

# Cohabitation PHP4 et PHP5

par [ska\\_r00t](#)

Date de publication : 03/10/2005

Dernière mise à jour : 03/10/2005

Ce tutoriel a pour objectif de vous aider à installer PHP4 et PHP5 sur une même machine. Cette manipulation a été testée sur Fedora Core4 , Apache2.0, PHP4.4.0, PHP5.04

- I - Introduction
  - I.A - Pourquoi ?
  - I.B - Solution CGI
  - I.C - Solution SAPI
  - I.D - Conclusion
- II - Pré-requis
- III - Préparation
- IV - Apache
- V - PHP
- VI - Démarrage
- VII - Tests

[Télécharger l'article au format PDF \(288ko\) \[ miroir \]](#)

## I - Introduction

Avant de suivre ce tutoriel, il convient de faire le point sur votre besoin. En effet, il existe plusieurs procédés pour arriver à utiliser des scripts PHP4 et PHP5 sur la même machine.

### I.A - Pourquoi ?

Afin de mieux comprendre quelle peut être votre motivation, je vous propose de vous expliquer pourquoi j'ai eu besoin de le faire de cette manière.

Tout commence dans mon milieu professionnel, lorsqu'on me demande de créer un serveur avec un profil test. Celui-ci devra pouvoir héberger le fruit du travail de deux équipes de développeurs web. Ces applications s'appuyant sur une méthodologie objet, l'une avec le moteur PHP4 et l'autre avec PHP5. Ce qui traduit une incompatibilité évidente entre les deux types d'applicatif.

Le système installé sur la machine étant Fedora Core 4, avec installation par défaut (station de travail), nous avons déjà Apache 2.0 ainsi que PHP5 configuré pour être module dynamique d'Apache. Il nous suffit donc d'installer PHP4, reste à savoir comment. Nous avons deux solutions, soit on configure 2 instances d'Apache avec chacune leur module dynamique PHP (SAPI), soit nous ajoutons PHP4 en tant que module statique (CGI).

### I.B - Solution CGI

D'un point de vue fonctionnel, la deuxième solution n'est pas pratique car il faudrait donner un moyen au serveur HTTP de distinguer les différentes applications et ainsi diriger les requêtes dynamiques vers le moteur PHP approprié. C'est à dire qu'il faudrait que tous les fichiers aient l'extension .php4 ou alors créer un addHandler propre au répertoire hébergeant l'application PHP4. Or, le projet existe déjà depuis un bon moment, et modifier les noms de fichiers dans l'arborescence ainsi que tous les liens utilisés relève du défi. Et dans un souci d'homogénéité, je préfère avoir deux répertoires prêts au déploiement en production. Donc pas de manipulation directe sur un htaccess.

De plus je peux ainsi laisser le choix à une équipe ou l'autre de mettre à jour la version stable quand ils le veulent, un ZIP du répertoire et hop, c'est fait... Toutefois, si tel est votre choix, il vous faudra créer un fichier .htaccess dans le répertoire web concerné, de manière à indiquer à Apache, ceci grâce à la directive addHandler quel moteur est censé travailler avec les fichiers portant l'extension PHP. Ou encore, comme dit précédemment ajouter le addHandler directement dans la conf HTTP et suffixer les fichiers de l'extension correspondante ( ex: \*.php4 ).

### I.C - Solution SAPI

La solution que j'ai retenue pour mon projet est donc de créer deux instances d'Apache, avec pour chacune d'entre elles, un moteur PHP différent basé sur un module dynamique. Ainsi nous aurons deux répertoires web distincts avec des fichiers ayant pour extension \*.php. Il suffira ensuite, dans un esprit développement, je le répète, de faire pointer deux domaines sur ces répertoires, un sur Apache port 80 et un sur Apache port 81.

