From EULER Project

PmWiki: Page Lists

PmWiki comes with two directives for generating lists of pages -- `(:pagelist:)` and `(:searchresults:)`. Both directives are basically the same and each accepts the parameters documented below. The primary difference between the two is that searchresults generates the "Results of search for ..." and "### pages found out of ### searched" messages around the results.

The `(:searchbox:)` directive generates a search form (input text box) to submit search queries. The markup generally accepts the same parameters as `(:pagelist:)`, which makes it possible to restrict, order and format searchresults in the same ways that are described below for a `(:pagelist:)`. For more information about the `(:searchbox:)` directive, and the ways in which it differs from a `(:pagelist:)`, skip to the section below.

Basic syntax

- `(:pagelist:)`

without any arguments shows a bulleted list of all pages, as links, ordered alphabetically and in groups.

- `(:pagelist group=`*abc* `name=`*def* `fmt=`*template* `list=`*ghi* `order=`*jkl* `count=`*123* `link=`*mno* `trail=`*pqr* `wrap=`*stu* `passwd=`*vwx* `if=`*yz* `$:`*ptv*`=`*yz* `$`*pv*`=`*za* `cache=0` *argument1* `-`*argument2* *etc* `variable=`*value* `class=`*class* `:)`

shows a pagelist according to the parameters supplied. Parameters are optional.

- `(:searchbox value=`*abc* `size=`*99* `target=`*def*`:)`
- `(:searchresults` *request=1 req=1* `:)`

Parameters

Any argument supplied within `(:pagelist:)` that isn't in the form `'key=value'` is treated as text that either must (or must not) exist in the page text.

The minus sign (-) or exclamation mark (!) can be used to indicate the logical *not*. Thus

`(:pagelist trail=PmWiki.DocumentationIndex list=normal apple !pie:)`
lists all "normal" pages listed in the Documentation Index trail that contain the word "apple" but not "pie".

With page text variables

You can also use page text variables as a *key* to list pages according to the existence of a page text variable. Eg :

`(:pagelist $:pagetextvar=avalue:)`

lists pages having *$:pagetextvar* set to *avalue*.
Minus sign (-), wildcards (?*) and a comma separated list of values also works when specifying a selection based on pagetextvariables. Eg :

`(:pagelist $:apagetextvar=t*,-test:)`
lists pages having $:apagetextvar like 't*' but not 'test'.
Examples:

PTV is set (is not empty):	`(:pagelist $:MyPageTextVariable=- :)`
PTV is empty or not set: (ie, is not set to one char followed by 0 or more chars)	`(:pagelist $:MyPageTextVariable=-?* :)`
PTV is not VALUE:	`(:pagelist $:MyPageTextVariable=-VALUE :)`
PTV is set and not YES:	`(:pagelist $:MyPageTextVariable=?*,-YES :)`

Be aware that if using `(:pagelist $:MyPTV=$:YourPTV :)` PTVs include PmWiki formatting, so you may not get the matches you expect. Currently the only way around this is to use wild cards, so if the formatting is embedded you may be out of luck.

With page variables (PV)

Page variables can be used within pagelists in the same way as page text variables. See Page Text Variables above for more details. Simply use $var instead of $:var.

group= and name=

The "`group=`" and "`name=`" parameters limit results to pages in a specific group or with a specific name:

All pages in the Pmwiki group:	`(:pagelist group=PmWiki :)`
All pages except those in the PmWiki or Site groups:	`(:pagelist group=-PmWiki,-Site :)`
All RecentChanges pages	`(:pagelist name=RecentChanges :)`
All pages except RecentChanges	`(:pagelist name=-RecentChanges :)`

Wildcards

Name and group parameters can contain *wildcard* characters that display only pages matching a given pattern:

- An asterisk (*) represents zero or more characters
- A question mark (?) represents exactly one character

Examples:

All pages in any group beginning with "PmWiki"	`(:pagelist group=PmWiki* :)`
All pages in any group beginning with "PmWiki", except for Chinese	`(:pagelist group=PmWiki*,-PmWikiZh :)`
All pages in the PmCal group with names starting with "2005":	`(:pagelist name=PmCal.2005* :)`

All Cookbooks with names beginning with a A and a B letter
note the different separators used for the same result
	`(:pagelist group=Cookbook name=A*,B* :)`
	`(:pagelist group=Cookbook name="A* B*" :)`
	`(:pagelist group=Cookbook name=[AB]* :)`
	`(:pagelist group=Cookbook, name=[AB]* :)`

If you want to use multiples conditions in name you need to use quotes or commas to delimit the string.

trail=

The "`trail=`" option obtains the list of pages to be displayed from a WikiTrail:

- Display pages in the documentation by modification time

```
(:pagelist trail=PmWiki.DocumentationIndex order=-time:)
```

- Display five most recently changed pages

```
(:pagelist trail=RecentChanges count=5:)
```

list=

The "`list=`" option allows a search to include or exclude pages according to predefined patterns set by the administrator.

