

Performance

Performance parameters

This page combines many tools and tips that might help you improve your shop's performance server-wise – not sales-wise, although a server that performs well is able to serve more customers, and therefore make more sales.

Smarty

Smarty is the name of the template language used by PrestaShop's themes. You can learn more about it at <http://www.smarty.net/>.

SMARTY

Template compilation Never recompile template files
 Recompile templates if the files have been updated
 Force compilation

Cache YES NO

Multi-front optimizations YES NO

Caching type File System
 MySQL

Clear cache Never clear cache files
 Clear cache everytime something has been modified

Save

There are several options:

- **Template cache.** For better front-end performance, PrestaShop caches your HTML pages.
 - **Never recompile template files.** The normal behavior: HTML pages are compiled and cached, then displayed as-is, even if the theme has changed in the meantime.
 - **Recompile templates if the files have been updated.** PrestaShop is able to see when a theme file has changed.
 - **Force compilation.** Only enable this if you are making edits theme, and need to see your changes every time you reload your page.
- **Cache.** This option makes it possible for you to disable all file caches, and not just the one pertaining to template files. Only disable this if you are debugging a theme or a module for PrestaShop. Otherwise, you should leave it enabled. The "Clear cache" button at the top of the page makes it possible to delete the cache in a single click instead of having to go delete files on your FTP server.
- **Multi-front optimizations.** When you have several servers, this option helps you with their cache synchronization.
- **Caching type.** By default, Smarty uses a file-based caching mechanism. You can choose to have it use MySQL as the storage resource for Smarty's output caching.
- **Clear cache.** Depending on how often your store changes, you might want to either never clear the cache files, or clear them every time your store is modified (either products or design).

Debug mode

When in debug mode, you can choose to lessen the impact of certain features on PrestaShop, in order to best pinpoint where an error comes from:

DEBUG MODE

Disable non PrestaShop modules YES NO

Disable all overrides YES NO

Debug mode YES NO

 Save

- **Disable non PrestaShop modules.** PrestaShop's own module are heavily tested and should present no issue whatsoever. If you enable this setting, you will be able to tell if the problem comes from PrestaShop's own code (core or module), or from a third-party module.
- **Disable all overrides.** Many features in PrestaShop can be overridden. If you enable this settings, all overriding code will be disabled, and you will be able to tell if the problem comes from PrestaShop's own code, or from a third-party override.
- **Debug mode.** By enabling this option, technical error messages will be visible. It is useful when you are being helped by someone who does not have access to your shop but who needs to know what is going on technically.

Optional features

Some PrestaShop features can be disabled if you do not use them, as they can slow down your shop.

If your catalog currently has products which make use of these features, then you will not be able to disable them. You will have to delete some data before you can turn them off.

You can disable the following features:

OPTIONAL FEATURES

 Some features can be disabled in order to improve performance.

Combinations YES NO
You cannot set this parameter to No when combinations are already used by some of your products

Features YES NO

Customer Groups YES NO

 Save

- **Combinations.** Product combinations enable you to have a whole product line out of a single product: varying size, colors, capacity, etc.
- **Features.** Product features enable you to indicate the product's specific information: weight, material, country of origin, etc.
- **Customer groups.** Customer groups enable you to group customers in order to give them certain privileges and restrictions: discount, module restrictions, etc.

Combine, Compress and Cache (CCC)

CCC is a set of tools aimed at minimizing server load and theme loading time.

It does what it says: it combines textual files of the same type into one bigger file, which makes for fewer files to download; it then compresses the file using the common Zip algorithm, which makes for faster downloads; finally, it caches the compressed file, so that the server does not have to do this process every time a page is loaded, which relieves the burden on the server's processor.

CCC (COMBINE, COMPRESS AND CACHE)

CCC allows you to reduce the loading time of your page. With these settings you will gain performance without even touching the code of your theme. Make sure, however, that your theme is compatible with PrestaShop 1.4+. Otherwise, CCC will cause problems.

