les caractères spéciaux d'une commande sql.
- Protège une chaîne de caractères pour l'utiliser dans une requête
- Protège une chaîne pour la passer à mysql_query.
- Puissance
- Puissance
- Puissance et modulo
- Pushes mouse event to queue
- Put a character back into the input stream
- Put terminal into halfdelay mode

Q

- Quitte le canal

R

- Réactive l'ancienne fonction de gestion des erreurs
- Récupère toutes les en−têtes des requêtes http.
- Récupère un ienumvariant
- Récupère une ligne de résultat dans un objet.
- Récupère une ligne de résultat dans un tableau énuméré.
- Récupère une ligne de résultat dans un tableau.
- Réduit itérativement un tableau
- Réouvre une connexion mcal
- Répéte une chaîne
- Réserve un sémaphore.

- Résolution dns d'une adresse ip.
- Résoud les informations d'hôte pour une relation
- Racine carrée avec reste gmp
- Racine carrée gmp
- Racine carrée.
- 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
- Rassemble plusieurs tableaux
- 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 objet `snmp` .
- Read a character from keyboard
- Read content
- Reads mouse event
- Reads string from terminal screen
- 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 par expression régulière insensible à la casse.
- Recherche sur le serveur ldap
- Recherche un événement dans le calendrier.
- Recherche un message dans le domaine courant.
- Recherche un objet dans un groupe.
- Recherche un objet dans un groupe.
- Recherche un objet.
- Recherche un objet.
- Rechercher et remplacer par expression régulière standard.
- Rechercher/remplacer avec 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
- Refresh screen
- Refresh window on terminal screen
- Rejoint un canal irc
- Remet à zéro la session courante.
- Remet le pointeur interne de tableau au début
- Removes attribute
- Remplace des caractères
- 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 chaînes

- Remplace un attribut dans l'entrée courante
- Remplace un enregistrement dans une base dbase.
- Remplace un segment dans une chaîne
- Remplace une valeur.
- Remplacement par expression régulière insensible à la casse.
- Remplacement par expression régulière.
- Remplis et dessine le chemin courant.
- Remplis le chemin courant.
- Remplis les entêtes du document
- Remplis un tableau avec une même valeur
- Remplis, dessine et ferme le chemin courant.
- Remplit avec une région avec une couleur spécifique.
- Remplit le chemin courant.
- Remplit le chemin, dessine le bord et ferme le chemin.
- Remplit le chemin, et dessine le bord.
- Remplit.
- Renomme un calendrier
- Renomme un fichier
- Renomme un fichier sur un serveur ftp.
- Renomme une boîte aux lettres.
- Renomme une relation
- Renverse l'ordre des éléments d'un tableau
- 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 caractère que pointe le pointeur 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 ou d'un lien symbolique.
- Renvoie les informations à propos d'un fichier.
- Renvoie les informations à propos d'un lien.
- Renvoie les permissions affectées au fichier.
- Repère la dernière occurence d'un caractère dans une chaîne
- Repère la première occurence d'un caractère dans une chaîne
- Repère un fichier à partir d'un masque
- Replace le pointeur de fichier au début.
- Replaces an object
- Représente un id globale en un id virtuel local.
- Reprend le téléchargement d'un fichier (non bloquant)
- Requiert les informations sur un utilisateur
- Restaure la valeur de l'option de configuration
- Restaure un environnement sauvé.
- Restaure un environnement.
- Reste de la division de deux nombres gmp

