

Table of Contents

Wiki Macros

Using Macros

Examples

Available Macros

Macros from around the world

Developing Custom Macros

3

3

3

3

3

4

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: [\[Timestamp\]](#)

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

Display: [\[HelloWorld\(Testing\)\]](#)

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](#).

[\[MacroList\]](#)

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:

```
#!/python
# 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:

```
#!/python
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

From:

<https://wiki.linux-ottawa.org/> - **Linux-Ottawa (OCLUG) Wiki**

Permanent link:

<https://wiki.linux-ottawa.org/doku.php?id=wikimacros>

Last update: **2015/06/09 19:23**

