

Wiki Macros

Trac macros are plugins to extend the Trac engine with custom 'functions' written in Python. A macro inserts dynamic HTML data in any context supporting [WikiFormatting](#).

Another kind of macros are [WikiProcessors](#). They typically deal with alternate markup formats and representation of larger blocks of information (like source code highlighting).

Using Macros

Macro calls are enclosed in two *square brackets*. Like Python functions, macros can also have arguments, a comma separated list within parentheses.

Examples

```
[[Timestamp]]
```

Display:

Sun Sep 5 14:16:16 2010

```
[[HelloWorld(Testing)]]
```

Display:

Hello World, args = Testing

Available Macros

Note that the following list will only contain the macro documentation if you've not enabled `-OO` optimizations, or not set the `PythonOptimize` option for `mod_python`.

```
[[InterTrac]]
```

Provide a list of known [InterTrac](#) prefixes.

```
[[TitleIndex]]
```

Inserts an alphabetic list of all wiki pages into the output.

Accepts a prefix string as parameter: if provided, only pages with names that start with the prefix are included in the resulting list. If this parameter is omitted, all pages are listed.

```
[[RecentChanges]]
```

Lists all pages that have recently been modified, grouping them by the day they were last modified.

This macro accepts two parameters. The first is a prefix string: if provided, only pages with names that start with the prefix are included in the resulting list. If this parameter is omitted, all pages are listed.

The second parameter is a number for limiting the number of pages returned. For example, specifying a limit of 5 will result in only the five most recently changed pages to be included in the list.

```
[[PageOutline]]
```

Displays a structural outline of the current wiki page, each item in the outline being a link to the corresponding heading.

This macro accepts three optional parameters:

- ◇ The first is a number or range that allows configuring the minimum and maximum level of headings that should be included in the outline. For example, specifying "1" here will result in only the top-level headings being included in the outline. Specifying "2-3" will make the outline include all headings of level 2 and 3, as a nested list. The default is to include all heading levels.
- ◇ The second parameter can be used to specify a custom title (the default is no title).
- ◇ The third parameter selects the style of the outline. This can be either `inline` or `pullout` (the latter being the default). The `inline` style renders the outline as normal part of the content, while `pullout` causes the outline to be rendered in a box that is by default floated to the right side of the other content.

`[[Image]]`

Embed an image in wiki-formatted text.

The first argument is the file specification. The file specification may reference attachments or files in three ways:

- ◇ `module:id:file`, where `module` can be either **wiki** or **ticket**, to refer to the attachment named *file* of the specified wiki page or ticket.
- ◇ `id:file`: same as above, but `id` is either a ticket shorthand or a Wiki page name.
- ◇ `file` to refer to a local attachment named 'file'. This only works from within that wiki page or a ticket.

Also, the file specification may refer to repository files, using the `source:file` syntax (`source:file@rev` works also).

The remaining arguments are optional and allow configuring the attributes and style of the rendered `` element:

- ◇ `digits` and `unit` are interpreted as the size (ex. 120, 25%) for the image
- ◇ `right`, `left`, `top` or `bottom` are interpreted as the alignment for the image
- ◇ `nolink` means without link to image source.
- ◇ `key=value` style are interpreted as HTML attributes or CSS style indications for the image. Valid keys are:
 - ◇ `align`, `border`, `width`, `height`, `alt`, `title`, `longdesc`, `class`, `id` and `usemap`
 - ◇ `border` can only be a number

Examples:

```
[[Image(photo.jpg)]]           # simplest
[[Image(photo.jpg, 120px)]]    # with size
[[Image(photo.jpg, right)]]    # aligned by keyword
[[Image(photo.jpg, nolink)]]   # without link to source
[[Image(photo.jpg, align=right)]] # aligned by attribute
```

You can use `image` from other page, other ticket or other module.

```
[[Image(OtherPage:foo.bmp)]]    # if current module is wiki
[[Image(base/sub:bar.bmp)]]    # from hierarchical wiki page
```

```
[[Image(#3:baz.bmp)]]          # if in a ticket, point to #3
[[Image(ticket:36:boo.jpg)]]
[[Image(source:/images/bee.jpg)]] # straight from the repository!
[[Image(htdocs:foo/bar.png)]]  # image file in project htdocs dir.
```

Adapted from the Image.py macro created by Shun-ichi Goto <gotoh@taiyo.co.jp>

[[MacroList]]

Displays a list of all installed Wiki macros, including documentation if available.