### I.D - Conclusion

La solution d'employer deux instances du processus httpd, s'avère être plus gourmande en terme de ressources système que l'emploi d'un mode CGI pour PHP4, même si l'on considère le fait que le CGI engendre un processus

pour chaque requête vers le moteur PHP. Elle sera donc réservée à un serveur de type "Développement", entendez par là : non exploitable en production. Si votre but est de tester la compatibilité d'une application sous les deux versions de moteur PHP, peut-être gagnerez-vous en temps à changer les lignes concernées dans le fichier de configuration d'Apache et de re-démarrer le serveur HTTP.

## II - Pré-requis

- Un système d'exploitation Linux
- Un serveur HTTP
- Un moteur PHP5

### III - Préparation

Nous allons commencer par nous munir des éléments nécessaires pour la suite de ce tutoriel.

Vous trouverez les sources PHP4 à cette adresse : <http://www.php.net>

Décompressez l'archive dans le répertoire de votre choix.

```
$ tar xzvf php4.x.x.tar.gz
```

Nous allons à présent créer les répertoires. Loguez vous en tant que super-utilisateur ( root ) et tapez ceci

```
# mkdir /var/www/html2 /var/www/cgi-bin2
# chown -R apache:apache /var/www/html2 /var/www/cgi-bin2
# chmod -R 755 /var/www/html2 /var/www/cgi-bin2
# cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd2.conf
# cp -R /etc/httpd/conf.d /etc/httpd/conf2.d
```

## IV - Apache

Commençons par arrêter le démon httpd

```
# /etc/init.d/httpd stop
```

Puis éditons le fichier de configuration précédemment créé **/etc/httpd/conf/httpd2.conf**.

Il faut modifier le pid utilisé par httpd, le port écouté, le documentRoot et les fichiers de log.

Recherchez les lignes suivantes et modifiez-les comme indiqué.

```
PidFile run/httpd.pid => PidFile run/httpd2.pid
.....
Listen 80 => Listen 81
.....
Include conf.d/*.conf => Include conf2.d/*.conf
.....
DocumentRoot "/var/www/html" => DocumentRoot "/var/www/html2"
.....
<Directory "/var/www/html"> => <Directory "/var/www/html2">
.....
ErrorLog logs/error_log => ErrorLog logs/error2_log
.....
CustomLog logs/access_log combined => CustomLog logs/access2_log combined
.....
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/" => ScriptAlias /cgi-bin/ "/var/www/cgi-bin2/"
.....
<Directory "/var/www/cgi-bin"> => <Directory "/var/www/cgi-bin2">
```

Éditons maintenant le fichier inclus **/etc/httpd/conf2.d/php.conf**. Effacez tout le contenu et mettez les lignes ci-dessous

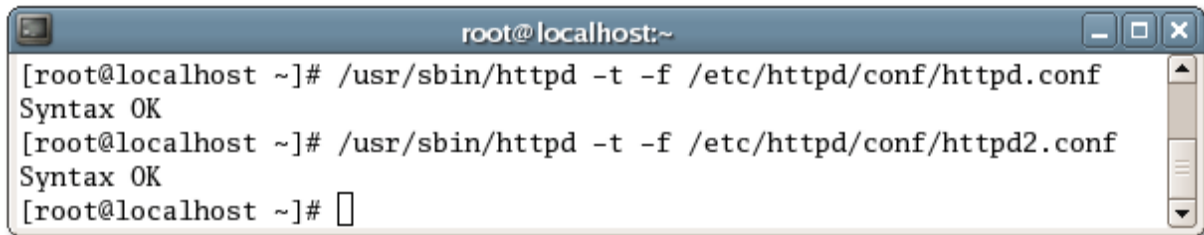
```
LoadModule php4_module /usr/lib/httpd/modules/libphp4.so
AddType application/x-httpd-php .php .phtml
DirectoryIndex index.php
AddType application/x-httpd-php-source .phps
```

