

Guide d'installation de la bibliothèque PEAR

par Jérôme CHABAUD ([site](#))

Date de publication : 23 avril 2004

Dernière mise à jour :

Cet article expose la méthode d'installation manuelle de la librairie PEAR pour PHP

- I - Avant-propos
 - I-A - Pré-requis
 - I-B - Public
 - I-C - Pourquoi ce document ?
- II - PEAR
 - II-A - Qu'est-ce que PEAR (succinctement)?
 - II-B - Pourquoi une installation manuelle ?
 - II-C - Notion de package et de dépendances.
 - II-D - Où télécharger ?
- III - Assez parlé, passons à l'action !
 - III-A - Arborescence
 - III-B - Installation du package DB
- IV - include_path
- V - Notion de N#uds
- VI - Conclusion

I - Avant-propos

I-A - Pré-requis

Connaître les bases du langage PHP, avoir un serveur HTTP avec PHP 4.3.0 (ou supérieur) et MySQL (optionnel : uniquement pour faire des tests). Pour l'utilisateur de Windows, EasyPHP1.7 ira très bien.

I-B - Public

Webmasters, développeurs php désireux d'utiliser la bibliothèque PEAR.

I-C - Pourquoi ce document ?

Parce que je me suis rendu compte que les questions sur l'installation de PEAR étaient récurrentes sur le forum PHP de developpez.com.

II - PEAR

II-A - Qu'est-ce que PEAR (succinctement)?

PEAR (pour *PHP Extension and Application Repository*) est une bibliothèque de scripts PHP. Tous les scripts déposés dans PEAR respectent un certain nombre de règles qui les rend portables (indépendants de la plate-forme) et réutilisables.

II-B - Pourquoi une installation manuelle ?

La question est légitime puisqu'il existe un "installer" !

La réponse est simple : en installant soi-même les scripts on comprend déjà un peu le fonctionnement. Et puis on installe ce que l'on veut, à la version désirée.

De plus, si vous hébergez votre site sur un serveur mutualisé, vous ne pourrez pas utiliser l'"installer".

II-C - Notion de package et de dépendances.

La bibliothèque est composée de packages eux-mêmes composés de scripts.

Chaque package constitue une fonctionnalité. Par exemple le package DB correspond à la fonctionnalité : accès aux bases de données (couche d'abstraction).

Chaque package est susceptible d'en appeler un autre ou d'être appelé par un autre. D'autre part certains packages utilisent les fonctionnalités d'autres packages. On parle alors de dépendance.

Par exemple le package **HTML_QuickForm** est dépendant du package **Html_common**. Ceci veut dire que pour fonctionner correctement le package **Html_quickform** a besoin que **HTML_common** soit installé.

Nota: certaines dépendances sont optionnelles. Dans ce cas, seules les fonctionnalités qui ne font pas appel aux dépendances seront utilisables.

II-D - Où télécharger ?

Les packages sont téléchargeables à partir de la page : <http://pear.php.net/packages.php>

Chaque package a une "page d'accueil" sur laquelle on a les informations suivantes :

- Un résumé, le ou les auteurs, le type de licence et une description du package.
- Une série de liens notamment celui permettant le téléchargement du package. [download latest]
- Les différentes versions disponibles au téléchargement.
- Les dépendances du package en question.

III - Assez parlé, passons à l'action !

Je vous propose pour illustrer ce qui vient d'être dit, d'installer notre premier package - **DB** - qui permet d'accéder à différents type de base de données (couche d'abstraction).

Avant de commencer l'installation proprement dite, je vous conseille de supprimer les versions de PEAR installées par défaut surtout si vous utilisez EasyPhp. Pour cela, faites une recherche du répertoire "PEAR" et supprimez-le (ou renommez-le).

Procédons par étapes :

III-A - Arborescence

Il faut d'abord définir le répertoire dans lequel seront installés les packages.

Ce répertoire peut être en dehors de l'arborescence accessible par le serveur HTTP. Par exemple `c:\php\pear` si vous êtes sous Windows ou `/home/php/pear` pour Linux.

Pour plus de facilité, nous appellerons ce répertoire **PEARDIR** par la suite.

