

Using PHP with mod_fcgid

BY MICHAEL STUCKI

The upcoming version 4.2 of TYPO3 will depend on PHP 5.2. Here is some help on migrating your server.

Index of Contents

Using PHP with mod_fcgid	1
Intended Audience.....	1
What is mod_fcgid.....	1
Advantages of mod_fcgid.....	1
Disadvantages of mod_fcgid.....	1
Installation	2
Prerequisites.....	2
Installation of packages.....	2
Configuration.....	2
Additional configuration.....	4
Appendix	5
Further reading.....	5
FAQ.....	5
ChangeLog.....	5
About the author.....	5

Intended Audience

This howto explains how to run PHP as a process which runs independently from Apache. It is written for server administrators who want to use a flexible and at the same time easy to manage PHP environment.

What is mod_fcgid

mod_fcgid is an Apache module. It allows execution of external programs who create web documents as their output. This procedure is well-known from CGI (Common Gateway Interface) which is often used in shared hosting environments, for example to run Perl scripts.

mod_fcgid was created as a binary compatible alternative to mod_fastcgi. Both extend the traditional CGI principle with the feature that they use persistent processes, managed by a server component. This means that the web server will not launch a new process for every request as it is the case with CGI. The speed gain is enormous.

Using PHP with mod_fcgid offers a full-featured PHP environment combined with all the features that mod_fcgid provides. This solution has a lot of advantages over the traditional way of running PHP on a web server, while the only disadvantage seems to be that it is a little bit more difficult to be set up.

Advantages of mod_fcgid

- Speed
mod_fcgid seems to be as fast as the traditional mod_php Apache module. However, it also allows to be run on a multi threaded Apache server. For various reasons this still does not work reliable with mod_php4 / mod_php5.
- Configuration flexibility
With mod_fcgid it is possible to run many different PHP versions on the same server, and even with multiple different users.

Disadvantages of mod_fcgid

- Installation is more complex
Installation of PHP using mod_fcgid is probably more difficult than setting up mod_php5, mainly because there is only a few documentation about it that can be found. However, since you found this document, you may be lucky...

Installation

Prerequisites

This manual is designed for Debian Etch servers. It expects a server running Apache 2.x. Basically this should work equally for other systems – they probably just use different paths and package names...

Installation of packages

The following packages are required:

- `php5-cgi`
This is the CGI binary of PHP5. It was compiled with FCGI support and works perfectly together with `mod_fcgid`.
- `libapache2-mod-fcgid`
This is the FCGID module
- `apache2-mpm-worker`
This is the multi threaded multi-processing module (MPM) for Apache2. It replaces `apache2-mpm-prefork` with a single-threaded MPM and is required by `mod_php4` / `mod_php5`. Of course these packages must also be removed.
Keep in mind that you may want to make a backup of your current `php.ini`, because `php5-cgi` will set up a new configuration file.

```
# apt-get -u install php5-cgi libapache2-mod-fcgid apache2-mpm-worker
```

Configuration

1. Remove `mod_php4` / `mod_php5` if not already done

```
# a2dismod php4  
# a2dismod php5
```

2. Enable `mod_actions` and `mod_fcgid`

```
# a2enmod actions  
# a2enmod fcgid
```

3. Raise the communication timeout (= maximum execution time) for FCGI applications in `/etc/apache2/mods-enabled/fcgid.conf` by adding the "IPCCommTimeout" directive

```
<IfModule mod_fcgid.c>  
    AddHandler fcgid-script .fcgi  
    SocketPath /var/lib/apache2/fcgid/sock  
  
    # Communication timeout: Default value is 20 seconds  
    IPCCommTimeout 60  
  
    # Connection timeout: Default value is 3 seconds  
  
    #IPCConnectTimeout 3  
</IfModule>
```