Puis le fichier inclus **/etc/httpd/conf2.d/ssl.conf**, il ne nous est pas directement nécessaire, mais si nous ne le faisons pas, Apache lors du démarrage nous signalera une incohérence du service SSL (en fait un doublon de l'hôte avec la première instance).

```
Listen 443 => Listen 444
.....
<VirtualHost _default_:443> => <VirtualHost _default_:444>
.....
ErrorLog logs/ssl_error_log => ErrorLog logs/ssl_error2_log
TransferLog logs/ssl_access_log => TransferLog logs/ssl_access2_log
```

Nous pouvons désormais tester la syntaxe de nos fichiers de configuration

```
# /usr/sbin/httpd -t -f /etc/httpd/conf/httpd.conf
# /usr/sbin/httpd -t -f /etc/httpd/conf/httpd2.conf
```

A terminal window titled "root@localhost:~" with standard window controls. The terminal shows three lines of command execution: 1. "[root@localhost ~]# /usr/sbin/httpd -t -f /etc/httpd/conf/httpd.conf" followed by "Syntax OK". 2. "[root@localhost ~]# /usr/sbin/httpd -t -f /etc/httpd/conf/httpd2.conf" followed by "Syntax OK". 3. "[root@localhost ~]# " followed by a cursor. A vertical scrollbar is visible on the right side of the terminal window.

```
root@localhost:~  
[root@localhost ~]# /usr/sbin/httpd -t -f /etc/httpd/conf/httpd.conf  
Syntax OK  
[root@localhost ~]# /usr/sbin/httpd -t -f /etc/httpd/conf/httpd2.conf  
Syntax OK  
[root@localhost ~]# █
```

*test des fichiers de configuration*



## V - PHP

Passons à l'installation de PHP4, placez vous dans le répertoire que nous avons décompressé tout à l'heure.

Pour éviter les conflits avec la version installée par défaut dans Fedora, nous allons installer les binaires dans un répertoire propre (exemple **/usr/local/php4**) à l'aide du paramètre **--prefix** puis nous mettrons le php.ini avec le paramètre **--config-file-path** dans un repertoire autre que /etc où se trouve celui de PHP5. Je choisis pour l'exemple le répertoire conf d'Apache (**/etc/httpd/conf**).

L'option **--with-gd** nous permettra d'utiliser la librairie graphique de php, celle-ci nécessite la dépendance zlib ce que nous indiquerons grâce au paramètre **--with-zlib**

Il nous reste plus qu'à inclure l'option **--with-apxs2** indiquant au compilateur que PHP sera utilisé en tant que module dynamique d'Apache2.

```
$ ./configure --enable-track-vars --with-mysql --with-apxs2=/usr/sbin/apxs \  
\  
--prefix=/usr/local/php4 --with-config-file-path=/etc/httpd/conf \  
\  
--with-zlib --with-gd  
$ make
```

Puis en tant que root :

```
# make install
```

Copier le fichier **php.ini-dist** situé dans le repertoire d'installation vers le repertoire choisi avec l'option **--config-file-path** en prenant soin de le renommer en **php.ini**.

```
# cp /home/toto/php4.4.0/php.ini-dist /etc/httpd/conf/php.ini
```

## VI - Démarrage

Créons maintenant le script qui nous permettra de démarrer l'instance Apache que nous avons configurée.

