

External Plugins installed

Image Gallery

Syntax

A basic gallery can be added by selecting a [namespace](#) like this:

```
{{gallery>:namespace}}
```

All image files in the selected namespace will be added to the image gallery. Don't forget the ":" in front of the namespace.

Instead of using a whole namespace of images, you can also specify a single image – this makes more sense when combined with the lightbox mode (see below).

```
{{gallery>:namespace:someimage.jpg}}
```

The created gallery can be aligned by using white space (defaults to centered):

```
{{gallery> namespace}} (right aligned)
{{gallery>namespace }} (left aligned)
{{gallery> namespace }} (centered)
```

Instead of a namespace, you can also give an HTTP(s) URL to any [Media RSS](#) or ATOM feed with enclosures (as produced by most photo sharing sites like Flickr). The images will then be pulled from that feed instead:

```
{{gallery>http://www.23hq.com/rss/schabloni}}
```

Note: since the question mark is used to separate the parameters (see next section) the URL can not contain any question mark. To use such a feed URL with the gallery plugin, just use one of the many short URL services like <http://bit.ly>.

E.g. instead of

http://api.flickr.com/services/feeds/photos_public.gne?id=22019303@N00&lang=en-us&format=rss_200 use a shortened URL like <http://bit.ly/HurZM>.

Additionally, to have thumbnail creation correctly working you need to set [fetchsize](#) big enough to get the remote images downloaded.

Parameters

A number of parameters can be set by appending them with ? character to the namespace or image. Each parameter needs to be separated with a & character. Defaults for all parameters can be set in

the config manager. If a parameter is enabled by default it can be disabled in the syntax by prefixing it with the syllable no. E.g. the parameter cache is usually enabled and can be disabled using the keyword nocache. Below is a list of all recognized parameters

Parameter	Default	Description
<code><number>x<number></code>	120x120	Sets the size for thumbnails. Unless the crop option is set, this is a boundary box into which the thumbnail will be fitted, maintaining the correct aspect ratio.
<code><number>X<number></code>	800X600	Sets the size for the linked images in direct mode. This is a boundary box into which the image will be fitted, maintaining the correct aspect ratio. Note the uppercase X.
<code><number></code>	5	The number images per row in the gallery table. If you specify a 0 no table is used instead all thumbnails are added in a sequence.
<code>=<number></code>	=0	Limits the output to the given number of images. 0 means all.
<code>+<number></code>	+0	Skip the first number of images. Useful with the option above.
<code>~<number></code>	~0	Add a pagination for the thumbnails displaying the number of given thumbnails per page. 0 disables pagination. Pagination is added through JavaScript - when no JavaScript is available all thumbnails are displayed
cache	enabled	Usually the output of the created gallery is cached. When the images in your selected namespace change, you have to manually force an update of the gallery page's cache . To disable the cache for the page showing the gallery, set nocache as option.
crop	disabled	Make thumbnails the exact given thumbnail size big, cropping when needed.
direct	disabled	Link thumbnails with the bigger sized images not with their detail page
lightbox	disabled	Show images in a fancy JavaScript modal browsing window, see below for details. Setting this option automatically implies the direct mode
reverse	disabled	Reverse the order of the displayed images
recursive	enabled	Find images in the given namespace and all sub namespaces
random	disabled	Sort images randomly. You might want to use nocache as well
modsort	disabled	Sort images by file modification date
datesort	disabled	Sort images by EXIF date
titlesort	disabled	Sort images by EXIF title
showname	disabled	Show filename below thumbnails
showtitle	disabled	Show the EXIF tag Headline as title below thumbnails
anything containing a *	jpg,gif,png images	This can be used to filter the list of files found in the given namespace. * work as simple wildcard symbol.

Example

```
{{gallery>images:vacation?image_*.jpg&80x80&crop&lightbox}}
```

This displays all images beginning with `image_` and ending in `.jpg` from the namespace `images:vacation`. Thumbnails are cropped to 80×80 pixels and images will be opened in lightbox mode.

