

## 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*.

Wiki processors can be used in any Wiki text throughout Trac, for various different purposes, like:

- [syntax highlighting](#) or for rendering text verbatim,
- rendering [Wiki markup inside a context](#), like inside `<div>` blocks or `<span>` or within `<td>` or `<th>` table cells,
- using an alternative markup syntax, like [raw HTML](#) and [Restructured Text](#), or [■textile](#)

## Using Processors

To use a processor on a block of text, first delimit the lines using a Wiki *code block*:

```
{{{
The lines
that should be processed...
}}}
```

Immediately after the `{{{` or on the line just below, add `#!` followed by the *processor name*.

```
{{{
#!processorname
The lines
that should be processed...
}}}
```

This is the "shebang" notation, familiar to most UNIX users.

Besides their content, some Wiki processors can also accept *parameters*, which are then given as `key=value` pairs after the processor name, on the same line. If `value` has to contain space, as it's often the case for the `style` parameter, a quoted string can be used (`key="value with space"`).

As some processors are meant to process Wiki markup, it's quite possible to *nest* processor blocks. You may want to indent the content of nested blocks for increased clarity, this extra indentation will be ignored when processing the content.

## Examples

Wiki Markup	Display
<u>Example 1:</u> Inserting raw HTML	
<pre>{{{ #!html &lt;h1 style="color: grey"&gt;This is raw HTML&lt;/h1&gt; }}}</pre>	This is raw HTML
<u>Example 2:</u> Highlighted Python code in a <code>&lt;div&gt;</code> block with custom style	
<pre>{{{#!div style="background: #ffd; border: 3px ridge"  This is an example of embedded "code" block:  {{{ #!python def hello():     return "world" }}}  }}}</pre>	<div>This is an example of embedded "code" block:</div> <pre>def hello():     return "world"</pre>

**Example 3:** Searching tickets from a wiki page, by keywords.

```

{{{
#!html
<form action="/query" method="get"><div>
<input type="text" name="keywords" value="~" size="30"/>
<input type="submit" value="Search by Keywords"/>
<!-- To control what fields show up use hidden fields
<input type="hidden" name="col" value="id"/>
<input type="hidden" name="col" value="summary"/>
<input type="hidden" name="col" value="status"/>
<input type="hidden" name="col" value="milestone"/>
<input type="hidden" name="col" value="version"/>
<input type="hidden" name="col" value="owner"/>
<input type="hidden" name="col" value="priority"/>
<input type="hidden" name="col" value="component"/>
-->
</div></form>
}}}
```

**Available Processors**

The following processors are included in the Trac distribution:

<b>#!default</b>	Present the text verbatim in a preformatted text block. This is the same as specifying <i>no</i> processor name (and no <b>#!</b> )
<b>#!comment</b>	Do not process the text in this section (i.e. contents exist only in the plain text - not in the rendered page).
<b>HTML related</b>	
<b>#!html</b>	Insert custom HTML in a wiki page.
<b>#!htmlcomment</b>	Insert an HTML comment in a wiki page ( <i>since 0.12</i> ).
	Note that <b>#!html</b> blocks have to be <i>self-contained</i> , i.e. you can't start an HTML element in one block and close it later in a second block. Use the following processors for achieving a similar effect.
<b>#!div</b>	Wrap an arbitrary Wiki content inside a <code>&lt;div&gt;</code> element ( <i>since 0.11</i> ).
<b>#!span</b>	Wrap an arbitrary Wiki content inside a <code>&lt;span&gt;</code> element ( <i>since 0.11</i> ).
<b>#!td</b>	Wrap an arbitrary Wiki content inside a <code>&lt;td&gt;</code> element ( <i>since 0.12</i> )
<b>#!th</b>	Wrap an arbitrary Wiki content inside a <code>&lt;th&gt;</code> element ( <i>since 0.12</i> )
<b>#!tr</b>	Can optionally be used for wrapping <b>#!td</b> and <b>#!th</b> blocks, either for specifying row attributes of better visual grouping ( <i>since 0.12</i> )
	See <a href="#">WikiHtml</a> for example usage and more details about these processors.
<b>Other Markups</b>	
<b>#!rst</b>	Trac support for Restructured Text. See <a href="#">WikiRestructuredText</a> .
<b>#!textile</b>	Supported if <a href="#">Textile</a> is installed. See <a href="#">a Textile reference</a> .
<b>Code Highlighting Support</b>	

<pre>#!c #!cpp (C++) #!python #!perl #!ruby #!php #!asp #!java #!js (Javascript) #!sql #!xml (XML or HTML) #!sh (Bourne/Bash shell) etc.</pre>	<p>Trac includes processors to provide inline syntax highlighting for source code in various languages.</p> <p>Trac relies on external software packages for syntax coloring, like <a href="#">Pygments</a>.</p> <p>See <a href="#">TracSyntaxColoring</a> for information about which languages are supported and how to enable support for more languages.</p>
--	--

Using the MIME type as processor, it is possible to syntax-highlight the same languages that are supported when browsing source code.

MIME Type Processors		
Some examples:	The result will be syntax highlighted HTML code:	
<pre>{{{ #!text/html &lt;h1&gt;text&lt;/h1&gt; }}}</pre>	<pre>&lt;h1&gt;text&lt;/h1&gt;</pre>	
	The same is valid for all other <a href="#">mime types supported</a> .	
	<b>#!diff</b> has a particularly nice renderer:	
	<a href="#">Version</a>	
<pre>{{{ #!diff --- Version 55 +++ Version 56 @@ -115,8 +115,9 @@      name='TracHelloWorld', version='1.0',      packages=find_packages(exclude=['*.tests*']), -    entry_points = "" -    [trac.plugins] -    helloworld = myplugins.helloworld -    "", +    entry_points = { +        'trac.plugins': [ +            'helloworld = myplugins.helloworld', +        ], +    }, +    ) }}}</pre>	<div>115</div> <div>116</div> <div>117</div> <div>118</div> <div>119</div> <div>120</div> <div>121</div>	<div>name='TracHelloWorld', version='1.0',</div> <div>packages=find_packages(exclude=["*.tests*"])</div> <div>entry_points = ""</div> <div>[trac.plugins]</div> <div>helloworld = myplugins.helloworld</div> <div>""</div> <div>entry_points = {</div> <div>'trac.plugins': [</div> <div>'helloworld = myplugins.helloworld',</div> <div>],</div> <div>}</div> <div>)</div> <div>117</div> <div>118</div> <div>119</div> <div>120</div> <div>121</div> <div>122</div>

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

- [ProcessorBazaar](#)
- [MacroBazaar](#)
- [Trac Hacks](#) community site

Developing processors is no different from Wiki macros. In fact they work the same way, only the usage syntax differs. See [WikiMacros#DevelopingCustomMacros](#) for more information.

---

See also: [WikiMacros](#), [WikiHtml](#), [WikiRestructuredText](#), [TracSyntaxColoring](#), [WikiFormatting](#), [TracGuide](#)