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:

> **Mon Sep 6 13:52:00 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 `<img>` 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.

`[[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

To install TracNav?, place the file `TracNav.py` in the `wiki−macros` subdirectory and the accompanying `tracnav.css` file in the `templates` subdirectory of your Trac project. Add this line

```
@import url(<?cs var:chrome.href ?>/site/tracnav.css);
```

to the `templates/site_css.cs` file of your Trac project.

The `tracnav.css` file defines the styles for displaying the navigation bar. These styles build upon the styles for TracGuideToc that come with your Trac distribution. If you just install the macro but miss to install the style file, TracNav? will work but look somewhat strange.

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 an pass the name of your table of contents as argument.

Author and license

Copyright 2005

- ◊ Bernhard Haumacher (haui 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
```

Additional information and a life example

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

[[ShellExample]]

Preprocessor Highlight ShellExample?.

[[HtmlCode]]

Example macro.

[[sortedtable]]

[[Anchor]]

[[RSSget]]

A Simple RSS News Retriever: Returns the content of an rss2 feed, parsed and pretty.

Usage:

Error: Macro RSSget(http://www.example.com/feed.xml) failed

```
−2Name or service not known
```

This edition:

- ◊ caches fetched feeds to /tmp/tracrss
- ◊ returns the feed as a <dl>

Customisation:

◊ RESULTS_FULL = 1; < number of results for which descriptions are displayed
◊ RESULTS_TOTAL = 5; < total number of results, titles of posts are displayed
◊ CACHE_INTERVAL = 1500 < time till feed is rechecked.

Known Issues: Doesn't parse atom or rss or rdf. This is a one trick pony

[[WikiCalendar]]

[[Folding]]

small macro to fold areas and toggle the visibility on click. first parameter is one of the following actions

"printscript" prints the necessary javascript "begin" marks the beginning of the area to be folded "end" marks the end of the area to be folded "activator" prints the area that will activate the folding

second parameter is the id of the area to be folded/unfolded third parameter can be

"visibility" of the area when the action is "begin" (possible values "inline","none")
"formatting" of the content when the action is "activator" (possible values "wiki","none")

forth paramerter is the content for the activator action.

example usage:

Error: Macro Folding(activator,idoftheregiontobeunfolded,wiki,== click me to view ==) failed

`href`

this area is folded by default

<h2>click me to hide</h2> this area is visible by default

Author: Thorsten Ott (wanagi at web−helfer.de)

[[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

[[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]]

GraphvizMacro? (http://trac−hacks.org/wiki/GraphvizPlugin) provides a plugin for Trac to render graphviz (http://www.graphviz.org/) drawings within a Trac wiki page.

[[S5]]

Allow the current Wiki page to be viewed as an S5 slidewhos.

[[ShowSmileys]]

Renders in a table the list of available smileys. Optional argument is the number of columns in the table (defaults 3).

[[ShowEntities]]

Renders in a table the list of HTML entities. Optional argument is the number of columns in the table (defaults 3).

[[ShowSymbols]]

Renders in a table the list of known symbols. Optional argument is the number of columns in the table (defaults 3).

[[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:

```
# 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 â€œclassicâ€ 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