Example

```
{{gallery>?crop&lightbox}}
```

This displays all images in the current namespace using 2 parameters. Parameterlist begins with `?` additional ones are concatenated with `&`.

About the Lightbox mode

This mode will open the clicked picture inside the current browser window without leaving the current page.

You can close the picture view by clicking the X button in the upper right corner. You can move to the next or previous image by using the arrow buttons in the lower bar. You can also use the keyboard shortcuts listed below for navigation and closing. Mobile users can swipe to navigate and use the back button to close the gallery.

The following keys can be used to navigate:

Key	Action
→	next image
←	previous image
ESC	close the image view

Manipulate EXIF Tags

If you want to show a title below the image using the parameter `showtitle` it may appear that the title shown is not as you expected (e.g. `einstein.jpg` instead of `Albert Einstein`).

If you want to adapt the image's subtitle shown in the gallery, you have to manipulate the EXIF Tag called `Headline`. This can be done with the [fullscreen media manager](#). Alternatives include [ExifTool](#) for which even a [Windows GUI](#) exists to adapt the EXIF data to your needs. On *Windows*, the small picture viewer [IrfanView](#) can also be used to change only IPTC values.

Note that the EXIF-tags used for display can be changed. See [EXIF and IPTC metadata](#) for details. By default, the following tags are used:

- `IPTC:Headline` for the title
- `IPTC:Caption-Abstract` for the description below the picture in lightbox-mode
- `IPTC:By-line` for the photographer name
- `IPTC:CopyrightNotice` for the copyright
- `IPTC:Keywords` for keyword list

Known Limitations and Caveats

Uploading Images

Uploading images is beyond the scope of this plugin. Do not request any features regarding this.

- Use the [media manager popup](#) or [fullscreen media manager](#) to upload multiple image at once with compatible browsers, otherwise one by one.
- Use the [archiveupload](#) plugin to upload multiple images in a Zip file
- Upload the files manually via FTP to the data/media directory. Keep in mind that image names need to be valid [pagenames](#), all lowercase, no spaces or special chars!

Caching

The gallery output is cached by default. When you add pictures later, they may not show up in the gallery: add `&purge=true` to the end of the URL to clear the cache. See [caching](#) for details.

Optionally use the `nocache` parameter of the plugin (not recommended).

EXIF Data Problems

Problems with accessing [EXIF or IPTC](#) data in the images, should be reported as DokuWiki bugs and not for this plugin. All EXIF handling is in DokuWiki core. Currently EXIF Data is expected in UTF-8 encoding.

Lightbox Problems

When the lightbox mode doesn't work and instead images are simply opened in the same window, the JavaScript was not correctly loaded. This is most likely a Browser-Cache issue. Simply follow the steps described for fixing a similar problem with the [toolbar](#). Also make sure you don't have any conflicting plugin installed. You should **not** install any additional lightbox plugin.

Images are not Resized

There are different reasons why thumbnails are not created:

- libGD extension is missing ⇒ Install the extension or configure DokuWiki to use [imagemagick](#)
- libGD extension is installed, but the source image size + overhead is larger than `memory_limit` set in `php.ini` ⇒ Lower the source image size or increase the `memory_limit`

MediaRSS feed displays "nothing found"

When no images from your feed are shown, be sure you don't have a question mark in your URL. Use

an URL shortener as suggested above. Also be sure your feed URL (before shortening) starts with `http://` or `https://` and not with `feed://`. The latter is just a renamed HTTP link - simply rename it back.

External Images are not Resized

As written above, you need to increase the `fetchsize` config option. Also make sure LibGD or ImageMagick are installed.

Demo Installations

Here are a few user provided examples of the gallery plugin in use:

- <http://www.wikepixes.com.br/> 
- <http://eolienne.f4jr.org/medias/start>
- http://www.splitbrain.org/blog/2008-01/04-new_york_impressions
- <http://www.hemmerling.com/doku.php/en/repository.html>
- <http://urbangirlblue.org/ecscd/doku.php> 

Discussion

Be sure to read the [Known Limitations and Caveats](#) section first. Questions should be asked in the [forum](#). Bugs and feature wishes should be reported in the [issue tracker](#). Any comments/discussions on this page will be removed.

