

Variable substitutions in the skin template are all managed by the `FmtPageName()` function from `pmwiki.php`. Pmwiki variable substitutions available on pages are managed by the substitutions from `stdmarkup.php` or superseded in `local/config` files.

[\\$ActionSkin](#)

This array is used to override the current skin when performing a given action. The most common use is to set [\\$ActionSkin](#) []

[\\$WikiTitle](#)

A variable which contains the Wiki title as displayed by the browser

[\\$EnablePageTitlePriority](#)

A variable defining how to treat multiple (`:title ...:`) [page directives](#) (added in PmWiki 2.2.9).

[\\$EnablePageTitlePriority](#) = 0 ; # PmWiki default, last encountered title wins (the title may be changed from included pages)

[\\$EnablePageTitlePriority](#) = 1 ; # First title wins; if a title is defined in the page, directives from included pages cannot change

[\\$EnableDiffInline](#)

If set to 0, this variable switches off the word-level highlighting on the markup in the [page history](#).

[\\$EnableDiffInline](#) = 0 ; # Disable colors, show plain text differences

[\\$HTMLStylesFmt](#)

An array of CSS statements to be included in the page's output along with other HTML headers. This array provides an easy place

[\\$HTMLHeaderFmt](#)

An array of HTML text to be included in the page's `<head>` section, at the point where the [skin template](#) specifies a `<!--HTMLHeader-->`

For example, if you want to specify a logo for all the pages of your wiki (a png image for Firefox (and others...), an ico for Internet

Another example, if you want to get the rss notification on some browsers (the rss icon in firefox for instance):

```
$HTMLHeaderFmt
'<link rel="
href="$Sc
```

[\\$HTMLFooterFmt](#)

Like [\\$HTMLHeaderFmt](#) above, this contains an array of HTML text to be included near the end of an HTML document, at the point just before a closing </body> tag). Primarily used by scripts to add custom HTML output after the body of the page output.

[\\$MetaRobots](#)

Sets the value of the <meta name='robots' ... /> tag generated by PmWiki to control search engine robots accessing the page view, and to not index pages in the PmWiki [wiki group](#). Explicitly setting [\\$MetaRobots](#) overrides this default.

```
# never index this site
$MetaRobots = 'noindex,nofollow'; # disable the robots tag entirely $MetaRobots =
'';
```

[\\$MessagesFmt](#)

An array of HTML text to be displayed at the point of any (:messages:) markup. Commonly used for displaying messages with

[\\$RecentChangesFmt](#)

An array specifying the format of the RecentChanges listing.

The key of the array specifies the page where changes will be logged, as in

```
$RecentChangesFmt['$SiteGroup.AllRecent
```

The value of the array specifies the format in which the changes will be logged, as in

```
'* [[{$Group}.$Name]] . . . $
```

Note the two consecutive spaces before the three dots (. . .). The two spaces separate two parts of the format: the first part does change (e.g. a link to the changed page) and the second part does not change (e.g. the date and author of the change). Upon saving a page, PmWiki removes a line that matches the first part and adds a line with the current format before the first line with 2 spaces. In this way, any line without two consecutive spaces stays at the top of the recent changes page.

You can use and adapt the following to change the format (put it in config.php):

```
$RecentChangesFmt['$SiteGroup.AllRecent
'* [[{$Group}.$Name]] . . . $Cu
$RecentChangesFmt['$Group.RecentChan
'* [[{$Group}/{$Name}]] . . . $Cu
```

Note that changes made to the format will only affect new edits. In other words, you will need to edit a page for your new format to be visible. Note also that you need to have two spaces between the page name and the other information about the edit.

Also note that this variable has other uses, such as not reporting at all to RecentChanges and AllRecentChanges as found here [PmWiki Questions](#).

[\\$RecentUploadsFmt](#)

An array specifying the format for uploaded files at the RecentChanges listing. It is similar to [\\$RecentChangesFmt](#). If enabled, RecentChanges pages. Default is disabled. See [Cookbook:RecentUploadsLog](#) for more information.

[\\$DraftRecentChangesFmt](#)

An array specifying the format of the RecentChanges listing when saving Draft pages.

