

USER MANUAL

Frequently asked questions

(CMS Prestashop)

07.11.2018

Author: [zapalm](mailto:zapalm@prestashop.modulez.ru), prestashop.modulez.ru

Terms and conditions of use: [CC BY-NC-ND 3.0](http://creativecommons.org/licenses/by-nc-nd/3.0/)

Contents

About FAQ.....	1
Abbreviations.....	1
Information about the FAQ progress.....	2
Where else to look for answers about Prestashop?.....	2
Questions / Answers.....	2
Qn: [zapalm] In a module's manual is written «See PDF-attach for (some files)». I don't see these files. Where are these files?.....	2
Qn: [PS1.5-1.6] Why «The version of your module is not compliant with your PrestaShop version» message is appeared?.....	2
Qn: [PS1.2-1.6] How to enable displaying error messages or debugging in Prestashop?.....	3
Qn: [PS1.5-1.6] How to enable a debug mode for a programmer when a shop on the production?.....	3
Qn: [PS1.2-1.6] How to disable Smarty caching in Prestashop?.....	4
Qn: [PS1.5] How to clear Prestashop classes cache?.....	5
Qn: [PS1.4-1.5] How to fix problem like «The method X in the class X is already overridden» when installing a module?.....	5

About FAQ

Abbreviations

Qn – question.

Ar – answer.

PS – Prestashop.

[] – search tag.

Information about the FAQ progress

The FAQ is first started to walk, but still stumbles and falls ...

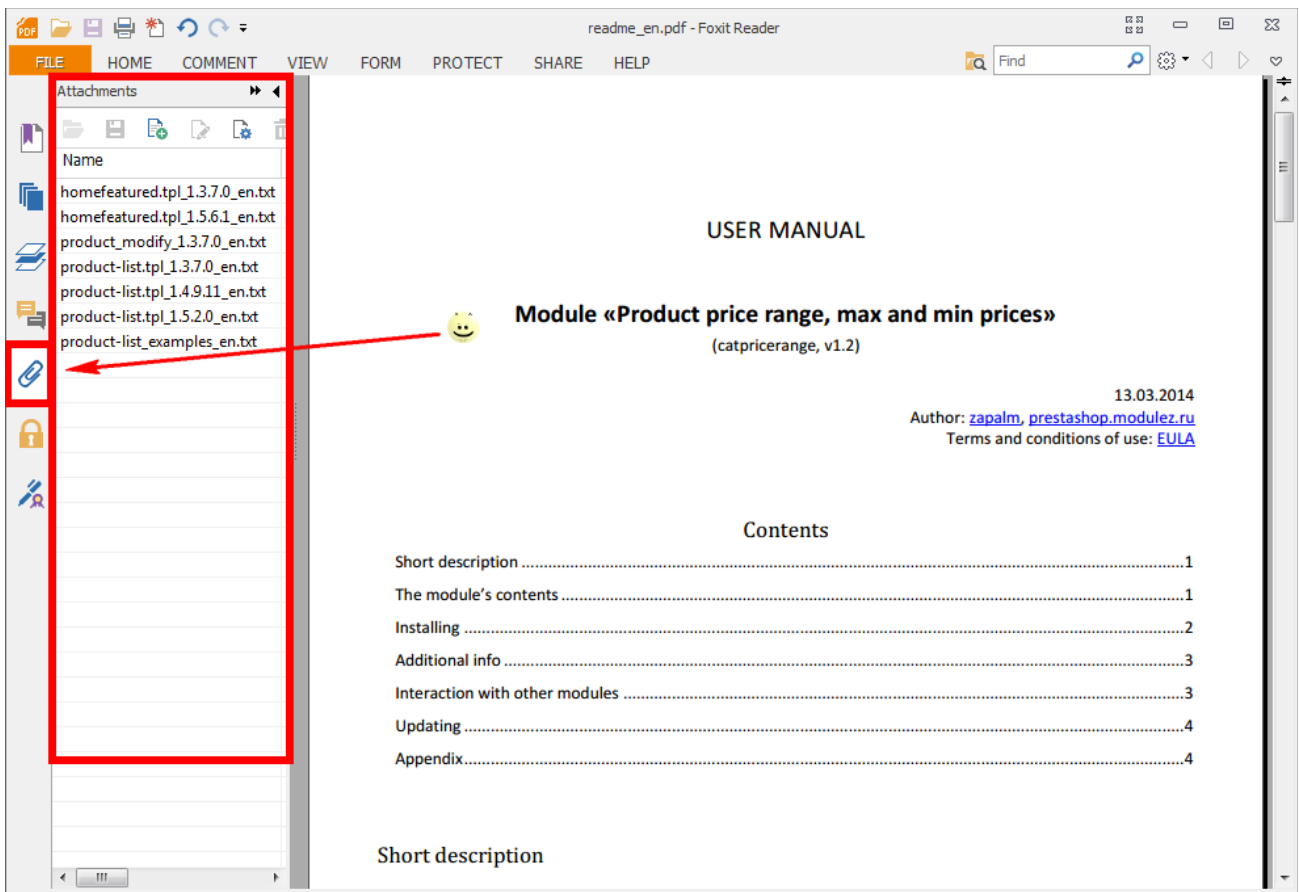
Where else to look for answers about Prestashop?