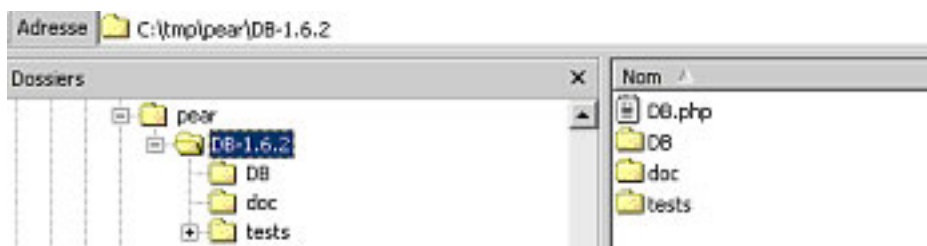
III-B - Installation du package DB

Sur la page <http://pear.php.net/packages.php>, choisir **Database** puis **DB**.

On clique sur [download latest]. Le fichier `DB-x.x.x.tgz` est alors téléchargé. (x.x.x représente la version).

Décompresser cette archive dans un répertoire temporaire (par exemple `c:\tmp` ou `/home/users/tmp`).

On a donc maintenant :



Il faut alors copier le contenu du répertoire `DB-x.x.x` dans **PEARDIR**. On aura donc :



On va pouvoir tester notre premier script utilisant PEAR. Pour cela nous utiliserons l'exemple de la page :

<http://pear.php.net/manual/en/package.database.db.intro-connect.php>

Script de test

```

<?php
$user = 'root';
$password = '';
$mysqlServer = 'localhost';
$dataBaseName = 'mabase';
require_once 'DB.php';
$dsn = 'mysql://'. $user . ':' . $password . '@' . $mysqlServer . '/' . $dataBaseName;
$db =& DB::connect($dsn);
if (DB::isError($db)) {
    die($db->getMessage());
}
echo 'la connexion a reussie' ;
$db->disconnect();
?>

```

Bien sûr, vous modifierez les lignes 2 à 5 afin quelles correspondent à vos paramètres de connexion.

Et là oh surprise : vous avez 2 belles erreurs !!

```

Warning: main(DB.php): failed to open stream: No such file or directory in
C:\data\www\testPear\testDB.php on line 2

Fatal error: main(): Failed opening required 'DB.php' (include_path='.;c:\php\includes')
in C:\data\www\testPear\testDB.php on line 2

```

Le script que nous avons écrit ne trouve pas le fichier **DB.php**. C'est normal puisque nous n'avons pas dit où se situaient les scripts des packages PEAR.

Pour résoudre ceci, nous allons utiliser la fonction `set_include_path()`.

IV - include_path

Lorsque dans un script on utilise une des fonctions `include`, `require`, `require_once`, les fichiers passés en argument sont cherchés dans les répertoires définis comme «`include_path`». Pour connaître ces répertoires, il suffit de faire un `phpinfo()` ou d'utiliser la fonction `get_include_path()`.

Il faut donc que l'on ajoute le répertoire **PEARDIR** à ces **include_path**. Pour cela, on va utiliser la fonction `set_include_path()` de la façon suivante :

Pour Windows

```
set_include_path(get_include_path() . ";c:\php\pear");
```

Pour Linux

```
set_include_path(get_include_path() . ":/home/php/pear");
```

Nota: il est possible de modifier la configuration de PHP pour inclure automatiquement les répertoires voulus, mais ceci est souvent (toujours) impossible sur les serveurs mutualisés.

Votre code ne fonctionnerait alors pas sur ce type de serveur.