[\\$RecentChangesFmt](#) is set to [\\$DraftRecentChangesFmt](#) when a Draft page is saved. For example, you could save drafts in

[\\$RCLinesMax](#)

The maximum number of lines to be stored in RecentChanges pages. The default is zero, meaning "no limit".

```
\$RCLinesMax = 1000; # maintain at most 1000 recent changes
```

[\\$PageRedirectFmt](#)

The text to be used when a page is redirected via the (:redirect:) markup.

```
$PageRedirectFmt = '<p><i>redirected
$PageRedirectFmt = '';
```

For display options, see also the FAQ on [PageDirectives](#).

[\\$WikiStyle](#)

An array which contains the predefined WikiStyles which can be used on a textpage.

See: [PmWiki.CustomWikiStyles](#)

[\\$WikiStyleApply](#)

An array which defines the scope of wiki styling per HTML element. Default settings are:

```
'item' => 'li|dt',
'list' => 'ul|ol|dl',
'div' => 'div',
'pre' => 'pre',
'img' => 'img',
'block' => 'p(?:!\sclass=)|div|ul|ol|dl|li|dt|pre|h[1-6]',
'p' => 'p(?:!\sclass=)'
```

This defines that we can apply wiki styling on:

- LI elements using the *item* keyword
- UL, OL, DL elements using the *list* keyword
- etc.

An example of applying scope to an LI element is below. For more information refer to [WikiStyle scopes](#).

```
* %apply=item red%Here is a red styled
list item
* This item would not be styled.
```

- Here is a red styled list item
- This item would not be styled.

You can [add additional HTML elements to \\$WikiStyleApply](#) to apply wiki styles to other HTML elements. For example to allow styling on table rows, or anchor tags.

[\\$MaxIncludes](#)

Controls the number of times that pages can be included via the (:include:) and other directives, used to control recursion and otherwise pose a sanity check on page contents. [\\$MaxIncludes](#) defaults to 50, but can be set to any value by the wiki administrator.

```
\$MaxIncludes = 50; # default \$MaxIncludes = 1000; # allow lots of includes
\$MaxIncludes = 0; # turn off includes
```

\$Skin

Lists the name(s) of skins to load, unless overridden by [\\$ActionSkin](#). Normally \$Skin contains a single string which is a the name of the skin used.

[\\$SkinDirUrl](#)

Set by *scripts/skins.php* to be the base url of the current skin's directory (i.e., within a 'pub/skins/' directory). This variable is typically used to construct the URL of the skin's directory.

[\\$SkinLibDirs](#)

An array which, given the filesystem path (array key) to a skin (or a directory containing several skins), provides the corresponding URL of the skin's directory.

The array key is the directory containing the skin.tmpl and skin.php files, as seen by the PmWiki program. It does not have to be the same as the URL of the skin's directory.

The value is the URL (web address) of the directory containing the .css, .gif, and other files which appear in the HTML code sent to the browser.

By default [\\$SkinLibDirs](#) is set to:

```
$SkinLibDirs = array(
    "./pub/skins/\$Skin" => "$PubDirUrl/skins/\$Skin",
    "$FarmD/pub/skins/\$Skin" => "$FarmPubDirUrl/skins/\$Skin");
```

Extra details: When PMWiki is searching for a skin it looks for a directory named for the skin in the array index/keys, and if it finds it then it will use the files in that directory and also the files in the matching array value url. The two sides normally point to the same publicly accessible directory, but they do not have to.

[\\$PageLogoUrl](#)

is the url that refers to a logo image which most skins display somewhere in the page's header (top left usually).

