

# Wikiprint Book

**Title:** Wiki Processors

**Subject:** Java Path Finder - WikiProcessors

**Version:** 3

**Date:** 03/07/2013 11:08:37 PM

## Table of Contents

|  |          |
|--|----------|
| <b>Wiki Processors</b>                       | <b>3</b> |
| Using Processors                             | 3        |
| <b>This is raw HTML</b>                      | <b>3</b> |
| 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
<h1 style="color: orange">This is raw HTML</h1>
}}}
```

**Results in:**

**This is raw HTML**

Note that since 0.11, such blocks of HTML have to be self-contained, i.e. you can't start an HTML element in one block and close it later in a second block. Use div or span processors for achieving similar effect (see [WikiHtml](#)).

**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[])
{
    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](#).
- **div** -- Wrap an arbitrary Wiki content in a <div> element (*since 0.11*). See [WikiHtml](#).
- **span** -- Wrap an arbitrary Wiki content in a <span> element (*since 0.11*). See also [WikiHtml](#).
- **rst** -- Trac support for Restructured Text. See [WikiRestructuredText](#).
- **textile** -- Supported if [Textile](#) is installed. See [a Textile reference](#).
- **comment** -- Do not process the text in this section (i.e. contents exist only in the plain text - not in the rendered page).
- **diff** -- Pretty print patches and diffs.

## Code Highlighting Support

Trac includes processors to provide inline [syntax highlighting](#) for the following languages:

- **c** -- C
- **cpp** -- C++
- **csharp** -- C# (use `#!text/x-csharp`)
- **python** -- Python
- **perl** -- Perl
- **ruby** -- Ruby
- **php** -- PHP
- **asp** -- ASP
- **java** -- Java
- **js** -- Javascript
- **sql** -- SQL
- **xml** -- XML
- **sh** -- Bourne/Bash shell

**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:

```
<h1>text</h1>
```

The same is valid for all other mime types supported.

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

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

## **Advanced Topics: Developing Processor Macros**

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

---

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