

What are PHP Handlers and Why Do They Matter?

Apache does not natively support PHP scripts without a special module. The module that tells Apache how to handle PHP scripts is referred to as a PHP handler. Without a properly configured module, Apache will just send you the PHP file as a download since it doesn't know what else to do.

- [What handlers do cPanel and Plesk support with Apache?](#)
- [In broad terms, how does each PHP handler work and what are the pros and cons of working with each of them?](#)
 - [DSO/Apache Module](#)
 - [CGI](#)
 - [FCGI/FastCGI](#)
 - [PHP-FPM](#)
 - [suPHP](#)
 - [LSAPI](#)
- [Which PHP handler should I use?](#)
- [How do I change the PHP handler that Apache uses?](#)

What handlers do cPanel and Plesk support with Apache?

cPanel	Plesk
DSO/mod_php	Apache Module/mod_php
CGI	CGI
FCGI/FastCGI	FCGI/FastCGI
PHP-FPM	PHP-FPM
suPHP	-- (CGI w/ suExec enabled)
LSAPI	-- (unless LiteSpeed Web Server is installed)

In broad terms, how does each PHP handler work and what are the pros and cons of working with each of them?

Each module provides a different method of parsing PHP.

DSO/Apache Module

This is also referred to as mod_php. This module allows Apache itself to directly parse and display PHP files. PHP scripts parsed by mod_php run as the same user that Apache itself does (rather than the user account that hosts the PHP files).

Pros

- One of the fastest handlers available.
- Works with mod_ruid2 or mpm_itk modules.

Cons

- Only works with a single version of PHP on cPanel servers (you'll need to use other handlers for other versions of PHP if you offer them).
- Scripts run as the Apache user rather than the owner of the domain or subdomain. For example, on a cPanel server, if the script creates a file or directory, that file will be owned by the user "nobody" which can cause problems when the account owner goes to backup or remove them.

CGI

Stands for Common Gateway Interface. Using this handler, the system will run PHP scripts as the user that owns the domain or subdomain.

Pros

- Scripts run as the domain or subdomain user, not as the Apache user.

Cons

- One of the slowest handlers.
- Doesn't work well with PHP opcode caching.
- Cannot put PHP configuration changes in an .htaccess file.

FCGI/FastCGI

FastCGI is a variation of the CGI protocol that provides a number of benefits over the older CGI handler. Using this module, the system will run PHP scripts as the user that owns the domain or subdomain. There are some differences between `mod_fastcgi` and `mod_fcgid`, but none that are relevant to the scope of this article.

Pros

- Scripts run as the domain or subdomain user, not as the Apache user.
- Very fast handler.
- Works with PHP opcode caching.

Cons

- This handler uses more memory than most of the others.
- Cannot put PHP configuration changes in an `.htaccess` file.

PHP-FPM

FPM stands for FastCGI Process Manager. It is an improved way of implementing FastCGI processing of PHP. Using this handler, the system will run PHP scripts as the user that owns the domain or subdomain. Each FPM pool can have independent settings.

Pros

- Scripts run as the domain or subdomain user, not as the Apache user.
- One of the fastest PHP handlers.
- Works with PHP opcode caching.
- Allows for some additional level of flexibility per pool.

Cons

- This handler can use more memory than any other handler listed here, but that depends on the number of sites using PHP-FPM and the configuration of the FPM pool.
- Can be somewhat more complicated to manage.
- Cannot put PHP configuration changes in an `.htaccess` file and some directives can only be changed on a global level.

suPHP

This handler was specifically designed to serve PHP scripts as the owner of the domain or subdomain that is executing the PHP script. On cPanel servers, it is also configured to disallow execution of files with unsafe permissions. cPanel their copy of suPHP with the latest security fixes.

Pros

- Scripts run as the domain or subdomain user, not as the Apache user.
- cPanel configures suPHP so that it blocks accessing or executing any files or directories with permissions higher than 755 for security.

Cons

- Slowest PHP handler in most cases.
- PHP Opcode caching has no performance improvement and only wastes memory.
- Cannot put PHP configuration changes in an `.htaccess` file.

LSAPI

This handler implements the LiteSpeed Web Server (LSWS) SAPI. This handler requires CloudLinux or LSWS for the maximum benefits. Using this handler, the system will run PHP scripts as the user that owns the domain or subdomain.

Pros

- Designed to perform as well or better than PHP-FPM under certain circumstance.
- Less memory use than most other handlers.
- Scripts run as the domain or subdomain user, not as the Apache user.
- No special configuration required.
- Can read PHP values out of a `.htaccess` file.

Cons

- You don't get full benefits without purchasing a third-party commercial product.
- Not compatible with `mod_ruid2` or `mpm_itk` (but it shouldn't need them).

Which PHP handler should I use?

What you should use depends on your server's resources and the types of PHP sites you need to serve. On cPanel, we recommend defaulting to suPHP, LSAPI or FCGI. On Plesk we recommend FastCGI. Once you understand how PHP-FPM works, you may want to try enabling it for at least some sites and see if it helps at all.

How do I change the PHP handler that Apache uses?

Do not change the PHP handler you are using without fully understanding what the handler will do. If you switch from a handler that does run scripts as the user to one that doesn't, or vice versa, you may need to adjust the permissions of some files or directories so that Apache can access things properly.

On cPanel, you can adjust the PHP handler in WHM by clicking on **MultiPHP Manager** and then select the **PHP Handler** tab and adjust the handler as needed.

The settings in this area of cPanel only apply if you haven't configured a site to use PHP-FPM on the main **MultiPHP Manager** screen.

On Plesk, navigate to **Service Plans > Example plan > PHP Settings > run PHP as** and select the handler you want to use.

If you have service with HostDime and you are unsure what to do about your PHP handlers, please [contact us](#) for assistance before trying to make any changes.