1 <http://stackoverflow.com/questions/tagged/prestashop>

Questions / Answers

Qn: [zapalm] In a module's manual is written «See PDF-attach for (some files)». I don't see these files. Where are these files?

Ar: See screenshot for details (PDF-file is opened in the [Foxit Reader](#)). You may use another modern PDF-reader like [Adobe Reader](#) or others you like.



Qn: [PS1.5-1.7] Why «The version of your module is not compliant with your PrestaShop version» message is appeared?

Ar: Comment by // the line of code like this:

```
// $this->ps_versions_compliancy = array('min' => '1.3.0.0', 'max' => '1.6.0.0');
```

You can find the line of this code in the main file of the module, for example: /modules/mymodule/mymodule.php

Qn: [PS1.2-1.6] How to enable displaying error messages or debugging in Prestashop?

Ar: PrestaShop's default settings prevent the customer to see any server error message or any debugging code. You, on the other hand, need this information in order to correct any potential mistake in your code.

For PS1.5-1.6: Open the «/config/defines.inc.php» file, and edit it to set «_PS_MODE_DEV_» to «true»:

```
define('_PS_MODE_DEV_', true);
```

For PS1.2-1.4: Open the «/config/config.inc.php» file, and edit it to set «display_errors» to «on»:

```
@ini_set('display_errors', 'on');
```

For PS1.4 (additional): you may to enable debug functions: *p()* and *d()* by setting «_PS_MODE_DEV_» to «true» in the «/config/defines.inc.php» file.

See also «How to enable a debug mode for a programmer when a shop on the production?».

Qn: [PS1.5-1.6] How to enable a debug mode for a programmer when a shop on the production?

Ar: It is can do by this modification of the «/config/defines.inc.php» file.

1 open in a text editor '/config/defines.inc.php' file and before this code:

```
/* Debug only */
if (!defined('_PS_MODE_DEV_'))
define('_PS_MODE_DEV_', false);
```

2 place the code:

```
// may be helpful when the shop on the production but needed debug for a programmer
// enabling debug mode by get-param when _PS_MODE_DEV_ is false
// comment next 3 lines when the debugging has done
if (!defined('_PS_MODE_DEV_') && isset($_GET['debug_me']))
    define('_PS_MODE_DEV_', true);
else
```

The result of the modification should look like this:

```

// may be helpful when the shop on the production but need
// enabling debug mode by get-param when _PS_MODE_DEV_ is
// comment next 3 lines when the debugging has done
if (!defined('_PS_MODE_DEV_') && isset($_GET['debug_me']))
    define('_PS_MODE_DEV_', true);
else
    /* Debug only */
    if (!defined('_PS_MODE_DEV_'))
        define('_PS_MODE_DEV_', false);
    /* Compatibility warning */
    define('_PS_DISPLAY_COMPATIBILITY_WARNING_', false);
    if (_PS_MODE_DEV_)
    {
        @ini_set('display_errors', 'on');
        @error_reporting(E_ALL | E_STRICT);
        define('_PS_DEBUG_SQL_', true);
    }
else
    {

```

To use it a programmer should append to URL an additional param 'debug_me', for example:

http://localhost/en/tshirts/1-faded-short-sleeve-tshirts.html?debug_me

Or like this if the 'Friendly URL' option is disabled:

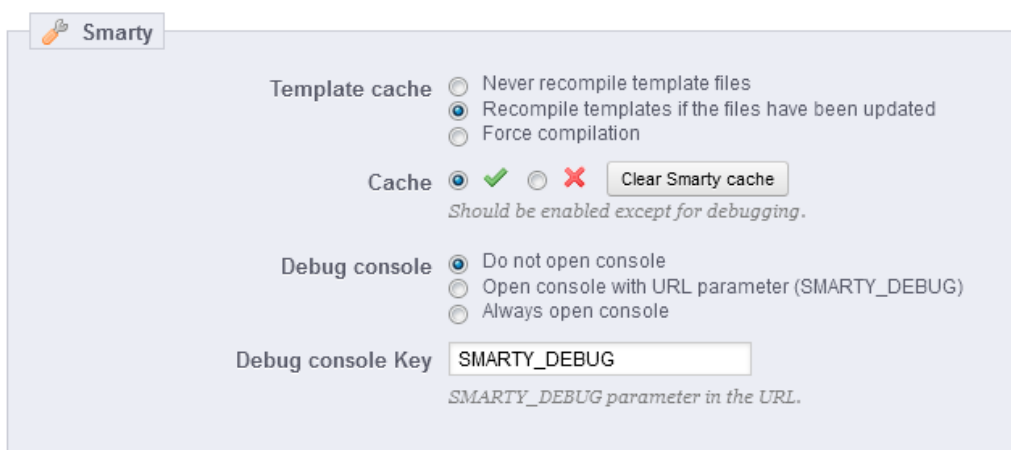
http://localhost/index.php?id_product=1&controller=product&id_lang=1&debug_me

When the debugging has done just comment by pair '//' those 3 lines of the new code to disable the feature but for further use.