Optionally, the name of a specific macro can be provided as an argument. In that case, only the documentation for that macro will be rendered.

Note that this macro will not be able to display the documentation of macros if the PythonOptimize option is enabled for mod_python!

[[TracIni]]

Produce documentation for Trac configuration file.

Typically, this will be used in the [TracIni](#) page. Optional arguments are a configuration section filter, and a configuration option name filter: only the configuration options whose section and name start with the filters are output.

[[HelloWorld]]

Example macro.

[[Timestamp]]

Inserts the current time (in seconds) into the wiki page.

[[TracGuideToc]]

This macro shows a quick and dirty way to make a table-of-contents for a set of wiki pages.

[[TicketQuery]]

Macro that lists tickets that match certain criteria.

This macro accepts two parameters, the second of which is optional.

The first parameter is the query itself, and uses the same syntax as for `query`: wiki links (but **not** the variant syntax starting with "?").

The second parameter determines how the list of tickets is presented: the default presentation is to list the ticket ID next to the summary, with each ticket on a separate line. If the second parameter is given, it must be one of:

◇ **compact** -- the tickets are presented as a comma-separated list of ticket IDs.

◇ **count** -- only the count of matching tickets is displayed

[[Branches]]

Render a list of available branches.

[[Screenshot]]

Allows embed screenshot image in wiki page. First mandatory argument is ID of the screenshot. Number or image attributes can be specified next:

◇ `align` - Specifies image alignment in wiki page. Possible values are: `left`, `right` and `center`.

◇ `alt` - Alternative description of image.

◇ `border` - Sets image border of specified width in pixels.

◇ `class` - Class of image for CSS styling.

- ◇ `description` - Brief description under the image. Accepts several variables (see below).
- ◇ `format` - Format of returned image or screenshot behind link.
- ◇ `height` - Height of image. Set to 0 if you want original image height.
- ◇ `id` - ID of image for CSS styling.
- ◇ `longdesc` - Detailed description of image.
- ◇ `title` - Title of image.
- ◇ `usemap` - Image map for clickable images.
- ◇ `width` - Width of image. Set to 0 if you want original image width.

Attribute `description` displays several variables:

- ◇ `$id` - ID of image.
- ◇ `$name` - Name of image.
- ◇ `$author` - User name who uploaded image.
- ◇ `$time` - Time when image was uploaded.
- ◇ `$file` - File name of image.
- ◇ `$description` - Detailed description of image.
- ◇ `$width` - Original width of image.
- ◇ `$height` - Original height of image.

Example:

```
[[Screenshot(2,width=400,height=300,description=The $name by $author: $description,align=left)]]
[[ScreenshotsList]]
```

Displays list of all available screenshots on wiki page. Accepts one argument which is template for list items formatting. Possible variables in this template are:

- ◇ `$id` - ID of image.
- ◇ `$name` - Name of image.
- ◇ `$author` - User name who uploaded image.
- ◇ `$time` - Time when image was uploaded.
- ◇ `$file` - File name of image.
- ◇ `$description` - Detailed description of image.
- ◇ `$width` - Original width of image.
- ◇ `$height` - Original height of image.
- ◇ `$tags` - Comma separated list of screenshot tags.
- ◇ `$components` - Comma separated list of screenshot components.
- ◇ `$versions` - Comma separated list of screenshot versions.