- Restore screen from file dump
- Restores saved terminal state
- Restores soft label keys
- Retarde l'exécution en micro−secondes
- Retarde l'exécution.
- Retire une contrainte d'une relation
- 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 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 statistiques sur les caractères utilisés dans une chaîne
- Retourne l'échelle d'un champ.
- Retourne l'élément courant d'un tableau
- Retourne l'élément racine
- Retourne l'état de la dernière requête sesam
- Retourne l'adresse ip correspondant à un hôte.
- Retourne l'en−tête d'un message.
- Retourne l'ensemble optimal de colonnes, qui permettent 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 de la note non lue pour l'utilisateur courant user_name
- Retourne l'identifiant du groupe de processus.
- Retourne l'identifiant du processus courant.
- Retourne l'identifiant du processus parent.
- Retourne l'identifiant du thread courant
- Retourne l'identifiant généré par la dernière requête insert.
- Retourne l'identifiant maximal de hash.
- 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, ou la plus proche possible.
- Retourne l'index de la couleur donnée.
- Retourne l'index de la couleur la plus proche d'une couleur donnée.
- Retourne l'inode du script.
- 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 d'erreur associée à la dernière opération.
- 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.
- 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'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 largeur du texte avec la police courante.
- Retourne la ligne associée.
- Retourne la ligne suivante 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 des constantes et leur valeur
- Retourne la liste des fichiers .ini analysédans les dossiers de configuration supplémentaires
- Retourne la liste des fichiers dans un dossier.
- Retourne la longueur d'un champs.
- Retourne la longueur d'une chaîne.
- Retourne la longueur du champs spécifié.
- Retourne la longueur du contenu du buffer de sortie
- Retourne la précision d'un champ.
- Retourne la première entrée
- Retourne la prochaine occurrence d'un événement.
- Retourne la taille d'un champ.
- Retourne la taille d'un champs
- Retourne la taille d'un champs.
- Retourne la taille d'un dossier
- Retourne la taille d'un fichier.
- Retourne la taille d'une chaîne
- Retourne la taille d'une colonne
- 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 de la colonne.
- 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 champs donné.
- Retourne la taille maximale de clé
- Retourne la taille maximale de la clé pour un mode
- Retourne la valeur d'un champs.
- Retourne la valeur d'une colonne
- Retourne la valeur d'une constante
- Retourne la valeur d'une option de php
- Retourne la valeur de la variable d'environnement.
- Retourne la valeur de la variable, au format chaîne.
- Retourne la valeur de pi
- Retourne la valeur numérique (double) de la variable.
- Retourne la valeur numérique (integer) de la variable.
- Retourne la version courante de curl
- Retourne le chemin canonique absolu.
- Retourne le chemin du terminal.
- Retourne le code ascii d'un caractère
- Retourne le code d'erreur de la dernière opération.
- Retourne le code d'erreur de la dernière requête informix.
- Retourne le code d'erreur frontbase
- Retourne le code d'erreur oracle.
- Retourne le code d'erreur.
- Retourne le code d'un processus fils terminé
- Retourne le code php utilisé pour générer une variable
- 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 couple (clé ; valeur) suivant d'une carte donnée.
- Retourne le dernier message d'erreur du serveur (min_message_severity?).
- Retourne le dernier message du serveur
- Retourne le domaine nis par défaut.
- Retourne le dossier de travail
- Retourne le fichier courant, ou la position courante.
- Retourne le flag d'un champs.
- Retourne le gid du propriétaire du script
- Retourne le logo
- Retourne le logo de zend
- Retourne le message 'derreur de la dernière opération qtdom, ou `false` si aucune erreur n'est survenue
- Retourne le message d'erreur
- Retourne le message d'erreur
- Retourne le message d'erreur bzip2
- Retourne le message d'erreur courant
- Retourne le message d'erreur de la dernière requête informix.
- Retourne le message d'erreur frontbase
- Retourne le message d'erreur openssl
- Retourne le message d'erreur oracle.
- Retourne le message de diagnostic
- Retourne le message ldap de la dernière commande ldap.
- Retourne le niveau d'utilisation des ressources.
- Retourne le nom d'hôte correspondant à une ip.
- Retourne le nom d'hôte.

- Retourne le nom d'un champ dans le résultat d'une requête.
- Retourne le nom d'un champs.
- Retourne le nom d'un champs.
- Retourne le nom d'une colonne
- Retourne le nom d'une colonne
- Retourne le nom d'une colonne
- Retourne le nom d'une colonne.
- Retourne le nom d'une table à partir d'un nom de champs.
- 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 la classe d'un objet
- Retourne le nom de la colonne de résultat.
- 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 curseur
- 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 possesseur du script courant.
- Retourne le nom du système.
- Retourne le nombre courant de colonne d'un analyseur xml.
- Retourne le nombre courant de colonne d'un analyseur xml.
- Retourne le nombre d'élément d'un tableau
- Retourne le nombre d'arguments passé à la fonction
- Retourne le nombre de champs
- Retourne le nombre de champs d'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.
- Retourne le nombre de champs dans un résultat.
- 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
- Retourne le nombre de colonnes dans un résultat
- Retourne le nombre de colonnes dans un résultat.
- Retourne le nombre de colonnes dans une requête.
- Retourne le nombre de jour d'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 lignes
- 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.
- Retourne le nombre de lignes affectées par une modification
- Retourne le nombre de lignes affectées par une requête.
- Retourne le nombre de lignes affectées.
- Retourne le nombre de lignes d'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.
- Retourne le nombre de lignes dans un résultat.
- 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 résultat de la dernière recherche
- 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 bzip2
- Retourne le numéro d'erreur courant
- 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
- 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 message d'erreur de la dernière opération mysql.
- Retourne le numéro de port associé à un service internet et un protocole.
- Retourne le numéro de port.
- 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 du jour de la semaine.
- Retourne le numéro et le message d'erreur bzip2 dans un tableau
- Retourne le premier couple (clé ; valeur) d'une carte donnée.
- Retourne le prochain identifiant d'objet libre.
- Retourne le résultat d'un scan
- Retourne le résultat de exp(number) − 1, calculé de manière précise, même si val est proche de 0
- Retourne le résultat de log(1 + number), calculé de manière précise, même si val est proche de zéro.
- Retourne le résultat de sqrt(num1*num1 + num2*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 nombre de grande taille.
- Retourne le reste de la division
- Retourne le sémaphore associé à la colonne spécifiée dans le résultat courant.
- Retourne le service internet qui correspond au port et protocole.
- Retourne le sid du processus.
- Retourne le statut courant du serveur
- 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 gmt.

- Retourne le timestamp unix d'une date.
- 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 champs donné par index.
- Retourne le type d'un champs.
- Retourne le type de champs.
- Retourne le type de commande oci.
- Retourne le type de données d'une colonne.
- Retourne le type de fichier
- Retourne le type de la colonne de résultat.
- Retourne le type de la colonne spécifiée dans le résultat courant.
- Retourne le type de la variable.
- Retourne le type de ressource
- Retourne le type numérique d'une colonne
- 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 d'un noeud
- Retourne les attributs, et verrouille l'objet.
- Retourne les bits de status de la connexion.
- Retourne les données de résultat.
- 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 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 lignes résultats sous la forme d'un objet.
- Retourne les limites système.
- Retourne les noms des méthodes d'une classe.
- Retourne les notes rangées dans database_name
- Retourne les options.
- Retourne les types d'images supportés par la version courante de php
- 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.
- Retourne un élément de la liste des arguments
- Retourne un analyseur destiné au traitement d'un message
- Retourne un attribut d'un noeud
- Retourne un caractère spécifique
- Retourne un champs d'un résultat.
- Retourne un document distant.
- Retourne un document texte.

