

One of the purposes of [Wiki Groups](#) is to allow a [Wiki Administrator](#) to customize the features of PmWiki on a per-group basis. Here is where *per group customizations* come into play.

- The *local/* subdirectory (typically in [\\$FarmD](#)) is used to hold local configuration files.
- The *css/* subdirectory (typically in [\\$FarmD](#)) is used to hold local css files.

To perform [local customizations](#) for a particular WikiGroup,

- place the customizations in a file called "*<groupname>.php*" (where *<groupname>* is the actual name of the page group in question) in the *local/* subdirectory
- place the css customizations in a file called "*<groupname>.css*" (where *<groupname>* is the actual name of the page group in question) in the *css/* subdirectory.

These files will be automatically processed after processing any local customizations in the *config.php* and *local.css* files.

For example, to change the image displayed in the upper-left corner of pages in the "GroupName" WikiGroup, one could create *local/GroupName.php* containing

```
<?php
$PageLogoUrl = "/myimages/chess.gif";
```

The example's effect would cause all pages in the GroupName [Wiki Group](#) to use "/myimages/chess.gif" as the logo image instead of the default.

To add markup to the beginning or end of each page in a wiki group, see [Group headers](#).

Per-page customizations

PmWiki also allows per-page customizations, simply use the full name of the page to be customized instead of the group. For example, one can use the file *local/Chess.HomePage.php* to set local customizations for Chess.HomePage.

Almost any customization that would be placed in *config.php* can be used as a per-group or per-page customization.

An important exception to this is setting **per-group or per-page customizations for recipe scripts** included in *config.php*. Most recipe scripts would need any customization variables defined before the script is included. Instead of using a per-group or per-page customization php file, use a per-group or per-page conditional statement in *config.php*, before including the recipe script. Example:

```
$page = PageVar($pagename, '$FullName');
$group = PageVar($pagename, '$Group');
//per-group customizations:
if($group=='GroupName') {
    $RecipeVariable = 'valueA';
    etc. ...
}
//per-page customizations:
if($page=='GroupName.PageName') {
    $RecipeVariable = 'valueB';
    etc. ...
}
//include recipe after variables are set:
include_once('cookbook/recipescript.php');
```

Processing order

For all local customizations, PmWiki first processes the *local/config.php* file, and then looks for a per-page customization file in the *local/* subdirectory to process, followed by any per-group customization file. If no per-page or per-group customizations are loaded, then PmWiki loads *local/default.php*. If a per-page customization wants to have the per-group customizations loaded first, it can do so directly by using PHP's `include_once()` function. For more information see [wiki cascades](#).

Custom CSS styles per group or per-page

To apply CSS styles to pages of a specific group named [Group Name](#), create a file named *GroupName.css* in the *pub/css/* directory and add the CSS style rules there. To apply styles to a specific page, create a file *GroupName.PageName.css* in this directory with your style rules. Any CSS rules to be applied for all wiki pages can be put into *pub/css/local.css*.

```
/pub/css/GroupName.css:
```

```
body { background: #F4C4B4; }
```

Preventing group-Level configurations

Any customization file can set [\\$EnablePGCust](#)=0; to prevent later page/group/default customizations from being automatically loaded. If a per-page customization needs to have the per-group customizations loaded first, it can do so directly by using PHP's `include_once()` function.

Authentication

Any passwords required for a group should be set in the group's [Group Attributes](#) page (see [Passwords Administration](#)) and not in a group customization file.

Consider Wiki Farms

[Wiki Groups](#) are an easy way to host multiple sites in a single PmWiki installation by giving each site its own group. Another approach is to use [Wiki Farms](#), which allows each site to have its own set of [Wiki Group](#) and local customization files. Read about

If you are looking for nested group levels, you may want to consider [Pm's design considerations on hierarchical groups](#).

How can I apply CSS styles to a particular group or page?

Simply create a *pub/css/Group.css* or *pub/css/Group.Page.css* file containing the custom CSS styles for that group or page.

Why shouldn't passwords be set in group (or page) customization files?

The reason for this advice is that per-group customization files are only loaded for the current page. So, if `$DefaultPasswords['read']` is set in *local/GroupA.php*, then someone could use a page in another group to view the contents of pages in GroupA. For example, Main.WikiSandbox could contain:

```
(:include GroupA.SomePage:)
```

and because the *GroupA.php* file wasn't loaded (we're looking at Main.WikiSandbox --> *local/Main.php*), there's no read password set.

The same is true for page customization files.

Isn't that processing order strange? Why not load per page configuration last (that is after global configuration an per group configuration)?

Many times what we want to do is to enable a certain capability for a group of pages, but disable it on a specific page, as if it was never enabled. If the per-group config file is processed first, then it becomes very difficult/tedious for the per-page one to "undo" the effects of the per-group page. So, we load the per-page file before the per-group.

If a per-page customization wants the per-group customizations to be performed first, it can use the techniques given in [PmWiki.GroupCustomizations](#) (using *include_once()* or setting `$EnablePGCust = 0`).

Copyright: Simon - 27/08/2009