Table of Contents

Wiki Processors 3
Using Processors 3
Available Processors 4
Code Highlighting Support 4
Advanced Topics: Developing Processor Macros 5

Wiki Processors

Processors are WikiMacros designed to provide alternative markup formats for the Wiki engine. Processors can be thought of as *macro functions to process user-edited text*.

The Wiki engine uses processors to allow using Restructured Text, raw HTML and textile in any Wiki text throughout Trac.

Using Processors

To use a processor on a block of text, use a Wiki code block, selecting a processor by name using *shebang notation* (#!), familiar to most UNIX users from scripts.

Example 1 (*inserting raw HTML in a wiki text*):

```
#!html
<pre class="wiki"><code>
#!html
&lt;h1 style="color: orange"&gt;This is raw HTML&lt;/h1&gt;
```

</pre> </code>

Results in:

```
#!html
<h1 style="color: orange">This is raw HTML</h1>
```

Example 2 (*inserting Restructured Text in wiki text*):

```
#!html
<pre class="wiki"><code>
#!rst
A header
--------
This is some **text** with a footnote [[:*]]_.

.. [[:*]] This is the footnote.
```

</pre> </code>

Results in:

```
#!rst
A header
--------
This is some **text** with a footnote [[:*]]_.

.. [[:*]] This is the footnote.
```

Example 3 (*inserting a block of C source code in wiki text*):

```
#!html
<pre class="wiki"><code>
#!c
int main(int argc, char *argv[])
{
  printf("Hello World\n");
  return 0;
}
```

</pre> </code>

Results in:

```
#!c
int main(int argc, char *argv[])
{
  printf("Hello World\n");
  return 0;
}
```

Available Processors

The following processors are included in the Trac distribution:

- **html** – Insert custom HTML in a wiki page. See WikiHtml.
- **rst** – Trac support for Restructured Text. See WikiRestructuredText.
- **textile** – Supported if Textile is installed. See a Textile reference.

Textile link above is rotten. this one works, allows to test example.

Code Highlighting Support

Trac includes processors to provide inline syntax highlighting for the following languages:

- **c** – C
- **cpp** – C++
- **python** – Python
- **perl** – Perl
- **ruby** – Ruby
- **php** – PHP
- **asp** — ASP
- **sql** – SQL
- **xml** – XML

Note: *Trac relies on external software packages for syntax coloring. See TracSyntaxColoring for more info.*

By using the MIME type as processor, it is possible to syntax-highlight the same languages that are supported when browsing source code. For example, you can write:

```
<code>
#!text/html
<h1>text</h1>
```

</code>

The result will be syntax highlighted HTML code. The same is valid for all other mime types supported.

For more processor macros developed and/or contributed by users, visit:

- ProcessorBazaar
- MacroBazaar

Advanced Topics: Developing Processor Macros

Developing processors is no different from WikiMacros. In fact they work the same way, only the usage syntax differs. See WikiMacros for more information.

Example: (*Restructured Text Processor*):

```
#!python
from docutils.core import publish_string

def execute(hdf, text, env):
    html = publish_string(text, writer_name = 'html')
    return html[[:html.find('<body>')+6:html.find('<:body>')]].strip()
```

See also: WikiMacros, WikiHtml, WikiRestructuredText, TracSyntaxColoring, WikiFormatting, TracGuide

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

Permanent link:
https://wiki.linux-ottawa.org/doku.php?id=wikiprocessors

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