- Retourne un document texte.
- Retourne un document.
- Retourne un entier représentant le dernier message d'erreur.
- 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 pointeur à utiliser pour lier les pointeurs de références
- Retourne un objet contenant la structure de date pour le flot courant.
- Retourne un résultat
- Retourne un segment de chaîne
- Retourne un tableau associatif avec des informations sur le message
- Retourne un tableau associatif décrivant les classes aggrégées
- Retourne un tableau associatif des propriétés d'un objet
- 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 noms des fichiers qui sont requis et 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 de message après recherche.
- Retourne un timestamp unix pour pâques, à minuit
- Retourne une adresse email proprement formatée
- Retourne une chaîne contenant le dernier message d'erreur
- Retourne une chaîne contenant les informations de version du serveur.
- Retourne une chaîne formatée
- Retourne une chaîne formattée
- Retourne une chaîne décrivant un message d'erreur
- Retourne une clé d'un tableau associatif
- Retourne une description de l'erreur
- Retourne une donnée d'une ligne lue.
- Retourne une ligne de résultat sous la forme d'un tableau associatif.
- Retourne une ligne de résultat sous la forme d'un tableau.
- Retourne une ligne de résultat.
- Retourne une ligne sous la forme d'un objet.
- Retourne une ligne sous la forme d'un objet.
- Retourne une ligne sous la forme d'un objet.
- Retourne une ligne sous la forme d'un tableau énuméré.
- Retourne une ligne sous la forme d'un tableau énuméré.
- Retourne une ligne sous la forme d'un tableau.
- 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.
- Retrieve attribute information
- Retrieve printer configuration data
- Retrieves header/meta data from streams/file pointers

- Return an array of printers attached to the server
- 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 a list of all destination anchors
- Returns a list of all logged in users
- Returns a list of all source anchors
- Returns a logical or of all attribute flags supported by terminal
- Returns all values of the attribute
- Returns baudrate of terminal
- Returns children of an object
- Returns content of an object
- Returns current erase character
- Returns current line kill character
- Returns current soft label key attribute
- Returns description of reason
- Returns destination of a source anchor
- Returns information about server configuration
- Returns mimetype
- Returns number of attributes
- Returns number of reasons
- Returns object of class hw_api
- Returns parents of an object
- Returns reason of error
- Returns source of a destination object
- Returns statistics about database server
- Returns statistics about document cache server
- Returns statistics about fulltext server
- Returns statistics about hyperwave server
- Returns terminals (short)−name
- Returns terminals description
- Returns the object an anchor belongs to
- Returns the own user object
- Returns the signal which caused the child to stop
- Returns the signal which caused the child to terminate
- Returns the title attribute
- Returns the translation table used by `htmlspecialchars` and `htmlentities`
- Returns type of reason
- Returns value for a given language
- Returns value of attribute
- Returns value of the attribute
- Rotation de la transformation courante.

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 la police et sa taille.
- Sélectionne le mode non bloquant d'un descripteur de fichier
- Sélectionne un format d'affichage pour les messages irc

- Sélectionne une base de données frontbase
- Sélectionne une base de données msql.
- Sélectionne une base de données msql.
- Sélectionne une base de données mysql.
- Sélectionne une base de données sybase.
- Sélectionne une nouvelle zone pour un dessin ultérieur.
- 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 sur un serveur cyrus imap
- Sauve dans un fichier.
- Sauve l'environnement courant.
- Sauve le dictionnaire personnel dans un fichier.
- Sauver un document fdf.
- Saves terminal state
- Saves terminals (program) mode
- Saves terminals (shell) mode
- 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 un tableau, grâce à une expression régulière.
- Scinde une chaîne en un tableau, grâce à une expression régulière.
- Scroll window content up or down without changing current position
- Se connecte à un serveur cyrus imap
- Se connecte à un serveur ldap.
- Se connecte à un serveur oracle avec une nouvelle connexion.
- Search for objects
- Select a brush
- Select a font
- Select a pen
- Set background property for terminal screen
- Set cursor state
- Set fore− and background color
- Set given attributes
- Set new position and draw a horizontal line using an attributed character and max. n characters long
- Set new position and draw a vertical line using an attributed character and max. n characters long
- Set new rgb value for color
- Set the effective uid of the current process
- Set timeout for mouse button clicks
- Set timeout for special key sequences
- Sets color for soft label keys
- Sets mouse options
- Sets the value of a configuration option
- Signe du nombre gmp
- Signe les données
- Signe un message s/mime
- Single character output including refresh
- Sinus
- Sinyus hyperbolique
- Sleep