Example:

```
[[ScreenshotsList($name - $description ($widthx$height))]]
[[graphviz.dot/png]]
[[graphviz.dot/jpg]]
[[graphviz.dot/gif]]
[[graphviz.dot/svg]]
[[graphviz.dot/svgz]]
[[graphviz.dot]]
[[graphviz.neato/png]]
[[graphviz.neato/jpg]]
[[graphviz.neato/gif]]
[[graphviz.neato/svg]]
[[graphviz.neato/svgz]]
```

```
[[graphviz.neato]]
[[graphviz.twopi/png]]
[[graphviz.twopi/jpg]]
[[graphviz.twopi/gif]]
[[graphviz.twopi/svg]]
[[graphviz.twopi/svgz]]
[[graphviz.twopi]]
[[graphviz.circo/png]]
[[graphviz.circo/jpg]]
[[graphviz.circo/gif]]
[[graphviz.circo/svg]]
[[graphviz.circo/svgz]]
[[graphviz.circo]]
[[graphviz.fdp/png]]
[[graphviz.fdp/jpg]]
[[graphviz.fdp/gif]]
[[graphviz.fdp/svg]]
[[graphviz.fdp/svgz]]
[[graphviz.fdp]]
[[graphviz/png]]
[[graphviz/jpg]]
[[graphviz/gif]]
[[graphviz/svg]]
[[graphviz/svgz]]
[[graphviz]]
```

Graphviz (<http://trac-hacks.org/wiki/GraphvizPlugin>) provides a plugin for Trac to render graphviz (<http://www.graphviz.org/>) drawings within a Trac wiki page.

```
[[TracNav]]
```

TracNav: The Navigation Bar for Trac

This macro implements a fully customizable navigation bar for the Trac wiki engine. The contents of the navigation bar is a wiki page itself and can be edited like any other wiki page through the web interface. The navigation bar supports hierarchical ordering of topics. The design of TracNav mimics the design of the TracGuideToc that was originally supplied with Trac. The drawback of TracGuideToc is that it is not customizable without editing its source code and that it does not support hierarchical ordering.

Installation

See <http://projects.edgewall.com/trac/wiki/TracPlugins>.

Usage

To use TracNav, create an index page for your site and call the [TracNav?](#) macro on each page, where the navigation bar should be displayed. The index page is a regular wiki page. The page with the table of contents must include an unordered list of links that should be displayed in the navigation bar.

To display the navigation bar on a page, you must call the TracNav macro on that page and pass the name of your table of contents as argument.

Additional information and a life example

Please visit: <http://svn.ipd.uka.de/trac/javaparty/wiki/TracNav>.

Author and License

Copyright 2005, 2006

◇ Bernhard Haumacher (hau at haumacher.de)

◇ Thomas Moschny (moschny at ipd.uni-karlsruhe.de)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

[[JPNav]]

TracNav: The Navigation Bar for Trac

This macro implements a fully customizable navigation bar for the Trac wiki engine. The contents of the navigation bar is a wiki page itself and can be edited like any other wiki page through the web interface. The navigation bar supports hierarchical ordering of topics. The design of TracNav mimics the design of the TracGuideToc that was originally supplied with Trac. The drawback of TracGuideToc is that it is not customizable without editing its source code and that it does not support hierarchical ordering.

Installation

See <http://projects.edgewall.com/trac/wiki/TracPlugins>.

Usage

To use TracNav, create an index page for your site and call the [TracNav?](#) macro on each page, where the navigation bar should be displayed. The index page is a regular wiki page. The page with the table of contents must include an unordered list of links that should be displayed in the navigation bar.

To display the navigation bar on a page, you must call the TracNav macro on that page and pass the name of your table of contents as argument.

Additional information and a life example

Please visit: <http://svn.ipd.uka.de/trac/javaparty/wiki/TracNav>.