[\\$EnablePathInfo](#)

Changes the handling of the page URL. When set to 1 page URL will be ...wiki.php/Main/Main, when set to 0 (default) it v

[\\$EnableFixedUrlRedirect](#)

When PmWiki is given a partial page name (e.g., just the name of a WikiGroup), it uses [\\$PagePathFmt](#) in order to make a comp with the correct full page name. Setting [\\$EnableFixedUrlRedirect](#)=0; blocks the redirect, so that PmWiki continues process

[\\$GroupHeaderFmt](#)

Defines the markup placed at the top of every page. Default value is:

```
$GroupHeaderFmt = '(:include $Group.GroupHeader:)(:nl:)' ;
```

[\\$GroupPrintHeaderFmt](#)

Defines the markup placed at the top of every page when action=print. Default value is:

```
SDV($GroupPrintHeaderFmt, '(:include $Group.GroupPrintHeader:)(:nl:)' ;
```

[\\$GroupFooterFmt](#)

Defines the markup placed at the bottom of every page. Default value is:

```
$GroupFooterFmt = '(:include $Group.GroupFooter:)(:nl:)' ;
```

[\\$GroupPrintFooterFmt](#)

Defines the markup placed at the bottom of every page when action=print. Default value is:

```
SDV($GroupPrintFooterFmt, '(:nl:)(:include $Group.GroupPrintFooter:)' ;
```

[\\$PageNotFoundHeaderFmt](#)

Specifies the HTTP header to send when attempting to browse a page that doesn't exist. Some webserver packages (notably Mi

```
# default
$PageNotFoundHeaderFmt = 'HTTP/1.1 404 Not Found';
# return all pages as found
$PageNotFoundHeaderFmt = 'HTTP/1.1 200 Ok' ;
```

Beware when expecting to return the content of a Group(header/footer) for an non existent page! By default PmWiki returns 404 (because the page does not exist), despite there is some content to show. Firefox shows the content, while Internet Explorer displays its default 404 page. [\\$PageNotFoundHeaderFmt](#) MUST be set to return 200 as described above in order to get the expected behaviour with all browsers.

[\\$HTMLVSpace](#)

Setting `$HTMLVSpace = ''` in a local customization file (e.g., `local/config.php`) prevents insertion of spacer paragraphs (place the `$HTMLVSpace = ''` statement in a `skin.php` file, preceded by the statement `global $HTMLVSpace`;

[\\$TableCellAttrFmt](#)

For [Tables](#), defines the HTML attributes given to each `<td>` or `<th>` cell in the output. Can contain references to `$TableCellCount`

[\\$TableRowAttrFmt](#)

For [Tables](#), defines the HTML attributes given to each `<tr>` element in the output. Can contain references to `$TableRowCount` and `$TableRowIndex` from 1 to [\\$TableRowIndexMax](#).

```
# Give each row a unique CSS class based on row number (tr1, tr2, tr3, ... )
\$TableRowAttrFmt = "class='tr\$TableRowCount'"; # Give each row alternating CSS
classes (ti1, ti2, ti1, ti2, ti1, ... ) \$TableRowIndexMax = 2; \$TableRowAttrFmt =
"class='ti\$TableRowIndex'";
```

[\\$TableRowIndexMax](#)

The maximum value for `$TableRowIndex` in [Tables](#).

```
# Set rows indexes as 1, 2, 3, 1, 2, 3, 1, 2, ...
\$TableRowIndexMax = 3;
```

[\\$FmtV\['\\$TableCellCount'\]](#)

PMWiki internal variable - Horizontal column number of the current cell. For use in [\\$TableCellAttrFmt](#) and [\\$TableRowAttrFmt](#). Administrators can use in [\\$TableCellAttrFmt](#) and/or [\\$TableRowAttrFmt](#).

```
Example: $TableCellAttrFmt = 'class=col\$TableCellCount';
```

[\\$FmtV\['\\$TableRowCount'\]](#)

PMWiki internal variable - Current row number. Administrators can use in [\\$TableCellAttrFmt](#) and/or [\\$TableRowAttrFmt](#).

```
Example: $TableRowAttrFmt = "class='row\$TableRowCount'";
```

[\\$FmtV\['\\$TableRowIndex'\]](#)

PMWiki internal variable - Row index number derived from [\\$TableRowIndexMax](#). (1,2,3,1,2,3,...). Administrators can use in [\\$TableCellAttrFmt](#) and/or [\\$TableRowAttrFmt](#).

```
Example: $TableRowAttrFmt = "class='ind\$TableRowIndex'";
```

See also: [Edit Variables](#)

Copyright: DaveG - 14/03/2010