- 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
- Spécifie la nouvelle clé primaire d'une relation
- Spécifie la syntaxe de lecture des lignes
- Spécifie le nombre maximal de résultat à lire
- Spécifie le type d'éléments à lire
- Specify different filedescriptor for typeahead checking
- Start a new document
- Start a new page
- Start using 'standout' attribute
- Start using colors
- Stop using 'standout' attribute
- Stop using ncurses, clean up the screen
- Stoppe une base de données
- Suggèegrave;re une orthographe
- Suggère l'orthographe d'un mot [obsolète]
- Supprime la ligne courante et retourne la nouvelle ligne courante
- Supprime la récurrence de la structure globale.
- 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 tous les événements marqués pour l'effacement
- Supprime toutes les lignes d'une relation
- Supprime un descripteur de lob
- 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 lotus notes
- Supprime une fonction de callback sur une connexion cyrus imap
- Supprime une structure usersetup
- Supprime une variable de la session
- Supprimer les anti−slash d'une chaîne
- Switch of input buffering
- Switch off keyboard input echo
- Switch terminal into raw mode
- Switch terminal out of raw mode
- Switch terminal to cooked mode
- Symbole de jacobi
- Symbole de legendre
- Synchronise avec le serveur postgresql
- Synchronise une base de données.
- Synonyme de `odbc_exec`

- **T**

- Télécharge un fichier depuis un serveur ftp et le sauve dans un fichier déjà ouvert.
- Télécharge un fichier depuis un serveur ftp.
- Taille d'un document.
- Tangente
- Tangente hyperbolique
- Terminate ccvs engine and do cleanup work
- Termine l'action courante.
- Termine la définition de symbole.
- Termine la définition du bouton courant.
- Termine la souscription à une boîte aux lettres.
- Termine le processus apache après cette requête
- Termine le script courant.
- Termine le suivi d'une connexion postgresql
- Termine un chiffrage
- Termine un document.
- Termine un flot imap.
- Termine une connexion mcve_conn
- Termine une connexion postgresql.
- Termine une page.
- Termine une page.
- Termine une section de texte.
- Teste un module ouvert
- Teste la fin du fichier.
- Teste le chiffrage par blocs d'un algorithme
- Teste le chiffrage par blocs d'un mode
- Teste si la valeur d'une colonne est null.
- Teste si le mode retourne les données par blocs
- Teste si un champ est annulable.
- Teste si un champs est à `null` .
- Teste un mode
- Tourne la forme
- Tourne un objet en angle absolu.
- Traite un fichier de configuration
- Transfer funds from the credit card holder to the merchant
- Transfer funds from the merchant to the credit card holder
- Transforme des données xml
- 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 la transformation courante.
- Translate newline and carriage return / line feed
- 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 d'un tableau en utilisant une fonction de comparaison définie par l'utilisateur.
- Trie des messages.
- Trie en ordre inverse
- Trie le tableau
- Trie les entrées d'un résultat ldap
- 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 définie par l'utilisateur
- ♦ Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur
- ♦ Trie un tableau suivant les clés
- ♦ Tronque le fichier
- ♦ Tronque un fichier.
- ♦ Tronque une chaîne
- ♦ Trouve la position d'un caractère dans une chaîne
- ♦ Trouve la position de la dernière occurence d'un caractère dans une chaîne
- ♦ Trouve la première occurence d'un caractère dans une chaîne
- ♦ Trouve la première occurence dans une chaîne
- ♦ Trouve la première occurence 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
- ♦ Turn off the given attributes
- ♦ Turn on the given attributes
- ♦ Type de données d'un champs.

- **U**

- ♦ Unlocks a locked object
- ♦ Utilisation des ressources.
- ♦ Utilise un analyseur xml à l'intérieur d'un objet.
- ♦ Utilise une variable php pour la phase de définition, dans un select.
- ♦ Utilise une variable php pour la phase de définition, dans une commande select.

- **V**

- ♦ Vérifie l'état de la connexion
- ♦ Vérifie l'usage d'un certificat
- ♦ Vérifie la signature d'un message s/mime
- ♦ Vérifie le courrier de la boîte aux lettres courante.
- ♦ Vérifie le message de confirmation
- ♦ Vérifie qu'un caractère est alpha−numérique
- ♦ Vérifie qu'un caractère est alphabétique
- ♦ Vérifie qu'un caractère est caractère blanc (espace, tabulation...)
- ♦ Vérifie qu'un caractère est en majuscule
- ♦ Vérifie qu'un caractère est en minuscule
- ♦ Vérifie qu'un caractère est imprimable
- ♦ Vérifie qu'un caractère est imprimable (sauf " ", espace)
- ♦ Vérifie qu'un caractère est imprimable, sans être ni un espace, ni un caractère alpha−numérique
- ♦ Vérifie qu'un caractère est numérique
- ♦ Vérifie qu'un caractère est un caractère de contrôle
- ♦ Vérifie qu'un caractère représente un nombre héxadécimal
- ♦ Vérifie qu'un identifiant d'objet est dans un groupe.
- ♦ Vérifie qu'une clé existe.
- ♦ Vérifie qu'une classe a été définie
- ♦ Vérifie qu'une constante existe.
- ♦ 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 flot imap est toujours actif.
- ♦ Vérifie si la transaction a réussi
- ♦ Vérifie si le buffer de sortie est vide ou pas
- ♦ Vérifie si un fichier existe.

- ♦ Vérifie si une assertion est fausse
- ♦ Vérifie si une clé éxiste dans un tableau
- ♦ 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 sesam en cours
- ♦ Valide le mot de passe d'un utilisateur pour un domaine. retourne `true` / `false`
- ♦ Valide les transactions en cours.
- ♦ Valide une date.
- ♦ Valide une date/heure.
- ♦ Valide une heure.
- ♦ Valide une transaction
- ♦ Valide une transaction
- ♦ Valide une transaction
- ♦ Valide une transaction odbc
- ♦ Valide une transaction oracle.
- ♦ Valide une transaction.
- ♦ Verifie le mot de passe
- ♦ Verrouille le fichier.
- ♦ Verrouille une session
- ♦ Vide les buffers de sortie.