Ouvrez votre éditeur préféré et copiez-collez le script ci-dessous

```
#!/bin/bash
#
# httpd-PHP4          Startup script for the Apache HTTP Server Instance 1
#
# chkconfig: - 86 14
# description: Apache is a World Wide Web server.  It is used to serve \
#              HTML files and CGI
# processname: httpd
# config: /etc/httpd/conf/httpd2.conf
# config: /etc/sysconfig/httpd
# pidfile: /var/run/httpd2.pid

# Source function library.
. /etc/rc.d/init.d/functions

httpd=${HTTPD-/usr/sbin/httpd}
prog=httpd-PHP4
conf=/etc/httpd/conf/httpd2.conf
pidfile=${PIDFILE-/var/run/httpd2.pid}
lockfile=${LOCKFILE-/var/lock/subsys/httpd2}
RETVAL=0

start() {
    $httpd -f $conf -k start
    ret=$?
    if [ $ret = 0 ] && touch ${lockfile}; then
        # si tout ok
        action $"Démarrage de $prog: " /bin/true
    else
        # sinon
        action $"Démarrage de $prog: " /bin/false
    fi
    return $ret
}

stop() {
    $httpd -f $conf -k stop
    ret=$?
    if [ $ret = 0 ] && rm -f ${lockfile} ${pidfile}; then
        # si arret ok
        action $"Arrêt de $prog: " /bin/true
    else
        # sinon
        action $"Arrêt de $prog: " /bin/false
    fi
    return $ret
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status $httpd
        ret=$?
        ;;
    restart)
        stop
        start
        ;;
    *)
        echo $"Usage: $prog {start|stop|restart|status}"
        exit 1
esac
```

```
exit $ret
```

Enregistrez votre fichier dans **/etc/init.d/httpd2** Il faut maintenant lui donner les droits d'exécution et configurer le system V de manière à ce que le script prenne effet dès le boot de la machine.

```
# chmod +x /etc/init.d/httpd2
# chkconfig --add httpd2
# chkconfig --level 345 httpd2 on
```

## VII - Tests

Voici arrivé le moment tant attendu, les tests de notre nouvelle configuration. Editez un fichier que vous nommerez **index.php**

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

Enregistrez votre fichier dans le répertoire principal d'Apache (/var/www/html), puis copiez-le dans le répertoire principal de la seconde instance (/var/www/html2).

```
# cp /var/www/html/index.php /var/www/html2/index.php
```

Passons aux choses sérieuses...Pour démarrer les deux instances d'Apache :

```
# /etc/init.d/httpd start
# /etc/init.d/httpd2 start
```

Pour vérifier

```
# ps aux | grep httpd
```

Nous donnera quelque chose comme ceci

```

root@localhost:~
[root@localhost ~]# /etc/init.d/httpd start
Démarrage de httpd : [ OK ]
[root@localhost ~]# /etc/init.d/httpd2 start
Démarrage de httpd-PHP4: [ OK ]
[root@localhost ~]# ps aux | grep httpd
root      4955  3.6  2.8 32912 14496 ?        Ss   19:13   0:00 /usr/sbin/httpd
apache    4958  0.0  2.8 32912 14508 ?        S    19:13   0:00 /usr/sbin/httpd
apache    4959  0.0  2.8 32912 14508 ?        S    19:13   0:00 /usr/sbin/httpd
apache    4960  0.0  2.8 32912 14508 ?        S    19:13   0:00 /usr/sbin/httpd
apache    4961  0.0  2.8 32912 14508 ?        S    19:13   0:00 /usr/sbin/httpd
apache    4962  0.0  2.8 32912 14508 ?        S    19:13   0:00 /usr/sbin/httpd
apache    4963  0.0  2.8 32912 14508 ?        S    19:13   0:00 /usr/sbin/httpd
apache    4964  0.0  2.8 32912 14508 ?        S    19:13   0:00 /usr/sbin/httpd
apache    4965  0.0  2.8 32912 14508 ?        S    19:13   0:00 /usr/sbin/httpd
root      4971  2.1  2.0 20360 10536 ?        Ss   19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
apache    4974  0.0  2.0 20360 10560 ?        S    19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
apache    4975  0.0  2.0 20360 10560 ?        S    19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
apache    4976  0.0  2.0 20360 10560 ?        S    19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
apache    4977  0.0  2.0 20360 10560 ?        S    19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
apache    4978  0.0  2.0 20360 10560 ?        S    19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
apache    4979  0.0  2.0 20360 10560 ?        S    19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
apache    4980  0.0  2.0 20360 10560 ?        S    19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
apache    4981  0.0  2.0 20360 10560 ?        S    19:14   0:00 /usr/sbin/httpd -f /etc/htt
pd/conf/httpd2.conf -k start
root      4984  0.0  0.1  3796   780 pts/2    S+   19:14   0:00 grep httpd
[root@localhost ~]# /etc/init.d/httpd stop
Arrêt de httpd : [ OK ]
[root@localhost ~]# /etc/init.d/httpd2 stop
Arrêt de httpd-PHP4: [ OK ]
[root@localhost ~]# ps aux | grep httpd
root      5003  0.0  0.1  3792   760 pts/2    R+   19:14   0:00 grep httpd
[root@localhost ~]# █