Author and License

Copyright 2005, 2006

◇ Bernhard Haumacher (hau1 at haumacher.de)

◇ Thomas Moschny (moschny at ipd.uni-karlsruhe.de)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

[[TOC]]

Generate a table of contents for the current page or a set of pages. If no arguments are given, a table of contents is generated for the current page, with the top-level title stripped:

```
[[TOC]]
```

To generate a table of contents for a set of pages, simply pass them as comma separated arguments to the TOC macro, e.g. as in

```
[[TOC(TracGuide, TracInstall, TracUpgrade, TracIni, TracAdmin, TracBackup, TracLogging, TracPermissions, TracWiki, WikiFormatting, TracBrowser, TracRoadmap, TracChangeset, TracTickets, TracReports, TracQuery, TracTimeline, TracRss, TracNotification)]]
```

The following *control* arguments change the default behaviour of the TOC macro:

Argument	Meaning
heading=<x>	Override the default heading of "Table of Contents"
noheading	Suppress display of the heading.
depth=<n>	Display headings of <i>subsequent</i> pages to a maximum depth of <n>.
inline	Display TOC inline rather than as a side-bar.
titleindex	Only display the page name and title of each page, similar to TitleIndex .

Note that the current page must also be specified if individual wiki pages are given in the argument list.

[[WikiInclude]]

Inserts the contents of another wiki page. It expands arguments in the page in the form {{N}}.

Example:

```
[[WikiInclude(Foo)]]
[[WikiInclude(Foo|Argument1|Argument2..|ArgumentN)]]
```

[[BlogShow]]

Displays a blog based on tags

The list of tags to be shown can be specified as arguments to the macro. If no tags are specified as parameters, then the default 'blog' tag is used.

Because any wiki page can be a blog entry, it is suggested that one uses a unique tag for any page that should appear in the blog, such as the tag: 'blog' (the default).

The following options can be specified:

union - Specify whether the join for the tags listed should be a union or intersection(default).

num_posts - Number of posts to display.

year - Year for which to show posts.

month - Month for which to show posts.

day - Day of the month for which to show posts.

delta - How many days of posts should be shown.

mark_update - Specify whether to show "Updated on" for posts that have been updated.

format - Show as RSS feed ('rss') or HTML (else).

post_size - Number of bytes to show before truncating the post and providing a (...) link. Posts are

truncated at the next line break after the byte count is reached. **poster** - Show only entries made by a single named user.

If specifying dates with `year`, `month`, and/or `day`, the current value is specified if missing. For example, if `day` is specified but `year` and `month` are not, then `year` will be filled in with the current year and `month` will be filled with the current month. If only `year` and `month` are specified, then that indicates the whole month is desired.

The `num_posts` options is bounded by the date options if combined. For example, if `num_posts=5` and `month=4` is specified, it will show up to 5 posts from the month of April. If only 3 posts exist, then only 3 are shown. If a date option is not specified, then it will show the last `num_posts` posts.

The `poster` option is bounded by all of the above options and also limits the calendar to display tallies respective of the named user.

Examples

```
[[BlogShow()]]
[[BlogShow(blog,pacopablo)]]
[[BlogShow(blog,pacopablo,union=True)]]
[[BlogShow(blog,pacopablo,num_posts=5)]]
[[BlogShow(blog,pacopablo,month=4,num_posts=5)]]
[[BlogShow(blog,pacopablo,year=2006,month=4)]]
[[BlogShow(blog,pacopablo,year=2006,month=4,day=12)]]
[[BlogShow(blog,pacopablo,delta=5)]]
[[BlogShow(blog,pacopablo,delta=5,mark_updated=False)]]
[[BlogShow(blog,poster=UserName)]]
[[SplitPost]]
```

Displays a blog based on tags

The list of tags to be shown can be specified as arguments to the macro. If no tags are specified as parameters, then the default 'blog' tag is used.

Because any wiki page can be a blog entry, it is suggested that one uses a unique tag for any page that

should appear in the blog, such as the tag: 'blog' (the default).

The following options can be specified:

union - Specify whether the join for the tags listed should be a union or intersection(default).

num_posts - Number of posts to display.

year - Year for which to show posts.

month - Month for which to show posts.

day - Day of the month for which to show posts.

delta - How many days of posts should be shown.

mark_update - Specify whether to show "Updated on" for posts that have been updated.

format - Show as RSS feed ('rss') or HTML (else).

post_size - Number of bytes to show before truncating the post and providing a (...) link. Posts are truncated at the next line break after the byte count is reached. **poster** - Show only entries made by a single named user.