- **W**
- ♦ Write all prepared refreshes to terminal
- ♦ Write data to the printer

10.3 Index des exemples

- ♦ **A**
- ♦ ◊ A stream for reading/writing global variables
  - ◊ Accéder aux données du formulaire
  - ◊ Exemple avec addaction.php">swfbutton−>addaction
  - ◊ Exemple avec addcslashes
  - ◊ Exemple avec addfill.php">swfshape−>addfill
  - ◊ Affectation d'une colonne de type "champs multiple"
  - ◊ Affichage d'adresse ip
  - ◊ affichage de $http_post_vars avec each
  - ◊ Affichage de texte
  - ◊ Affichage de texte
  - ◊ Affichage des fichiers inclus et requis
  - ◊ Affichage sous la forme d'une table html
  - ◊ Affiche la lsite des attributs d'une entrée
  - ◊ Afficher des données issues d'un formulaire
  - ◊ Afficher une erreur sesam
  - ◊ Afficher une structure xml
  - ◊ Exemple avec aggregation_info
  - ◊ Ajout d'entrées dans un tableau.
  - ◊ Ajouter un attribut à un élément

◊ Ajouter une mise en relief
◊ Annulation d'une transaction sesam
◊ Exemple avec apache_request_headers
◊ Argument de fonction de lecture
◊ Exemple avec array_change_key_case
◊ Exemple avec array_chunk
◊ Exemple avec array_count_values
◊ Exemple avec array_diff
◊ Exemple avec array_fill
◊ Exemple avec array_filter
◊ array_flip example : collision
◊ Exemple avec array_intersect
◊ Exemple avec array_key_exists
◊ array_map − création d'un tableau de tableaux
◊ 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_slice
◊ Exemple avec array_sum
◊ array_unique et les types de valeurs
◊ 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
◊ authentification http avec nom d'utilisateur/mot de passe forcé
◊ Avoid_error_require_once.php

♦ **B**

♦ ◊ base_convert
◊ Exemple avec basename
◊ Exemple avec bzcompress
◊ Exemple avec bzdecompress
◊ Exemple avec bzerror
◊ Exemple avec bzip2
◊ Exemple avec bzopen
◊ Exemple avec bzread

♦ **C**

♦ ◊ Calcul du hash et enregistrement d'un utilisateur
◊ Calcule le md5 et le hmac, puis l'affiche comme un hexadécimal
◊ Chargement multiple de fichier
◊ Exemple avec chr
◊ Exemple avec chunk_split
◊ Classer un tableau multidimensionnel
◊ Collection
◊ Collection

◊ Colorisation d'url
◊ Commande utilisables avec mysql_info()
◊ Exemple avec compact
◊ Compactage d'une chaîne
◊ Compilation cgi avec php (unix)
◊ Compilation sur apache avec l'extension postgresql (2) (librairie partagée)
◊ Configuration de php.ini pour les utilisateurs de sjis
◊ Configuration par défaut des paramètres
◊ Configurer php avec apache 2.0 en module
◊ Connection à un serveur informix
◊ connection au serveur ovrimos sql server et sélection d'une table système
◊ connexion à un serveur ovrimos sql server et préparation d'une requête
◊ Connexion à une base sesam
◊ Conversion d'un nombre gmp en chaîne
◊ Conversion html en texte
◊ Exemple avec copy
◊ Exemple avec count
◊ Counting the number of hits of a single user
◊ création d'un fichier flash simple, basé sur une entrée de l'utilisateur, et sauvegarde dans une base.
◊ Création d'un fichier pdf en mémoire
◊ Création d'un nombre gmp
◊ Création d'une base dbase
◊ Création d'une en−tête html simple
◊ Création d'une image png avec php
◊ Créer un fichier gzip
◊ Créer un nouveau bloc
◊ Cracklib example
◊ Critères de tri
◊ Exemple avec crypt

♦ **D**

♦ ◊ Déconnexion d'une base sesam
◊ Définition d'une constante
◊ Désactiver l'analyse des fichiers php dans un dossier avec .htaccess
◊ Détecter des variables non protégées avec e_all
◊ 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_query
◊ Exemple avec dbx_sort
◊ Definir une constante
◊ Demarrer les règles de réécriture d'apache pour hyperwave
◊ Dernier jour du mois
◊ Destruction d'une session avec $_session
◊ Deuxième étape de paiement (équivalent à cgi2.c)
◊ Exemple avec dir
◊ Exemple avec dirname
◊ Exemple avec disk_free_space
◊ Exemple avec disk_total_space
◊ Division de nombres gmp

