

From EULER Project

PmWiki: Page Directives

PmWiki uses a number of directives to specify page titles, descriptions, page keywords, and control the display of various components. Keywords are not case sensitive.

`(:attachlist:)`

Shows a list of attachments of the current group or page, depending on whether attachments are organised per group or per page. The attachlist is displayed at the foot of the uploads page form.

The parameter to `(:attachlist:)` always resolves to a pagename. The directive then displays all of the attachments currently available for that page.

Options

`(:attachlist NAME:)` shows a list of attachments of the group or page NAME.

`(:attachlist ext=xxx:)` specifies an extension for filtering by type of file.

`(:attachlist *:)` shows the uploads directory and permits browsing of all uploaded files by directory (will not work if *\$EnableDirectDownload* is set to 0).

`(:description text:)`

Descriptive text associated with the page. (Generates a `<meta name='description' content='...' />` element in the page output.)

`(:keywords word1, word2, ...:)`

Identifies keywords associated with the page. These are not displayed anywhere, but are useful to help search engines locate the page. (Essentially, this generates a `<meta name='keywords' content='...' />` element in the output.)

`(:linebreaks:), (:nolinebreaks:)`

Honors any newlines in the markup; i.e., text entered on separate lines in the markup will appear as separate lines in the output. Use `(:nolinebreaks:)` to cause text lines to automatically join again.

`(:linkwikiwords:), (:nolinkwikiwords:)`

Enables/disables WikiWord links in text.

`(:markup:) ... (:markupend:)` or `(:markup:) [=...=]`

Can be used for markup examples, showing first the markup and then the result of the markup.

Options

`(:markup class=horiz:)` will show the markup side by side instead of one upon the other.

`(:markup caption='...':)` adds a caption to the markup example.

`(:markupend:)` is not required when using `(:markup:) [=...=]`.

Note that the placement of newlines is very important for this markup. If you are using the [= . . . =] option then the opening [= MUST occur on the same line as the (:markup:). If you are using the full (:markup:) ... (:markupend:) form then your markup code must appear AFTER a newline after the initial (:markup:).

`(:noaction:)`

Turns off the section of the skin marked by `<!--PageActionFmt-->` thru `<!--/PageActionFmt-->`. In the pmwiki skin, this turns off the display of the actions at the top-right of the page (other skins may locate the actions in other locations).

`(:nogroupheader:)`

`(:nogroupfooter:)`

Turns off any groupheader or groupfooter for the page. (See [GroupHeaders](#).)

`(:noheader:), (:nofooter:)`

`(:noleft:), (:noright:), (:notitle:)`

If supported by the [skin](#), each of these turns off the corresponding portion of the page.

`(:redirect PageName:)`

Redirects to another wiki page.

`(:redirect PageName#anchor:)`

Redirects to an [anchor](#) within a page

`(:redirect PageName status=301 from=name quiet=1:)`

Redirects the browser to another page, along with a redirect message. For security reasons this only redirects to other pages within the wiki and does not redirect to external urls. The `status=` option can be used to return a different HTTP status code as part of the redirect. The `from=` option limits redirects to occurring only on pages matching the wildcarded *name* (helpful when `(:redirect:)` is in [another page](#)). The `quiet=1` option allows the target page not to display a link back to the original page ([\\$EnableRedirectQuiet](#) variable should be set to 1).

`(:spacewikiwords:), (:nospacewikiwords:)`

Enables/disables automatic spacing of WikiWords in text.

`(:title text:)`

Sets a page's title to be something other than the page's name. The title text can contain apostrophes and other special characters. If there are multiple titles in a page, the last one encountered wins.

`(:nl:)`

Creates a new line if there isn't one already there. See this [thread](#)([approve sites](#)) for more info. Similar to `[[<<]]`

Can I get `(:redirect:)` to return a "moved permanently" (HTTP 301) status code?

Use `(:redirect PageName status=301:)`.

Is there any way to prevent the "redirected from" message from showing at the top of the target page when I use `(:redirect:)`?

From version 2.2.1 on, set in `config.php` `$EnableRedirectQuiet=1;` and in the page `(:redirect OtherPage quiet=1:)` for a quiet redirect.

Is there any method for redirecting to the equivalent page in a different group, i.e. from `BadGroup/thispage` => `GoodGroup/thispage` using similar markup to `(:redirect Goodgroup.{Name}:)`?

`(:redirect Goodgroup.{Name}:)` works if you want to put it in one page.

If you want it to work for the entire group, put `(:redirect Goodgroup.{*Name}:)` into `Badgroup.GroupHeader` - however, that only works with pages that really exist in `Goodgroup`; if you visit a page in `Badgroup` without a corresponding page of the same name in `Goodgroup`, instead of being redirected to a nonexistant page, you get the redirect Directive at the top of the page.

With `(:if exists Goodgroup.{*Name}:)(:redirect Goodgroup.{*Name}:)(:ifend:)` in `Badgroup.GroupHeader` you get redirected to `Goodgroup.Name` if it exists, otherwise you get `Badgroup.Name` without the bit of code displayed.

Retrieved from <https://www-sop.inria.fr/mascotte/EULER/wiki/pmwiki.php/PmWiki/PageDirectives>

Page last modified on March 28, 2009, at 09:27 PM