```

### tests apache

Puis ouvrons notre navigateur sur **http://localhost** et **http://localhost:81** pour vérifier que nous avons bien pris en compte les deux versions de PHP

phpinfo() - Mozilla Firefox

Fichier Edition Affichage Aller à Marque-pages Outils Aide

http://localhost/

## PHP Version 5.0.4

<b>System</b>	Linux localhost.localdomain 2.6.12-1.1398_FC4 #1 Fri Jul 15 00:52:32 EDT 2005 i686
<b>Build Date</b>	Jul 4 2005 10:40:52
<b>Configure Command</b>	'./configure' '--build=i386-redhat-linux' '--host=i386-redhat-linux' '--target=i386-redhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/usr/com' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=../config.cache' '--with-libdir=lib' '--with-config-file-path=/etc' '--with-config-file-scan-dir=/etc/php.d' '--disable-debug' '--with-pic' '--disable-rpath' '--with-bz2' '--with-curl' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--enable-gd-native-ttf' '--without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-png' '--with-pspell' '--with-expat-dir=/usr' '--with-pcre-regex=/usr' '--with-zlib' '--with-layout=GNU' '--enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-sockets' '--enable-syssem' '--enable-sysvshm' '--enable-sysvmsg' '--enable-track-vars' '--enable-trans-sid' '--enable-yp' '--enable-wddx' '--with-pear=/usr/share/pear' '--with-kerberos' '--enable-ucd-snmp-hack' '--with-unixODBC=shared,/usr' '--enable-memory-limit' '--enable-shmop' '--enable-calendar' '--enable-dbx' '--enable-dio' '--with-mime-magic=/usr/share/file/magic.mime' '--without-sqlite' '--with-libxml-dir=/usr' '--with-xml' '--with-apxs2=/usr/sbin/apxs' '--without-mysql' '--without-gd' '--without-odbc' '--disable-dom' '--disable-dba'
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php.d

PHP5

<b>System</b>	Linux localhost.localdomain 2.6.12-1.1398_FC4 #1 Fri Jul 15 00:52:32 EDT 2005 i686
<b>Build Date</b>	Oct 2 2005 23:15:00
<b>Configure Command</b>	'./configure' '--prefix=/usr/local/php4' '--with-config-file-path=/etc/httpd/conf' '--enable-track-vars' '--with-apxs2=/usr/sbin/apxs' '--with-mysql' '--with-zlib' '--with-gd'
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/httpd/conf/php.ini
<b>PHP API</b>	20020918
<b>PHP Extension</b>	20020429
<b>Zend Extension</b>	20050606
<b>Debug Build</b>	no
<b>Zend Memory Manager</b>	enabled
<b>Thread Safety</b>	disabled
<b>Registered PHP Streams</b>	php, http, ftp, compress.zlib

This program makes use of the Zend Scripting Language Engine:  
 Zend Engine v1.3.0, Copyright (c) 1998-2004 Zend Technologies



PHP4

Et voilà, nos équipes sont prêtes à travailler chacune de leur côté sur le serveur test.

Il ne reste plus qu'à mettre en place un serveur CVS pour faciliter le développement ;-)

Merci à [Maximilian](#) pour la relecture de cet article.