♦ **E**

◊ Exemple avec easter_days
◊ Ecrire un bloc de mémoire partagée
◊ Effacement d'un bloc de mémoire partagée
◊ Effacer une ressource existante
◊ Encryption d'une valeur avec tripledes sous 2.4.x en mode ecb
◊ enregistrer une variable avec register_globals activé
◊ Enumerer tous les messages d'erreur ldap
◊ Envoi de mail complexe.
◊ Exemple avec ereg_replace
◊ exemple avec eval − inclusion de texte
◊ Evaluer un document fdf
◊ Example com (2)
◊ Exemle avec fbsql_create_db
◊ Exemle avec swftext
◊ Exemle avec trim
◊ Exempe avec constant
◊ Exemple
◊ Exemple bzwrite
◊ Exemple ereg
◊ Exemple fbsql_num_rows
◊ Exemple get_class_vars
◊ Exemple ibase_connect
◊ Exemple iconv_set_encoding
◊ Exemple imap_delete
◊ Exemple imap_mail_compose
◊ Exemple imap_mime_header_decode
◊ Exemple imap_status
◊ Exemple mhash_get_hash_name
◊ Exemple msql_tablename
◊ Exemple mysql_data_seek
◊ Exemple mysql_db_name
◊ exemple mysql_num_rows par crubel@trilizio.org
◊ Exemple mysql_stat
◊ Exemple mysql_tablename
◊ Exemple opendir
◊ Exemple ora_fetch_into
◊ Exemple pfpro_process_raw
◊ Exemple php_sapi_name
◊ Exemple php_uname
◊ Exemple pspell_add_to_personal
◊ Exemple realpath
◊ Exemple session_name
◊ Exemple setline.php">swfshape−>setline
◊ Exemple setname.php">swfdisplayitem−>setname
◊ Exemple socket_set_timeout
◊ Exemple strftime
◊ Exemple zend_version
◊ Exemple awt
◊ Exemple com (1)
◊ Exemple complet avec identification
◊ Exemple complet de vérification de mot de passe avec ldap
◊ exemple d'envoi de page compressée avec ob_gzhandler
◊ Exemple d'installation de php sous openbsd avec ports
◊ Exemple d'installation pour netscape enterprise sous solaris

◊ Exemple d'introduction
◊ Exemple d'ordre nis
◊ Exemple d'utilisation de zip
◊ Exemple de swfsprite
◊ Exemple de callback avec fonction utilisateur
◊ Exemple de chaîne heredoc
◊ Exemple de configuration apache
◊ Exemple de configuration ccl
◊ exemple de configuration pour netscape enterprise
◊ Exemple de configuration pour netscape enterprise 4.x
◊ Exemple de contrôle de processus
◊ Exemple de création de base mysql
◊ Exemple de fonction variable
◊ 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 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
◊ Exemple de maître nis
◊ Exemple de message du débuggeur
◊ Exemple de meta tags
◊ Exemple de modification d'option odbc
◊ Exemple de présentation de l'extension mysql
◊ Exemple de recherche nis
◊ Exemple de script avec mbstring
◊ Exemple ftp
◊ Exemple getrusage
◊ Exemple mysql connect
◊ Exemple mysql_close
◊ Exemple payflow pro
◊ Exemple pdfclock de la distribution pdflib 2.0
◊ Exemple pdfclock issue de la distribution pdflib
◊ exemple pour ingres_connect utilisant le lien par défaut
◊ Exemple pour ingres_fetch_array
◊ Exemple pour ingres_fetch_object
◊ Exemple pour ingres_fetch_row
◊ Exemple pour ingres_query
◊ Exemple simple avec recode_file
◊ Exemple simple avec recode_string
◊ Exemple swf
◊ Exemples
◊ Exemples echo
◊ Exemples avec array_splice
◊ Exemples avec array_unshift
◊ Exemples avec easter_date
◊ Exemples avec error_log
◊ Exemples avec error_reporting
◊ Exemples avec pg_connect
◊ Exemples avec print

♦ ◊ Exemples avec setcookie
◊ Exemples avec soundex
◊ Exemples de chaînes
◊ Exemples de masques invalides
◊ Exemple avec exit
◊ Exemple avec explode
◊ Expressions régulières
◊ Exemple avec extract
♦ **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_data_seek
◊ 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_blob
◊ Exemple avec fbsql_tablename
◊ Fermer une connexion informix
◊ Fermeture d'un bloc de mémoire partagée
◊ exemple avec fgetcsv
◊ Fichier batch pour exécuter un script php en ligne de commande (script.bat)
◊ Exemple avec fmod
◊ Fonction php qui scanne les titres
◊ Fonctions à nombre d'arguments variable
◊ Fonctions calendrier
◊ Exemple avec fopen
◊ Formatage avec date
◊ Exemple avec ftp_get_option
◊ Exemple avec ftp_set_option
♦ **G**
♦ ◊ Genérer et intercepter une erreur
◊ Gestion d'erreurs avancées en php
◊ Gestionnaire d'exception java
◊ Exemple avec get_browser
◊ Exemple avec get_class_methods
◊ Exemple avec get_object_vars
◊ Exemple avec getallheaders
◊ Exemple avec getdate
◊ getimagesize qui retourne iptc
◊ Exemple avec getlastmod
◊ Exemple avec getopt
◊ Exemple avec gmdate
◊ Exemple avec gmstrftime

