

# Exécuter plusieurs instances d'Apache et migrer de PHP 4 à PHP 5



par Arnold Daniels Guillaume Rossolini (traducteur) ([Tutoriels Web](#))

Date de publication : 5 septembre 2007

Dernière mise à jour :


Comment exécuter en parallèle deux instances de la même version d'Apache avec deux versions différentes de PHP ? Arnold Daniels vous explique la procédure en détail pour un serveur sous Linux.

Il passe ensuite à un exemple de méthode pour migrer de PHP 4 à PHP 5.


I - Introduction.....	3
I-A - Remerciements.....	3
I-B - Préambule.....	3
II - Compiler PHP.....	3
III - Préparer des fichiers additionnels de configuration pour Apache.....	4
IV - Modifier les scripts du système.....	4
V - Migrer depuis PHP 4 vers PHP 5.....	5
VI - Conclusion.....	6
II-A - Épilogue.....	6
II-B - Liens.....	6

## I - Introduction

### I-A - Remerciements

Tous mes (G. Rossolini) remerciements vont naturellement à **Arnold Daniels** pour avoir mis en ligne le tutoriel original :  **Running multiple instances of Apache.**

### I-B - Préambule

J'ai choisi de traduire ce tutoriel car il présente une alternative très intéressante à ce que j'ai présenté dans " **Alterner entre plusieurs versions d'Apache et de PHP**". Ces deux articles sont en fait complémentaires, car on peut ainsi tester le comportement d'applications PHP sous diverses versions d'Apache et de PHP, ce qui est très pratique dans certaines situations.

Arnold introduit son article ainsi :

La vie d'un hébergeur mutualisé n'a pas été simplifiée par la décision du *PHP Group* de briser la compatibilité entre les versions majeures. Certains hébergeurs ont pu contenter leurs clients en conservant uniquement PHP 4, mais de nombreux projets sont spécifiques à PHP 5 et le problème prendra de l'envergure après la sortie de PHP 6.

Habituellement, un *daemon* (ou *service* sous Windows) est démarré une fois, avec une seule instance en exécution. Il en va de même pour Apache. Pourtant, lorsque nous regardons de près la configuration d'Apache, nous voyons que des variables qui empêchent Apache d'être démarré plusieurs fois (fichier PID, lockfile et port/IP) peuvent être configurés. Lorsque l'on démarre Apache, le fichier de configuration peut être spécifié, ce qui permet d'exécuter plusieurs instances d'Apache (sur plusieurs IPs).

Ainsi, il nous suffit de changer l'adresse IP d'un domaine dans le DNS pour changer de PHP 4 à PHP 5. La même astuce peut être utilisée pour changer facilement le plan d'un compte depuis "basique" (uniquement des pages statiques HTML) à "gold" (support de PHP et Perl) à "platinum" (support de PHP, Perl, Ruby et JSP) sans être contraints de déplacer le compte sur un autre serveur.

Revenons à PHP. Nous allons voir les étapes permettant de configurer un serveur pour exécuter PHP 4 et PHP 5 côte à côte.

 *Ce tutoriel part du principe que vous utilisez un système Linux Ubuntu ou Debian.*

## II - Compiler PHP

Nous devons tout d'abord compiler PHP 4 et PHP 5 de manière à ce que les chemins et noms de fichiers ne soient pas en conflit. Je ne vais pas entrer dans les détails, mais il faudra ajouter ceci pour PHP 5 :

```
./configure
'--libdir=/usr/lib/php5' \
'--includedir=/usr/include/php5' \
'--program-suffix=5' \
'--with-exec-dir=/usr/lib/php5/libexec' \
'--with-config-file-path=/etc/php5/'
```

### III - Préparer des fichiers additionnels de configuration pour Apache

Puisque toutes les instances auront plus ou moins la même configuration, nous allons inclure **apache2.conf** pour la configuration commune et utiliser une macro pour les différences. Si vous n'avez pas encore installé **mod\_macro**, veuillez l'installer et l'activer :

```
apt-get install libapache2-mod-macro
a2enmod macro
```

Nous devons spécifier l'adresse IP par instance, il faut donc supprimer le contenu de **/etc/apache2/ports.conf**. Supprimez également les liens symboliques de tous les modules qui ne devraient pas être chargés dans toutes les instances (comme **php5\_module**) du fichier **/etc/apache2/mods-enabled**.

Créez un fichier **/etc/apache2/apache2-instance.conf** et saisissez votre configuration par instance, par exemple :

```
/etc/apache2/apache2-instance.conf
<Macro SetupInstance $instance $ip>
LockFile /var/lock/apache2/$instance.accept.lock
PidFile /var/run/apache2/$instance.pid
ErrorLog /var/log/apache2/$instance/error.log
Listen $ip:80
</Macro>
```

Créez ensuite :

```
apache2-php4.conf
Include /etc/apache2/apache2.conf
Include /etc/apache2/apache2-instance.conf
Use SetupInstance "apache2-php4" "192.168.1.50"

LoadModule php4_module /usr/lib/apache2/modules/libphp4.so
<IfModule mod_php4.c>
AddType application/x-httpd-php .php .phtml .php3
AddType application/x-httpd-php-source .phps
</IfModule>
```

Faites de même pour **apache2-php5.conf**. N'oubliez pas de créer également **/var/log/apache2/apache2-php4** ainsi que **/var/log/apache2/apache2-php5**.