4. Create a new file /etc/apache2/conf.d/php-fcgid.conf:

```
<IfModule !mod_php4.c>
<IfModule !mod_php4_filter.c>
<IfModule !mod_php5.c>
<IfModule !mod_php5_filter.c>
<IfModule !mod_php5_hooks.c>
<IfModule mod_actions.c>
<IfModule mod_alias.c>
<IfModule mod_mime.c>
<IfModule mod_fcgid.c>
  # Path to php.ini – defaults to /etc/phpX/cgi
  DefaultInitEnv PHPRC=/etc/php5/cgi

  # Number of PHP childs that will be launched. Leave undefined to let PHP decide.
  #DefaultInitEnv PHP_FCGI_CHILDREN 3

  # Maximum requests before a process is stopped and a new one is launched
  #DefaultInitEnv PHP_FCGI_MAX_REQUESTS 5000

  # Define a new handler "php-fcgi" for ".php" files, plus the action that must follow
  AddHandler php-fcgi .php
  Action php-fcgi /fcgi-bin/php-fcgi-wrapper

  # Define the MIME-Type for ".php" files
  AddType application/x-httpd-php .php

  # Define alias "/fcgi-bin/". The action above is using this value, which means that
  # you could run another "php5-cgi" command by just changing this alias
  Alias /fcgi-bin/ /var/www/fcgi-bin.d/php5-default/

  # Turn on the fcgid-script handler for all files within the alias "/fcgi-bin/"
  <Location /fcgi-bin/>
    SetHandler fcgid-script
    Options +ExecCGI
  </Location>
</IfModule>
</IfModule>
</IfModule>
</IfModule>
</IfModule>
</IfModule>
</IfModule>
</IfModule>
</IfModule>
```

5. Next, create the directory which is chosen by the alias, and put in a symlink to the php5-cgi binary

```
# mkdir /var/www/fcgi-bin.d/php5-default
# ln -s /usr/bin/php5-cgi /var/www/fcgi-bin.d/php5-default/php-fcgi-wrapper
```

6. Finally, restart Apache

```
# /etc/init.d/apache2 restart
```

Additional configuration

It is also possible to run different versions of PHP, even with different users.

1. Enable `mod_suexec`

```
# a2enmod suexec
```
2. Create a new user

```
# adduser <username>
```
3. For every instance of PHP, create a new subdirectory in `/var/www/fcgi-bin.d/`

```
# mkdir /var/www/fcgi-bin.d/php<version>-<username>/
```
4. Instead of creating a symlink, this time you need to add a new file "php-fcgi-wrapper" inside this directory

```
#!/bin/sh
# Wrapper for PHP-fcgi
# This wrapper can be used to define settings before launching the PHP-fcgi binary.

# Define the path to php.ini. This defaults to /etc/phpX/cgi.
#export PHPRC=/var/www/fcgi-bin.d/php5-web01/phprc
#export PHPRC=/etc/php5/cgi

# Define the number of PHP childs that will be launched. Leave undefined to let PHP decide.
#export PHP_FCGI_CHILDREN=3

# Maximum requests before a process is stopped and a new one is launched
#export PHP_FCGI_MAX_REQUESTS=5000

# Launch the PHP CGI binary
# This can be any other version of PHP which is compiled with FCGI support.
exec /usr/bin/php5-cgi
```
5. Make this script executable and change the user appropriately

```
# chmod a+x /var/www/fcgi-bin.d/php<version>-<username>/php-fcgi-wrapper
# chown <user>:<group> /var/www/fcgi-bin.d/php<version>-<username>/php-fcgi-wrapper
```
6. Modify the Apache configuration, e.g. in `/etc/apache2/sites-enabled/web01`

```
<VirtualHost *:80>
    ServerName web01.myserver.com
    DocumentRoot /var/www/websites/web01/
    SuexecUserGroup <user> <group>
    Action php-fcgi /fcgi-bin/php-fcgi-wrapper
    Alias /fcgi-bin/ /var/www/fcgi-bin.d/php<version>-<username>/
</VirtualHost>
```

Notice: It is important that user and group are owned by the same user and group as defined for the wrapper script above.
7. Restart Apache

```
# /etc/init.d/apache2 restart
```

Appendix

Further reading

[1] Website of mod_fcgid: <http://fastcgi.coremail.cn/>

FAQ

How fast is it?

Compared to mod_php5 the speed seems to be almost equal, probably just a little bit slower.

Can I turn off the PHP parser for a specific directory?

Of course! You can simply disable the parser for directories. This is very useful for example to use WebDAV for editing PHP files. Using mod_php5 these files would be sent as plain HTML output:

```
<Location /dav/>  
    RemoveHandler .php  
</Location>
```

ChangeLog

Changes in version 1.0

- Initial version

About the author

Michael Stucki is the leader of the TYPO3 core team. He lives close to Basel / Switzerland. When he is not developing, he likes doing sports and travel around. Michael is working for snowflake productions, where this document was initially written as a help guide for their customers.

This document is published under the Open Content License available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3 - a GNU/GPL CMS/Framework available from www.typo3.com