Smart cache for CSS YES NO

Smart cache for JavaScript YES NO

Apache optimization YES NO

Save

- **Smart cache for CSS.** CSS files are text-based, and can be safely combined and compressed.
- **Smart cache for JavaScript.** JavaScript files are also text-based, but their combination can sometimes prove problematic. Make sure to test everything before leaving that setting enabled.
- **Apache optimization.** This setting will change your web server's configuration file in order to make it more efficient for CCC.

Media servers

This section enables you to redirect part of your traffic (image and video files, for instance) to other servers under your control, through other domains or sub-domains – most often, the files are hosted on a CDN (Content Delivery Network). By default, PrestaShop supports one media server.

MEDIA SERVERS (USE ONLY WITH CCC)

You must enter another domain, or subdomain, in order to use cookieless static content.

Media server #1

Save

Putting your store's domain name in this field is not the proper way to get fantastic performance. That being said, it is easy to set up a media server, and the benefits are real and almost immediate. Here is how:

1. Open an account at a new host, preferably one who is a specialist of distributed content. The most popular are Akamai (<http://www.akamai.com/>), Amazon (with its AWS services, among which is CloudFront: <http://aws.amazon.com/>) or CloudFlare (<http://www.cloudflare.com/>). You should also ask your own host, maybe it has a CDN service you can subscribe to.
2. Copy your media files to that host's server. This means that the CDN server must contain an exact copy of the following folders from your store's main server: `/img`, `/themes` and `/modules`.
Reminder: you must make it so that these folders are always synchronized: even if you add new products or change your theme, the CDN server must contain the latest version of all these files.
3. Once the CDN server is in place, add the web address (as given by your CDN host) in the field, "Media server #1". If that host allows for more web addresses, add them.

In case you would rather that your files are still visually downloaded from your domain name rather than from an unknown domain name, follow this process:

1. Create a subdomain for your store's domain name, for instance <http://cdn1.example.com> (the way to do that depend on your host, ask him about it).
2. Put a `.htaccess` file at the root of the subdomain. That file should contain a single line:

```
Redirect Permanent / http://cdn-adress.com
```

Replace the `http://cdn-adress.com` with the one from your CDN server. This way, you are creating an automatic redirection from your subdomain to your CDN server.

3. Once the subdomain is in place, add it in the first field, "Media server #1". If that host allows for more web addresses, create as many subdomains to your store's main domain name.

Even if you do not have a CDN server, you can use the media server feature to have the customer's browser download more files at a time, thus making the whole page loading process faster:

1. Configure your web server to have virtual subdomains, such as images1.example.com, images2.example.com, and images3.example.com, pointing to PrestaShop's main folder.
2. In your back office's Performance page, set each "Media server" field to these virtual subdomains.

Once this is in place, PrestaShop will load your images from either of these subdomains. In effect, images will come from the same folder (the main one), but the browser will open several more connection threads to your web server than it would by default, thus making the whole page load faster.

Caching

Your server's cache stores static versions of your dynamic web page, in order to serve these to your customers and thus reduce server load and compiling time.

Most of the time, you should first check with your web host about this setting, as it requires special settings on the server.

This section enables you to choose to enable caching, and then choose the caching method:



The screenshot shows a web interface for configuring caching. At the top left, there is a tab labeled 'CACHING'. Below it, there is a section titled 'Use cache' with two buttons: 'YES' (disabled) and 'NO' (active). In the bottom right corner, there is a 'Save' button with a floppy disk icon.

- **Memcached.** A distributed caching system. Very effective, above all with multiple servers, but you need to make sure that your servers/hosts support it – most probably, if your PHP configuration features the Memcached PECL extension, you are good to go (you can download it here: <http://pecl.php.net/package/memcache>). You can add Memcached servers by clicking on the "Add server" link.
- **APC.** Alternative PHP Cache is free, open and robust, but only works with one server – which is the usual case when you start your online business. Again, check for the availability of the APC PECL extension on your server: <http://pecl.php.net/package/APC>.
- **Xcache.** Xcache is a new cache system, which is specific to the Lighttpd server – hence, it will not work with the popular Apache server. Read more about it at <http://xcache.lighttpd.net/>.