Une fois ajoutée cette ligne au début de notre script, on a... de nouveau 2 belles erreurs

```
Warning: main(PEAR.php): failed to open stream: No such file or directory in c:\php\pear\DB.php on line 26

Fatal error: main(): Failed opening required 'PEAR.php'
(include_path='.;c:\php\includes;c:\php\pear')
in c:\php\pear\DB.php on line 26
```

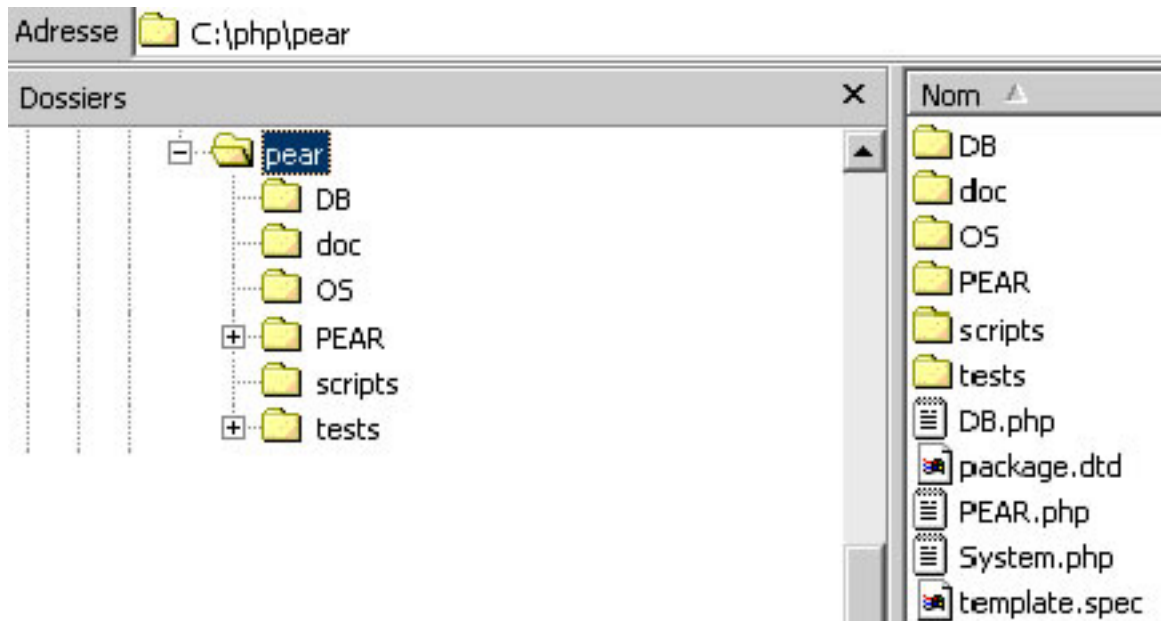
En effet le package **DB** a une dépendance au package **PEAR**: il faut donc installer le package **PEAR**. Au passage on voit bien que les `include_path` ont été modifiés.

Je vous laisse donc installer le package PEAR qui est disponible à l'adresse : <http://pear.php.net/package/PEAR>

Pour rappel :

- 1 Téléchargement
- 2 Décompression dans un répertoire temporaire
- 3 Copie du contenu du package dans **PEARDIR**

Votre répertoire **PEARDIR** doit maintenant ressembler à ça :



Relancer l'exécution de notre script de test et là... **plus d'erreur** !

V - Notion de N#uds

Dans l'arborescence du répertoire **PEAR** on distingue des n#uds. Ces n#uds sont identifiables dans le nom du package car ils sont séparés par des underscores. Par exemple le package **DB_DataObject** fait partie du n#ud **DB** que l'on vient de créer. Pour installer ce package il faut donc copier son contenu (après décompression) non pas dans le répertoire **PEARDIR** comme précédemment mais dans **PEARDIR/DB**.

Les n#uds de niveau le plus élevé sont identifiables par l'absence d'underscore dans le nom du package. C'est le cas du package **DB** que nous avons installé précédemment.

Certains packages font partie de n#uds qui n'ont pas de package propre. Je m'explique avec l'exemple du n#ud **HTML** : il existe un package **HTML_QuickForm** qui doit donc être installé dans le répertoire **PEARDIR/HTML**. Mais comme il n'existe pas de package **HTML**, il faut créer le répertoire **HTML** « à la main ».

Nota: Les noms des n#uds sont toujours en majuscule.

VI - Conclusion

Toutes ces manipulations vous semblent peut-être compliquées, fastidieuses, voire inutiles, mais je vous assure qu'installer un package prend moins d'une minute quand on a l'habitude.