Wrap Plugin

One plugin to rule them all

This plugin gives you the ability to wrap wiki text inside containers (divs or spans) and give them

1. a certain class (with loads of useful preset classes)
2. a width
3. a language with its associated text direction

It potentially replaces a lot of other plugins and is IMHO the better alternative for many.

It fully replaces: [class](#), [clearfloat](#), [div_span_shorthand](#), [divalign2](#), [divalign](#), [emphasis](#), [hide](#), [important_paragraph](#), [importanttext](#), [lang](#), [ltr](#), [noprnt](#), [pagebreak](#), [side_note](#), [tip](#), [wpre](#)

It partly replaces: [box](#), [button](#), [color](#), [columns](#), [fontcolor](#), [fontfamily](#), [fontsize2](#), [fontsize](#), [highlight](#), [layout](#), [note](#), [styler](#), [tab](#), [tablewidth](#), [typography](#)

Syntax

Basic Syntax:

```
<WRAP classes #id width :language>
"big" content
</WRAP>
```

```
or
<block classes #id width :language>
"big" content
</block>
```

```
or
<div classes #id width :language>
"big" content
</div>
```

An uppercase **<WRAP>** (or alternatively **<block>** or **<div>**) creates a **div** and should be used for **“big”** containers, **surrounding** paragraphs, lists, tables, etc.

```
<wrap classes #id width :language>"small" content</wrap>

or
<inline classes #id width :language>"small" content</inline>

or
<span classes #id width :language>"small" content</span>
```

A lowercase **<wrap>** (or alternatively **<inline>** or ****) creates a **span** and should be used for **“small”** containers, **inside** paragraphs, lists, tables, etc.

Since version 2013-06-13 there is also a shorthand syntax (for wraps without content):

```
<WRAP classes #id /> or <block classes #id /> or <div classes #id />
```

and

```
<wrap classes #id /> or <inline classes #id /> or <span classes #id />
```

⚠ Please note, some things **won't work with spans**: **alignments** (including alignments generated by changing the text direction), **multi-columns** and **widths** if the according wrap isn't floated as well.

Examples

The plugin comes with an example page, which should explain a lot and looks like this in the default template (see below).

Classes

The following classes are currently available:



class name	description/notes
columns - similar to columns , side_note , styler , tip	
column	same as <code>left</code> in LTR languages and same as <code>right</code> in RTL languages
left	same as <code>column</code> , will let you float your container on the left
right	will let the container float right
center	will position the container in the horizontal center of the page
col2..col5	will show the text in multiple columns (2, 3, 4 or 5), only works in modern browsers (Firefox, Chrome and Safari)
widths - experimental, might not work as expected, includes mobile support	
half	fits two columns in a row, should be used in pairs
third	fits three columns in a row, should be used in triplets
quarter	fits four columns in a row, should be used in quads
alignments - similar to divalign , columns , styler - don't work with spans!	
leftalign	aligns text on the left
rightalign	aligns text on the right
centeralign	centers the text
justify	justifies the text
boxes and notes - similar to box , note , tip	
box	creates a box around the container (uses colours from <code>style.ini</code>)
info (was information in first version)	creates a blue box with an info icon
important	creates an orange box with an important icon
alert (was warning in previous versions)	creates a red box with an alert icon
tip	creates a yellow box with a tip icon
help	creates a violet box with a help icon
todo	creates a cyan box with an todo icon
download	creates a green box with a download icon
round	adds rounded corners to any container with a background colour or a border (only works in modern browsers, i.e. no IE)
danger	creates a red danger safety note
warning	creates an orange warning safety note
caution	creates a yellow caution safety note
notice	creates a blue notice safety note
safety	creates a green safety note
marks - similar to emphasis , important_paragraph , importanttext	
hi	marks text as highlighted
lo	marks text as less significant
em	marks text as especially emphasised
miscellaneous	