### IV - Modifier les scripts du système

Lorsque vous exécutez "apache2ctl start", le script affiche *no listening sockets available, shutting down* et se termine. C'est normal, bien que nous ayons besoin de démarrer Apache. Nous allons donc modifier le script.

Copiez **/usr/sbin/apache2ctl** vers **/usr/sbin/apache2-php4ctl** et **/usr/sbin/apache2-php5ctl**. Éditez **apache2-php4ctl** pour changer "HTTPD" à :

```
apache2-php4ctl
HTTPD='/usr/sbin/apache2 -f /etc/apache2/apache2-php4/apache2.conf'
```

Faites de même pour **apache2-php5**.

Un autre script a besoin d'être modifié, il est situé dans `/etc/init.d` : copiez `/etc/init.d/apache2` vers `/etc/init.d/apache2-php4` et `/etc/init.d/apache2-php5` puis déplacez le fichier vers `/etc/init.d/apache2.org` et modifiez alors `/etc/init.d/apache2-php4` :

```
/etc/init.d/apache2-php4
APACHE2CTL="$ENV /usr/sbin/apache2-php4ctl"
```

Le script essaie de trouver le PID en cherchant le nom du fichier PID dans la configuration, mais cela ne fonctionnera pas puisque nous utilisons une macro. Ainsi, utilisez ce code :


```
/etc/init.d/apache2-php4
PIDFILE='/var/run/apache2/apache2-php4.pid'
if [ -e "$PIDFILE" ] ; then
PID=`cat "$PIDFILE"`
fi
```

Mettez tout en commentaires depuis `"for i in $AP_CONF"` jusqu'à `"done"`. Appliquez les mêmes modifications à `/etc/init.d/apache2-php5`.

Enfin, créez un nouveau script `/etc/init.d/apache2` (attention aux permissions) avec le contenu suivant :

```
/etc/init.d/apache2
#!/bin/sh
./apache2-php4 $@
./apache2-php5 $@
```

## V - Migrer depuis PHP 4 vers PHP 5

Si vous avez une entreprise d'hébergement Web et que vous avez toujours PHP 4, vous sentez probablement monter la pression. Des articles tels que " *Now showing: PHP's true colors*" vous enjoignent à basculer et à encaisser le coup, mais ils ne vous donnent pas un seul conseil sur la manière de procéder.

Le problème est que vous pouvez informer vos clients que vous migrez vers PHP 5, mais si vous ne pouvez pas leur fournir une plate forme de test, comment pourront-ils savoir si la migration va casser leurs sites ?

Il y a quelques mois, j'ai posté un article sur comment exécuter plusieurs instances d'Apache avec différentes versions de PHP [*NDT : traduit plus haut dans cette page*]. Dans cet article, j'ai montré comment créer diverses configurations exécutant Apache sur différentes adresses IP. En utilisant différents ports plutôt que différentes adresses IP, nous pouvons donner un moyen à nos clients de vérifier leurs sites avant de mettre en production.

Si vous n'avez pas lu l'article, commencez par là [*NDT : lisez le début de ce tutoriel*]. Lors de l'étape **"Préparer des fichiers additionnels de configuration"** cependant, vous devrez le faire légèrement différemment. Suivez l'article complet, mais à cette étape faites ceci (sans utiliser `mod_macro`) :

Retirez PHP4 de `mods_enabled` ainsi que la directive "Listen" du fichier `apache.conf`. Créez ensuite :

```
apache2-php4.conf
Include /etc/apache2/apache2.conf
Listen 192.168.1.50:80

LoadModule php4_module /usr/lib/apache2/modules/libphp4.so
<IfModule mod_php4.c>
  AddType application/x-httpd-php .php .phtml .php3
  AddType application/x-httpd-php-source .phps
```

**apache2-php4.conf**

```
</IfModule>
```

**apache2-php5.conf**

```
Include /etc/apache2/apache2.conf

LockFile /var/lock/apache2/apache2-php5.accept.lock
PidFile /var/run/apache2-php5.pid
ErrorLog /var/log/apache2/error-php5.log
Listen 192.168.1.50:8800

LoadModule php5_module /usr/lib/apache2/modules/libphp5.so
<IfModule mod_php5.c>
  AddType application/x-httpd-php .php .phtml .php3
  AddType application/x-httpd-php-source .phps
</IfModule>
```

Dans cet exemple, remplacez "192.168.1.50" par votre propre adresse IP. Redémarrez Apache et il lancera 2 instances, l'une avec PHP 4 sur le port 80 et l'autre avec PHP 5 sur le port 8800. Vous pouvez maintenant envoyer un message à tous vos clients en leur annonçant que vous migrerez vers PHP 5 sous un certain nombre de semaines et qu'ils devraient vérifier que leur site n'est pas sujet à des problèmes en se rendant à l'adresse <http://www.example.com:8800/>, où example.com doit être remplacé par leur propre nom de domaine.

 *L'exemple "Migrer depuis PHP 4 vers PHP 5" n'est pas testé, merci de me faire savoir si cela ne fonctionne pas.*

**VI - Conclusion****II-A - Épilogue**

Redémarrez simplement le serveur et vous êtes prêt. Vous aurez sans doute besoin d'adapter un peu cet exemple pour votre propre configuration, mais de toute manière j'espère que cela vous aura donné des idées.

**II-B - Liens**

-  [Cours, tutoriels, FAQ, livres et forums sur Linux](#)