- "`list=normal`" is predefined, and which excludes things like AllRecentChanges, RecentChanges, GroupHeader, GroupFooter, GroupAttributes, and the like from being displayed in the list results. Note that list=normal also excludes the current page.
- "`list=all`" over-rides a "default" list that may be set by the wiki's administrator to exclude groups such as PmWiki or Site from regular search results.
- Wiki administrators can define custom lists via the `$SearchPatterns` array (see Cookbook:SearchResults).

fmt=

The "`fmt=`" option determines how the resulting list should be displayed. PmWiki predefines several formats:

- `fmt=#bygroup` - Display pages within groups (default format)

- `fmt=#simple` - Display a simple ordered list of pages in the form Group.Name
- `fmt=#title` - Display a list of pages by page title. Use "`order=title`" to have them sorted by title (default is to order by page name).
- `fmt=#group` - Display a list of wikigroups (without listing the pages in the groups)
- `fmt=#include` - Display the contents of each page in the list (note, this could take a very long time for long lists!)

These formats are defined by page list templates, which can be customized.

This format is not predefined by a page list template:

- `fmt=count` - Display the number of pages in the list (note the absence of the "#").
- `fmt=authtable` - Display a table of pages with *read*, *edit*, *attr*, *upload*, and *publish* settings (note the absence of the "#").

link=

The "`link=`" option implements "backlinks" -- i.e., it returns a list of pages with a link to the target. It's especially useful for category pages and finding related pages.

- all pages with a link to PmWiki.DocumentationIndex

```
(:pagelist link=PmWiki.DocumentationIndex:)
```

- all pages with links to the current page

```
(:pagelist link={$FullName}:)
```

- all pages in the "Skins" category

```
(:pagelist link=Category.Skins:)
```

count=

The "`count=`" option provides the ability to

- limit the pagelist to a specific number of pages
- subsets of a list
- return items from the end of a list, subsets of a list
- display pages in reverse sequence

A simple bullet list of ten most recently modified pages	`(:pagelist trail=Site.AllRecentChanges count=10 fmt=#simple:)`
Display the first ten pages of a list	`count=10 # display the first ten pages of list`
Negative numbers specify pages to be displayed from the end of the list:	`count=-10 # display last ten pages of list`
Ranges may be specified using '..', thus:	`count=1..10 # first ten pages of list` `count=5..10 # 5th through 10th pages of list`

Negative numbers in ranges count from the end of the list:

```
count=-10..-5 # 10th from end, 9th from end, ..., 5th from end
```

Omitting the start or end of the range uses the start or end of the list:

```
count=10..        # skip first ten pages
count=..10        # 1st through 10th page of list
count=-10..       # last ten pages of list
count=..-10       # all but the last nine pages
```

Ranges can be reversed, indicating that the order of pages in the output should likewise be reversed:

```
count=5..10       # 5th through 10th pages of list
count=10..5       # same as 5..10 but in reverse sequence
count=-1..1       # all pages in reverse sequence
```

"Reverse sequence" here refers to the sequence *after* any sorting has taken place. Therefore the three directives to the right are equivalent:

```
(:pagelist order=-name count=10:)
(:pagelist order=-name count=1..10:)
(:pagelist order=name count=-1..-10:)
```

wrap=

The "wrap" option has the values, *none* and *inline*.

With "wrap=inline" and "wrap=none", the output from pagelist (markup or HTML) is directly embedded in a page's markup without any surrounding class=... tags.

With "wrap=inline", any surrounding <ul> is continued. Without "wrap=inline", the HTML output starts a new <ul>. This is important if you want to get a second level <ul> produced by the page list since starting a new <ul> with "**" doesn't yield a second level <ul> but <dl><dd>
...