class name	description/notes
clear	similar to clearfloat , should preferably be used with divs, i.e. uppercase <code><WRAP>s</code>
tabs	if wrapped around a list of links, will show those as tabs
hide	hides the text per CSS (the text will still appear in the source code, in non-modern browsers and is searchable)
noprint	displays text on the screen, but not in print, similar to noprint
onlyprint	displays text only in print, but not on the screen
pagebreak	forces a new page in printouts (not visible on the screen), similar to pagebreak
nopagebreak	tries to avoid a pagebreak in printouts (not visible on the screen)
spoiler	shows white text on a white background, only to be revealed by highlighting it; similar to hide
button	when wrapped around a link, styles it like a button
indent	indents the text, could be used instead of tab
outdent	"outdents" the text, could partly be used instead of outdent
prewrap	wraps text inside pre-formatted code blocks, similar to wpre

All tables inside a column or box will always be 100% wide. This makes positioning and sizing tables possible and partly replaces [tablewidth](#).

Known restrictions

- WRAPs won't export in ODT format.
- Round corners only work in modern browsers (no IE8 and below).
- Multiple columns only work in modern browsers (no IE9 and below).
- Width classes are experimental and only work in modern browsers (no IE8 and below).
- Normal DokuWiki Headlines used to not work and a work-around was added. Now that headlines do work, the work-around is not needed anymore but kept for backwards-compatibility. The following syntax would produce two different kinds of emulated headlines inside any of the columns or boxes/notes:
 - `/** __Big Underlined Headline__ */` (They will look a bit different in safety notes.)
 - `/**Small Headline**/`

You might need to adjust a few of the classes to your template's needs, especially `hi`, `lo` and `em`. If you have a dark or otherwise heavily coloured theme, please use the `darkTpl` [config option](#).

The classes are easily adjustable and extensible. Any wishes are welcome.

Widths

You can set any valid widths on any uppercase `<WRAP>` container: `%`, `px`, `em`, `ex`, `pt`, `pc`, `cm`, `mm`, `in`. Just set the width before or after or with the classes, e.g.

```
<WRAP someclass 50% anotherclass>...
```

All except percentages will be reduced to have the maximum width available on smaller screens.

You can also use the width keywords `half`, `third` and `quarter`. To work correctly they need another wrap around them. E.g.

```
<WRAP group>
  <WRAP half column>...</WRAP>
  <WRAP half column>...</WRAP>
</WRAP>
```

will result in two columns next to each other, which will wrap underneath each other on smaller screens and mobile devices.

Languages and Text Directions

You can change the language and the direction of a container by simply adding a colon followed by the language code, like this:

```
<wrap :en>This text is explicitly marked as English.</wrap>
```

The text direction (`rtl`, right to left or `ltr`, left to right) will get inserted automatically and is solely dependent on the language. The list of currently supported languages is taken from:

http://meta.wikimedia.org/wiki/Template:List_of_language_names_ordered_by_code

If you like to mark a text with a different text direction than the default one, you should use divs, i.e. uppercase `<WRAP>`s. Otherwise the text alignment won't change as well.

This makes it a better replacement of `ltr` (and `lang`).

Demo

You can see a demo of the plugin on demo.selfthinker.org.

[“Examples” \(demo\) in Russian](#) (for v2011-05-15). [Source](#).