If specifying dates with `year`, `month`, and/or `day`, the current value is specified if missing. For example, if `day` is specified but `year` and `month` are not, then `year` will be filled in with the current year and `month` will be filled with the current month. If only `year` and `month` are specified, then that indicates the whole month is desired.

The `num_posts` options is bounded by the date options if combined. For example, if `num_posts=5` and `month=4` is specified, it will show up to 5 posts from the month of April. If only 3 posts exist, then only 3 are shown. If a date option is not specified, then it will show the last `num_posts` posts.

The `poster` option is bounded by all of the above options and also limits the calendar to display tallies respective of the named user.

Examples

```
[[BlogShow()]]
[[BlogShow(blog,pacopablo)]]
[[BlogShow(blog,pacopablo,union=True)]]
[[BlogShow(blog,pacopablo,num_posts=5)]]
[[BlogShow(blog,pacopablo,month=4,num_posts=5)]]
[[BlogShow(blog,pacopablo,year=2006,month=4)]]
[[BlogShow(blog,pacopablo,year=2006,month=4,day=12)]]
[[BlogShow(blog,pacopablo,delta=5)]]
[[BlogShow(blog,pacopablo,delta=5,mark_updated=False)]]
[[BlogShow(blog,poster=UserName)]]
```

[[BlogPost]]

Inserts a link to create a new blog post

Accepts keyword arguments that specify default parameters. The macro will be hidden unless the user has `BLOG_POSTER` permissions.

tag - Tag that populates the "Tag under" field. This key may be specified as a tuple or list to pass multiple values.

blogtitle - Default blog entry title.

text - Default entry body text.

pagename - Default wiki page name.

readonly - Default readonly page status.

link - Text to display as the link.

Examples

```
[[BlogPost()]]
[[BlogPost(tag=(blog,pacopablo) )]]
[[BlogPost(tag=blog,blogtitle="A Simple Title",text="Body Text")]]
[[BlogPost(tag=blog,pagename=blog/newpage,readonly=1)]]
[[BlogPost(tag=(blog,pacopablo),link="A New Blog Post")]]
[[Poll]]
```

Add a poll. Each argument must be separated by a semi-colon (;) or new-line (if used as a processor). The first argument is the title of the poll, which is also the identifier for each poll.

Usage: `[[TicketPoll(<title>; <arg> [; <arg>] ...)]]`

Where <arg> conforms to the following:

<arg>	Description
<code>query:<ticket-query></code>	Add tickets from a query to the poll.
<code>#<n></code>	Add an individual ticket to the poll.
<code><text></code>	A poll question.

eg.

```
[[ViewTopic]]
[[TracAdminHelp]]
Displays help for trac-admin commands.
```

Examples:

```
[[TracAdminHelp]] # all commands
[[TracAdminHelp(wiki)]] # all wiki commands
[[TracAdminHelp(wiki export)]] # the "wiki export" command
[[TracAdminHelp(upgrade)]] # the upgrade command
```

Macros from around the world

The [Trac Hacks](#) site provides a wide collection of macros and other Trac [plugins](#) contributed by the Trac community. If you're looking for new macros, or have written one that you'd like to share with the world, please don't hesitate to visit that site.

Developing Custom Macros

Macros, like Trac itself, are written in the [Python programming language](#). They are very simple modules, identified by the filename and should contain a single `execute()` function. Trac will display the returned data inserted into the HTML representation of the Wiki page where the macro is called.

It's easiest to learn from an example:

Examples

```
# MyMacro.py -- The world's simplest macro
```

```
def execute(hdf, args, env):  
    return "Hello World called with args: %s" % args
```

You can also use the environment (`env`) object, for example to access configuration data and the database, for example:

```
def execute(hdf, txt, env):  
    return env.config.get('trac', 'repository_dir')
```

Note that since version 0.9, wiki macros can also be written as [TracPlugins](#). This gives them some capabilities that ?classic? macros do not have, such as being able to directly access the HTTP request.

For more information about developing macros, see the [development resources](#) on the main project site.

See also: [WikiProcessors](#), [WikiFormatting](#), [TracGuide](#)