Qn: [PS1.2-1.6] How to disable Smarty caching in Prestashop?

Ar: It's necessary for debug purpose. Also you need to disable any caching to improve debugging. After debugging you should revert config changes.

For PS 1.4-1.6: Go to the «Advanced Parameters / Performance» tab to disable Smarty caching.



Set «Force compilation» of «Template cache» group option and disable «Cache» option.

For PS1.2-1.3: Open the «/config/smarty.config.inc.php» file, and edit it to set properties as is shown:

```

$smarty->caching          = false;
$smarty->force_compile    = true;
$smarty->compile_check    = false;

```

Qn: [PS1.5] How to clear Prestashop classes cache?

Ar: Modules may add an override to a default behavior, and PrestaShop takes care of resetting the «/cache/class_index.php» file. But sometimes you need to add that overriding code yourself, manually uploading the file to your server. In that case, you need to trigger the regeneration of the «/cache/class_index.php» file yourself. This is done simply by deleting the file: if PrestaShop cannot find the file, it will regenerate it, taking all the overrides into account. It is the same when manually removing an override: in order to reinstate the default behavior, you must delete the «/cache/class_index.php» file.

Qn: [PS1.4-1.7] How to fix problem like «The method X in the class X is already overridden» when installing a module?

Ar: This is not a bug or an issue of some module. This is the limitation of Prestashop installing system and [overriding system](#).

When you install PrestaShop module/themes from different vendors, you may encounter errors like above that prevent you from installing the new module.

All classes and methods are only allow override one time. For example, if your TV has only one HDMI input port and the port has been used by other device (like DVD player), and now you purchased new blueray player also uses HDMI port too. You will find you are not able to connect the new blueray player to your TV because the only HDMI port has been used.

What is class/method override?

PrestaShop provide a flexible architecture for 3rd party vendors or store owners to extend the functionality and features by addons or modules or just simple modifications. The new addon features and functions will need to modify some behaviour of original PrestaShop class and method, This is implemented by class/method override. All override class/methods are installed under following folders:

YourStoreRoot/override/classes/

YourStoreRoot/override/controllers/

Why override conflicts?

In the whole PrestaShop, you can have only one one override for each class or each method, and the override must be installed to the above folder. There are many theme vendors and modules vendors are developing themes and addons/modules. Some time they will override the same class and method. If that happens, the conflicts occurs. The first one install without a problem, but later one will not be able to installed because of the conflicts and PrestaShop installer will show the error message listed previously.

How to resolve the error

In order to resolve the conflicts, you will have to merge override classes/methods which are in conflicts from both module/theme, then you can continue to install the modules again.

Caution.

Code merge could be simple and could be very complicated, some time you have to read the code to understand the logic of code to see which piece of code needs to be tun first and which piece of code should be run after. If

you do not understand PHP code and PrestaShop, it is better do not do the merge by yourself. Otherwise you may break the code and make system not working properly.

Here is the basic steps to merge the files if you are able to do that by yourself.

Steps 1. Identify the file name and methods that are in conflicts.

The conflict could be common class or an page controller. You can tell from the class name part of the message.

For example, following message indicate the class : Category, the method: getRootCategory

The method getRootCategory in the class Category is already overridden.

All page controllers file name always ends with "Controller.php", something like xxxxxxxController.php, for example *CategoryController.php* for category page.

Step 2. Merge the code.

We are using the same Category.php as an example.

To be merged file: */modules/somemodule/override/classes/Category.php* - the module name will differ based on the module you are installing.

Destinationfile to merge code into: */override/classes/Category.php*

Step 3. Rename class file which has been merged in the new installing module.

After you have merged the code into the installation destination file, you will need to rename the file that has been merged so that it will not be installed again.

You need to change the file name to anything else that does not end with .php,for example you can change to *Category.php-org*

Step 4 Repeat above Steps 1-3 for each conflicts class/method to reolveve them all.

Step 5. Try install again.

After you have finish merging all conflicts code, you can try again the installation, it should be able to install the module now.

Note: The right to override class/method are equal to all vendors (theme/modules vendors). all vendors should be treated fairly.