Configuration options

Option	Description	Default value
<code>noPrefix</code>	Which (comma separated) class names should be excluded from being prefixed with “wrap_”	<code>tabs, group</code>
<code>restrictedClasses</code>	restrict usage of plugin to these (comma separated) classes	<i>(empty)</i>
<code>restrictionType</code>	restriction type, specifies if classes above shall be included or excluded	<code>0</code>
<code>syntaxDiv</code>	Which syntax should be used in the toolbar picker for block wraps?	<code>WRAP</code> (other choices: <code>div</code> , <code>block</code>)
<code>syntaxSpan</code>	Which syntax should be used in the toolbar picker for inline wraps?	<code>wrap</code> (other choices: <code>span</code> , <code>inline</code>)

Option	Description	Default value
darkTpl	Optimise colours for dark templates?	0

ODT Support

Since version 2015-06-13 the Wrap plugin supports exporting most of its functionality/styling to ODT when using at least version 2015-06-29 of the [ODT Plugin](#). By default, Wrap syntax will be exported to ODT using 'print' CSS styles. This means the exported Wrap elements will look the same when printing a wiki page. If you want to have the ODT exported Wrap elements look like displayed in the browser (i.e. with 'screen' CSS styles), then use the following ODT plugin configuration settings:

- add wrap to the 'usestyles' config setting
- set the 'media_sel' setting to 'screen'

If you prefer a user defined CSS style for the Wrap ODT export, then simply place a file 'odt.css' into the Wrap plugin folder with your own CSS code (and set config setting 'media_sel' to 'print').

Here is what is currently **not** supported:

- columns: left/right/center/column is partly supported; they are positioned correctly, but content is not floating around them
- widths are not supported except % and half/third/quarter
- boxes and notes: hardly any formatting inside them is supported, therefore emulated headings also don't work
- tabs will just show a list of links
- noprint
- nopagebreak
- onlyprint only works on boxes
- languages are set correctly but do not seem to affect text alignment
- shorthand syntax
- Not supported because not relevant in ODT: clear, prewrap

Toolbar picker



The wrap picker in the editing toolbar adds the most common wrap syntaxes.

- “columns” creates a set of half columns
- “simple centered box” creates a standard box (60% wide, centered)
- “info, tip, important, alert, help, download, todo box” creates specifically themed boxes (also 60% wide, centered)
- “clear floats” creates a `<WRAP clear></WRAP>`
- “especially emphasised, highlighted, less significant” creates the respective marks

Extend with custom styles

If you like to add your own classes and styles to the plugin, you can simply add the styles for your class preceded by “wrap_” to your [user styles](#). Please note, any classes need to be **lower case**.

E.g. if you need a <WRAP myclass>, you edit (or create if it doesn't exist) your conf/userstyle.css and add your .wrap_myclass{} with its style definitions to it. (If necessary, edit conf/userprint.css¹⁾ for the print view, conf/userrtl.css²⁾ for RTL languages and conf/userall.css³⁾ for all styles as well.)

User permissions for every file used must be similar to original DokuWiki files.

Since version 2010-03-14 you have the possibility to exclude certain class names from being prefixed with “wrap_”. Just add a comma separated list of class names into the config option “noPrefix” in the configuration manager.

Examples

in style.css

```
.dokuwiki div.wrap_note{ /* added */
    background-color: #eee;
    color: #000;
    padding: .5em .5em .5em .5em;
    margin-bottom: 1em;
    overflow: hidden;
}
```

call in DW-page:

```
<WRAP note>...</WRAP>
```

Here are some [useful wrap extensions](#) created by users of this plugin.

Add former typography classes

The old typography classes were removed in version 2011-05-15. If you need something similar, use the [Block](#) plugin instead. Or you can use your own and copy them to your own user styles (see [above](#)).

How to use the helper

From version 2011-05-15 on the plugin includes a helper plugin which you can use to add classes, width and lang/dir to any other plugin.