"wrap=inline" likely has other effects since it suppresses the call to $FPLTemplateMarkupFunction (being MarkupToHTML by default).

class=

By default, a pagelist has the 'fpltemplate' class. The 'bygroup', 'simple', 'group' and 'title' page list formats have specific class names fplbygroup, fplsimple etc. You can set any class using the class= parameter or by setting the $FPLFormatOpt array.

passwd=

The "passwd" option returns only those pages that have some sort of password attribute on them.

if=

The "if" option allows a condition to be specified as part of the pagelist processing, rather than from within the page list template. Only those pages for which the condition is true are retrieved. Anything that could go within an (:if ...:) can be used as a condition. For example

```
(:pagelist if="date {(ftime %GW%V {*$Name})} {=$Name}" :)
```

returns all of the pages where the name is in the same week as that of the current page.

order=

The "order=" option allows the pages in the list to be sorted according to different criteria. Use a minus sign to indicate a reverse sort. Multiple sorting criteria can be specified using a comma, and you can create your own custom pagelist sort order:

- order=name - alphabetically by name (default order)
- order=title - alphabetically by title rather than names
- order=time - most recently changed pages **last**
- order=ctime - time of page creation (see note)
- order=size - page size (not file size), smallest pages first
- order=group,title - by multiple criteria, in this instance sort by title within groups
- order=random - shuffle the pages into random sequence
- order=$:pagetextvarname - alphabetically by page text variable value

Also, the order= option allows custom ordering functions to be written.

Note: fmt=trail results in an unordered pagelist, i.e. the trail order is preserved in the pagelist. So PmWiki's alphabetical default order does not apply in this case.
Note: ctime was added to pages only from pmwiki 2.1.beta15 onwards, pages created by earlier versions don't carry a ctime attribute and can't be sorted that way.

cache=0

Pagelist has the capability to cache lists which greatly speeds up processing. Every once in a while this caching can result in undesired results. Specifying cache=0 disables caching.

Specifying variables as parameters

You can also specify variable values inline with the pagelist statement, and refer to the variables in the template using the {$$variable1} format:

(:pagelist fmt=#pagelist variable1="value" variable2="value2":)
This assumes that a site has $EnableRelativePageVars enabled, which is recommended in PmWiki 2.2.0 -- but disabled by default to help people upgrading from 2.1.x.

For example, in the template:

```
>>comment<<
[[#tvars]]
(:template default count=1 ParamName=Simon:)
Hi, {$$ParamName}, how are you today?
[[#tvarsend]]
>><<
```

(:template default count=1 ParamName=Simon:) Hi, {$$ParamName}, how are you today?
This gives:

(:pagelist fmt=#tvars ParamName="Sam":)	Hi, Sam, how are you today?
(:pagelist fmt=#tvars	Hi, Sally, how are you today?

```
ParamName="Sally":)

(:pagelist fmt=#tvars:)
```

Hi, Simon, how are you today?

Examples

Include the contents of a random page from the Banners group:

```
(:pagelist group=Banners order=random count=1 fmt=#include list=normal:)
```

Display a simple list of the last ten recently changed pages:

```
(:pagelist trail=Site.AllRecentChanges count=10 fmt=#simple:)
```

Display the "top twenty" biggest cookbook pages:

```
(:pagelist group=Cookbook order=-size count=20 :)
```

The Searchbox Directive

The `(:searchbox:)` directive generally accepts the same parameters as `(:pagelist:)` and `(:input text:)` directives:

- Pagelist parameters can be added to the input text of a searchbox (or to the markup, or both)
- Input text box parameters can be added to the searchbox markup
  - ♦ An initial search string can be specified in the searchbox markup, but it must be in the form `value='search string'`. That search string is displayed in the input text and can be modified by when the search is run.
  - ♦ The size of the text input field can be specified with the size parameter, where "size=40" would specify the current default value.
    - ◊ Tip: If more than one searchbox appears on a page, adding a blank initial value like this `value=''`, to the markup for each searchbox will prevent a search string for one box from populating all of the other boxes.
- The target page for displaying searchbox results can be set with the parameter `target=`*`GroupName.PageName`*. The default is the current page.
- The entire searchbox form can be overridden by defining the $SearchBoxFmt variable in one's configuration file. If $SearchBoxFmt is defined, then the parameters to `(:searchbox:)` are ignored, and the content of the $SearchBoxFmt variable are used instead.

The Searchresults directive

The `(:searchresults:)` directive generally accepts the same parameters as `(:pagelist:)` and `(:input text:)` directives:

request=1 req=1

(:searchresults:) without the introductory line that says "Results of search for..."

See Also

- PageDirectives#attachlist - display a list of attachments
- Site.PageListTemplates - default pmwiki pagelist templates

- Cookbook:PagelistTemplateSamples - contributed pagelist template samples
- PageListTemplates - how to create custom pagelist templates for the fmt= option
- PagelistVariables - *local/config.php* customizations
- Cookbook:Forms - documentation for `(:input text:)` markup, which applies to `(:searchbox:)`
- CustomPagelistSortOrder - creating custom order sort functions
- Cookbook:CustomPagelistSortOrderFunctions -