◊ gzopen example
◊ Exemple avec gzread
♦ **H**
♦ ◊ Exemple avec hexdec
◊ Exemple avec htmlspecialchars
♦ **I**
♦ ◊ Exemple avec iconv
◊ Exemple avec ifx_fetch_row
◊ Exemple avec ifx_fieldproperties
◊ Exemple avec imagettftext
◊ Exemple avec imagetypes
◊ Exemple avec imap_append
◊ Exemple avec imap_createmailbox
◊ Exemple avec imap_fetch_overview
◊ Exemple avec imap_get_quota
◊ Exemple avec imap_getmailboxes
◊ Exemple avec imap_listmailbox
◊ 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 de array_values pour les utilisateurs php 3
◊ Exemple avec implode
◊ In an url
◊ in_array avec le paramètre strict
◊ include en php 3 et php 4
◊ Inclusion d'une image mémoire
◊ initialiser une session curl et récupèration d'une page web.
◊ Insertion à grande vitesse dans une table
◊ Insertion de valeurs dans la table "catalogue"
◊ Installation de yaz
◊ Installation sur mac os x server
◊ Installer php comme module apache
◊ Instruction d'installation (module partagé apache 2)
◊ Instructions d'installation avec apache (version module)
◊ Instructions d'installation caudium
◊ instructions d'installation pour hp−ux 10
◊ Introduction à la mémoire partagée
◊ Inverse de htmlentities
◊ Inverser une chaîne avec strrev
♦ **J**
♦ ◊ Jeu de caractères qui fonctionnent avec php
♦ **K**
♦ ◊ Exemple avec krsort
◊ Exemple avec ksort
♦ **L**
♦ ◊ Exemple avec la table de traduction
◊ Exemple avec le domaine par défaut
◊ Exemple avec le paramètre convmap
◊ Lecture d'un fichier ligne par ligne
◊ Lecture d'un objet postgres

◊ Lecture de l'élément principal
◊ Exemple avec les sockets : client tcp/ip simple
◊ Lire la taille d'un bloc de mémoire partagée
◊ Lire les fils d'un noeud
◊ Lire les valeurs de sqlca.sqlerrd[x]
◊ Lire un bloc de mémoire partagée
◊ Lire un nom de domaine dans une url
◊ Exemple avec list
◊ 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 la durée de vie de la machine, et l'affiche
◊ Lit un exemple
◊ localeconv example
◊ Exemple avec ltrim

♦ **M**

♦ ◊ Méthode avancée
◊ 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_substitute_character
◊ Exemple avec mcrypt_create_iv
◊ Exemple avec mcrypt_encrypt
◊ Exemple avec mcrypt_get_cipher_name
◊ Exemple avec mcrypt_list_algorithms
◊ Exemple avec mcrypt_list_modes
◊ Exemple avec mcrypt_module_open
◊ Exemple avec mdecrypt_generic
◊ Exemple avec microtime
◊ Migration depuis 2.0: concaténation de chaînes
◊ Migration depuis 2.0: valeur retournées, nouvelle façon
◊ Migration: nouvelle syntaxe if..endif
◊ Migration: nouvelle syntaxe while..endwhile
◊ Mise à l'échelle
◊ Mise en italique d'un mot dans un texte
◊ Modification d'une variable d'environnement
◊ Modifier le taux de baud du port série
◊ Modifier les paramètres de configuration sesam
◊ Exemple avec mssql_next_result
◊ Exemple avec multcolor.php">swfdisplayitem−>multcolor
◊ Mysql fetch array
◊ Mysql fetch object
◊ Exemple avec mysql_client_encoding
◊ mysql_fetch_assoc
◊ Exemple avec mysql_list_processes
◊ mysql_query

◊ Exemple avec mysql_real_escape_string
◊ Exemple avec mysql_thread_id
♦ **N**
♦ ◊ Exemple avec natsort
◊ ncurses_getmouse example
◊ ncurses_mousemask example
◊ Nom de champs et type sql.
◊ Nombre de lignes affectées
◊ Notre premier script php : bonjour.php
◊ Nouveau comportement de strtok
◊ Nouvelles balises php
◊ Exemple avec number_format
♦ **O**
♦ ◊ Exemple avec ob_iconv_handler
◊ Exemple avec ocibindbyname
◊ Ocicolumnname
◊ Ocicolumnsize
◊ Exemple avec ocicolumntype
◊ Ocidefinebyname
◊ Ocifetchstatement
◊ Ocilogon
◊ Exemple avec ocinewdescriptor
◊ Exemple avec ocinlogon
◊ Ocinumcols
◊ Ocirowcount
◊ Exemple avec ociserverversion
◊ Exemple avec odbc_fetch_into (après php 4.0.6)
◊ 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 ord
◊ Output control example
◊ Overload avec une classe php
◊ Exemple avec ovrimos_connect
◊ Exemple avec ovrimos_result_all
♦ **P**
♦ ◊ Parcourir la liste des hash
◊ Exemple avec parse_ini_file
◊ Exemple avec parse_str
◊ parsemoldb.php − analyse moldb.xml et crée un tableau d'objet moléculaires
◊ Passer du mode php au mode html et vice−versa
◊ Passer en revue une base
◊ Passer en revue une base de données.
◊ Exemple avec pathinfo
◊ Exemple avec pg_affected_rows
◊ Exemple avec phpversion
◊ Postgres retourne une ligne
◊ Postgresql fetch array
◊ Pré remplir un formulaire pdf

◊ Préparer l'entête d'un document pdf
◊ Présentation de dbm
◊ Exemple avec preg_grep
◊ Première étape du paiment (équivalent à cgi1.c)
◊ printer_abort example
◊ printer_close example
◊ printer_create_dc example
◊ printer_draw_bmp example
◊ printer_draw_chord example
◊ printer_draw_elipse example
◊ printer_draw_line example
◊ printer_draw_pie example
◊ printer_draw_rectangle example
◊ printer_draw_roundrect example
◊ printer_draw_text example
◊ printer_get_option example
◊ printer_list example
◊ printer_logical_fontheight example
◊ printer_open example
◊ printer_select_brush example
◊ printer_select_font example
◊ printer_select_pen example
◊ printer_set_option example
◊ printer_start_doc example
◊ printer_write example
◊ Exemple avec proc_open
◊ Produit une liste de tous les services d'une société avec ldap_list
◊ Exemple avec pspell_add_to_personal
◊ Exemple avec pspell_add_to_personal
◊ pspell_check
◊ Exemple avec pspell_config_create
◊ Exemple avec pspell_config_ignore
◊ pspell_config_mode
◊ Exemple avec pspell_config_personal
◊ Exemple avec pspell_config_repl
◊ Exemple avec pspell_config_runtogether
◊ pspell_new
◊ pspell_new_config
◊ Exemple avec pspell_new_personal
◊ Exemple avec pspell_store_replacement
◊ pspell_suggest