Example how to get just one kind of attribute

```
// get lang from wrap helper plugin
$lang = '';
if(!plugin_isdisabled('wrap')) {
    $wrap = plugin_load('helper', 'wrap');
    $attr = $wrap->getAttributes($data);
    if($attr['dir']) $lang = ' lang="'. $attr['lang']. '"
xml:lang="'. $attr['lang']. '" dir="'. $attr['dir']. '"';
}

// add lang to your plugin's output
$renderer->doc .= '<span ' . $lang . ' class="foo">';
```

`getAttributes()` expects the string with “classes #id width :language”. It returns an array with

- `$attr['class']`: CSS class(es)
- `$attr['id']`: CSS ID
- `$attr['width']`: width
- `$attr['lang']` and `$attr['dir']`: language and text direction

Example how to get all attributes

```
// get attributes from wrap helper plugin
$attr = '';
if(!plugin_isdisabled('wrap')) {
    $wrap = plugin_load('helper', 'wrap');
    $attr = $wrap->buildAttributes($data, 'additionalClass');
}

// add those attributes to your plugin's output
$renderer->doc .= '<div ' . $attr . ">";
```

`buildAttributes()` expects the same string as above (“classes #id width :language”) and an optional string for additional classes, in case your plugin has CSS classes of its own which it needs to reuse. It returns a string with all the attributes prepared for HTML.

Development

Done



- [Merge pull request #278 from dokuwiki-translate/lang_update_827_17092...](#) (2024/02/29 21:51)
- [translation update](#) (2024/02/29 19:25)
- [Merge pull request #273 from dokuwiki-translate/lang_update_708_16959...](#) (2023/09/28 12:00)
- [translation update](#) (2023/09/28 11:50)
- [Merge pull request #271 from dokuwiki-translate/lang_update_694_16919...](#) (2023/08/14 18:39)
- [translation update](#) (2023/08/13 22:55)

- [Update plugin.info.txt](#) (2023/08/13 12:26)
- [Merge pull request #250 from saschaleib/saschaleib-patch-language-dir](#) (2023/08/13 12:14)

Localization

You can help me with translations and [update the language files](#). There are two files to translate:

- [lang/en/lang.php](#) is for the explaining titles on the picker images.
- [lang/en/settings.php](#) is for the configuration options.

Credits

- The code reuses parts of the [box](#) plugin by [Christopher Smith](#).
- The ODT support was implemented by [LarsDW223](#).
- The images for the notes are taken from the [Human-O2 icon set](#).
- The toolbar uses images from the [Silk Icon Set](#) and the [Silk Companion Icon Set](#).
- Thanks to [all contributors](#).

PageList

Description

The Pagelist Plugin takes a list of wiki pages and provides a nicely formatted table with information about them. The plug-in has a number of flags that can be used to control the information and format of the page list. The user can provide a list of specific page references as can some popular helper plugins such the [Blog](#), [Discussion](#), [Editor](#), [Tag](#), [Task](#) and [Dir](#) plugins.

Syntax

Just wrap a regular unordered list of internal links with the `<pagelist>` tag. You may provide specific internal page references or have plug-ins supply them as in the example below:

```
<pagelist&[flags]>
* [[...:blog:|Blog Plugin]]
* [[...:discussion:|Discussion Plugin]]
* [[...:editor:|Editor Plugin]]
* [[...:tag:|Tag Plugin]]
* [[...:wrap|Wrap Plugin|This is shown in the description cell]]
</pagelist>
```

[flags]	flags can be used to alter the appearance of the pagelist, flags	optional
----------------	--	----------

Flags

Setting	Default		Alternative	
style	default	table with horizontal lines	table, list or simplelist	standard DokuWiki table or list style
showheader	noheader	hide the heading row of the pagelist table	header	show the header
showdate	date	show the creation or last modification date	nodate	hide the date
showuser	user	show creator or contributors	nouser	hide the user
showdesc	nodesc	hide the description	desc	show the description (from metadata)
showcomments	nocomments	hide the number of comments	comments	show the number of comments (if Discussion Plugin is installed)
showtags	notags	hide the tags	tags	show the tags (if Tag Plugin is installed)
showfirsthl	firsthl	show the first headline	nofirsthl	show the page name
rsort/sort	nosort	no sortation of pages	rsort/sort	sorts the pages (reverse) alphabetically by pagename
showdiff	missing	no displaying of differences column	showdiff	displays the differences column with the diff icon linking to the corresponding diff page for each row

Example

```
<pagelist&header&comments>
  //an unordered list of pages to display//
</pagelist>
```

In the example above, pagelist will display information about the provided pages in a table with a header line and a comments column (if the [Discussion Plugin](#) is installed). The user (or a plugin) must supply the specific pages to display in the list.

