

# Wiki Processors

Processors are [WikiMacros](#) designed to provide alternative markup formats for the Trac 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](#) and [raw HTML](#) in any wiki text throughout Trac.

## Using Processors

To use a processor on a block of text, use a wiki blockquote, 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
<h1 style="color: orange">This is raw HTML</h1>
}}}
```

**Results in:**

## This is raw HTML

---

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

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

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

**Results in:**

```
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*):

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

**Results in:**

```
int main(int argc, char *argv[])
```

This is raw HTML

```
{
    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.

## 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:

```
{ {{
#!text/html
<h1>text</h1>
}}
```

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 than [WikiMacros](#). In fact they work the same way, only the usage syntax differs. See [WikiMacros](#) for more information.

**Example:** (*Restructured Text Processor*):

```
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](#)