♦ **Q**

♦ ◊ Quelques exemples avec pow

♦ **R**

♦ ◊ Exemple avec rawurlencode
◊ Exemple avec read_exif_data
◊ Exemple avec readline
◊ Recherche avec ldap
◊ Recherche ldap
◊ Recherche les couples de balises html (gourmand)
◊ Recherche paralléle utilisant yaz
◊ Reinstaller apache avec pthread
◊ Exemple avec rename
◊ reprise du téléchargement à la position 100 avec ftp_nb_get

◊ Resuming an upload with ftp_nb_put
◊ Retrieve an object
◊ Exemple avec rotateto.php">swfdisplayitem−>rotateto
◊ Exemple avec rsort
◊ Exemple avec rtrim

♦ **S**

♦ ◊ Sauver et restaurer un environnement pdf
◊ Sauver/restaurer
◊ Scinder une chaîne en caractères
◊ Exemple avec serialize
◊ Exemple avec sesam_fetch_array
◊ Exemple avec sesam_fetch_row
◊ Exemple avec session_cache_limiter
◊ exemple avec session_set_save_handler
◊ Set server controls
◊ Exemple avec set_file_buffer
◊ Exemple avec setcookie
◊ Exemple avec setlocale
◊ Exemple avec shuffle
◊ Simple stream example that displays the php home page
◊ Exemple avec socket_strerror
◊ Exemple avec sort
◊ spécifier le chemin de la catégorie courante avec le format suivant : '> root > sport > foot > psg'
◊ Exemple avec split
◊ sprintf : format de monétaire
◊ Exemple avec sql_regcase
◊ Exemple avec sscanf : utilisation des paramètres optionnels
◊ Stocker des données sur un serveur distant
◊ Exemple avec str_pad
◊ Exemple avec str_repeat
◊ Exemple avec str_replace
◊ Exemple avec strcasecmp
◊ Exemple avec stream_set_timeout
◊ Exemple avec stream_set_write_buffer
◊ Exemple avec streammp3.php">swfmovie−>streammp3
◊ Exemple avec strip_tags
◊ Exemple avec stristr
◊ Exemple avec strrchr
◊ Exemple avec strstr
◊ strtolower example
◊ Exemple avec strtotime
◊ Exemple avec strtoupper example
◊ Exemple avec strtr
◊ Exemple avec substr_count
◊ Exemple avec substr_replace
◊ Suppression d'un attribut
◊ exemple avec swf_addbuttonrecord
◊ Exemple avec swfaction
◊ Exemple avec swfbitmap
◊ Exemple avec swfgradient
◊ Exemple avec swfmorph
◊ Exemple avec swfshape

♦ **T**

◊ Tableau d'index commençant à 1
◊ Tableaux multi−dimensionnels, et récursifs
◊ Exemple avec tempnam
◊ Test_aggregation.php
◊ Test_script.php
◊ Exemple avec touch
◊ traitement des erreurs avec set_error_handler et trigger_error
◊ Translation
◊ Types mysql field
♦ **U**
◊ Exemple avec ucfirst
◊ Exemple avec ucwords
◊ Exemple avec uksort
◊ Un exemple simple de liste retournée par php_ini_scanned_files
◊ un moyen pratique pour remplacer opendir par glob
◊ Exemple avec unpack
◊ Exemple avec unserialize
◊ Exemple avec unset
◊ Exemple avec urldecode
◊ exemple avec urlencode et htmlentities
◊ Users.txt
◊ Exemple avec usort
◊ Utilisation d'une connexion udp
◊ Utilisation dangereuse de variables
◊ Utilisation de syslog
◊ Utilisation de xslt_process pour transformer trois
◊ Utilisation de curl et php pour récupérer une page
◊ Utilisation de fonctions anonymes comme fonction de callback
◊ Utilisation de paquets incrémentaux
◊ Utilisation de procédures stockées
◊ Utilisation des accolades {} dans les chaînes pour les tableaux
◊ Utilisation des constantes __file__ et __line__
◊ Utilisation des objets de grande taille (large objects)
◊ Utiliser dbase_numfields
◊ Utiliser le type html pour les extensions php
◊ Utiliser un ref cursor issue d'une commande select
◊ Utiliser une valeur négative de length
♦ **V**
◊ Vérification gettext
◊ Vérification de la version du protocole avec ldap_get_option
◊ Vérification de noms de fichier sécurisée
◊ Vérifier le nom d'une couleur avec un masque shell
◊ Validation de fichiers téléchargés
◊ Valider une transaction sesam
◊ Variables complexes de formulaire
◊ Exemple avec version_compare
♦ **W**
◊ wddx_serialize_vars
◊ Windows include_path
◊ Exemple avec wordwrap
♦ **X**
◊ Xml transtypage xml −> html
◊ Xmltest2.xml
♦ **Y**

♦ ◊ Exemple avec yp_first
◊ Exemple avec yp_next