Configuration

The plugin can be configured using the DokuWiki configuration manager available in the admin menu. The settings also apply to plugins which use the helper component of the pagelist plugin, like for example the archive component of the [blog](#) plugin.

style	List style (default, table, table/list, simplelist)
showheader	Show table header
showdate	Shows/hides the date column (hide, creation date, modification date)
showuser	Shows/hides the user column (hide, creator, contributors)
showdesc	Shows/hides a short description taken from the first paragraph of a page (hide, max. 160 characters, max. 500 characters)

showcomments	Shows/hides comments of a page (requires the discussion plugin)
showlinkbacks	Shows/hides linkbacks of a page (requires the linkback plugin)
showtags	Shows/hides tags of a page (requires the tag plugin)
sort	Sorts the pages alphabetically by pagename
showdiff	Displays a differences column with the diff icon linking to the corresponding diff page for each row

Helper Plugin

You can easily use the functionality of the Pagelist Plugin in your own plugins. Here is a basic code snippet:

```
$pages = array(
    array('id' => 'wiki:dokuwiki'),
    array('id' => 'wiki:syntax'),
);
$pagelist =& plugin_load('helper', 'pagelist');
if (!$pagelist) return false; // failed to load plugin
$pagelist->startList();
foreach ($pages as $page){
    $pagelist->addPage($page);
}
$renderer->doc .= $pagelist->finishList();
```

Bugs / Feature Requests

Please report bugs or feature requests at the [Bug tracker](#).

Further Resources

- [Git Repository](#)
- [Bug Tracker](#)

Changes

- [Merge pull request #161 from dokuwiki-translate/lang_update_808_17065...](#) by Klap-in (2024/01/30 15:33)
- [translation update](#) by Impeck (2024/01/30 07:10)
- [Merge pull request #160 from dokuwiki-translate/lang_update_797_17062...](#) by Klap-in (2024/01/27 13:37)
- [translation update](#) by alexgearbox (2024/01/26 06:55)
- [Merge pull request #158 from dokuwiki-translate/lang_update_782_17043...](#) by Klap-in (2024/01/05 12:55)
- [translation update](#) by alexgearbox (2024/01/04 00:00)

- [Merge pull request #157 from dokuwiki-translate/lang_update_740_17007...](#) by Klap-in (2023/11/23 09:07)
- [Merge pull request #155 from dokuwiki-translate/lang_update_712_16963...](#) by Klap-in (2023/11/23 09:07)

Localization

Discussion

This is a place for general plugin discussion. **Please submit bugs and feature requests to bug tracker linked on this page.**

Suppressing the new thread

Is it correct that I need to comment out the renderer if I want to prevent a “create this page” under the commentlist? ⁴⁾

```
// show form to start a new discussion thread?  
if ($perm_create && ($this->getConf('threads_formposition') ==  
'bottom'))  
//          $renderer->doc .= $this->_newThreadForm($ns);  
return true;
```


You should set the configuration option `threads_formposition` to `off` instead. — *Phil* 2010/01/29 16:05


How to add two columns?


I want to add two columns for a plugin. But the function `addColumn` only support one column for one plugin. How can I do this?


need the same, very important!! — *alfiox* 2010-12-22

Also need. If the “list” layout used html “list” syntax instead of being buried inside a table with invisible lines, I think it could interact with the wrap plugin to make columns. — *Ashtagon* 2011-07-28


That is quite complicated with the current layout of the plugin. The data storage isn't built to fetch multiple columns of another plugin. —  *M. S.* 2011/07/29 20:32

If the pagelist plugin (or more important for my needs, the tag plugin, which I think uses the same engine) had an option to return an actual list rather than a table that contains cells which are rendered to look like a list, the problem could then be solved by passing this plugin's output through the wrap plugin's `<div col3>{{topic>your flags here}}</div>` syntax. —  [Ashtagon](#) 2011/08/20 12:00

I went through the code. Inserting between 4 and 5 if statements that cause it to use an unordered list when `$this->style == 'list'` fixed it for me. I also had to ensure that it was only adding things when it should. But, overall, it wasn't that big of a deal. EDIT: I figured you guys would want to see it. Also, I am using the wrap plugin, so I edited helper.php to be compatible with that, it doesn't natively allow you to use multiple columns. <http://pastebin.com/1HDytYdh> WARNING: I only tested it using just the page name; I didn't use date or user or anything as I am using pagelist and tag to simulate the mediawiki category pages. YMMV and it might not work with other configs. Good luck! —  [serenewaffles](#) 2012/01/07 22:18


I commented on the open bug on the tracker with explanation and pastebin —  [serenewaffles](#) 2012/01/07 22:27

<https://github.com/dokufreaks/plugin-pagelist/issues/15>

Thanks for sharing your experiences. Could you please use the option “Minor Changes” when editing multiple times in a row, otherwise it generates a mail for every saving process. —  [M. S.](#) 2012/01/07 22:58

Showing "Edit summary"

How to show the “edit summary” of the latest edit? — *Flavius*

[changes](#) —  [M. S.](#) 2011/02/15 19:03

pagelist: howto specify an additional argument for "date"-flag?

I would like to use pagelist together with “changes”-plugin to list those changes (within a namespace) that have been made on a specific date (e.g. 2011/01/13). How can i tell pagelist to do that? Thanks for help in advance! — *Peter*

Pick up descriptions from Description Plugin

If you want to pick up the Meta-fields from the Description Plugin you have to patch this function in the pagelist-plugin:


```
function _descCell() {
    /*Begin patch to get meta data from plugin description */
    $desc = $this->_getMeta(array('plugin_description', 'keywords'));
    if (!$desc)
    { /* end patch */
        if (array_key_exists('desc', $this->page)) {
            $desc = $this->page['desc'];
        } elseif (strlen($this->page['description']) > 0) {
            // This condition will become true, when a page-description
```

is given

```

        // inside the syntax-block
        $desc = $this->page['description'];
    } else {
        $desc = $this->_getMeta(array('description', 'abstract'));
    }
    } //close patch
    $max = $this->column['desc'];
    if (($max > 1) && (utf8_strlen($desc) > $max)) $desc =
utf8_substr($desc, 0, $max).'...';
    return $this->_printCell('desc', hsc($desc));
}


```

—  Harald 2014/01/19 22:27

pagelist: howto show parent namespace

I am using pagetag to display a list of pages that contain a given tag. As now, only the title of the page is displayed: additionally I want to display (at least) the first level containing namespace (ie: the parent). How do I do that ?

I have updated <https://github.com/dokufreaks/plugin-pagelist/issues/26> with an example because I guess my request was not explicit enough !

 martinus 2013/03/28 11:58

I made additions to the plugin for my personal use, maybe it can help you.


The additions are:

- Displays the **parent namespace** (with the title or name of the page) (flag: **parentns** or **noparentns**)
- **French translation**

I do translations for English and French only. All changes are placed after the following pattern coment :

```
// MODIF : date
```

link to source : box.com

 Sphaira /2013/01/16 11:33

comments not updating

I am unsure if there is a conflict. I have pagelist control settings to how i want, however the comments don't seem to want to update when a comment is added.

1)

conf/printstyle.css in Anteater

2)

conf/rtlstyle.css in Anteater

3)

conf/allstyle.css in Anteater

4)

/lib/plugins/discussion/syntax/threads.php

From:

<https://www.ff6hacking.com/wiki/> - **ff6hacking.com wiki**

Permanent link:

<https://www.ff6hacking.com/wiki/doku.php?id=wiki:explugins&rev=1457315732>

Last update: **2019/02/12 07:59**

