WikiFormatting

Error: Macro TracGuideToc(None) failed

```
'NoneType' object has no attribute 'find'
```

Wiki markup is a core feature in Trac, tightly integrating all the other parts of Trac into a flexible and powerful whole.

Trac has a built in small and powerful wiki rendering engine. This wiki engine implements an ever growing subset of the commands from other popular Wikis, especially ?MoinMoin and ?WikiCreole.

This page will give you an in-depth explanation of the wiki markup available anywhere WikiFormatting is allowed.

The *Cheat sheet* below gives you a quick overview for the most common syntax, each link in the *Category* column will lead you to the more detailed explanation later in this page.

A few other wiki pages present the advanced features of the Trac wiki markup in more depth:

- TracLinks covers all the possible ways to refer precisely to any Trac resource or parts thereof,
- WikiPageNames talks about the various names a wiki page can take, CamelCase or not
- WikiMacros lists the macros available for generating dynamic content,
- WikiProcessors and WikiHtml details how parts of the wiki text can be processed in special ways

Cheat sheet

Category	Wiki Markup	Display

Font Styles	`'''bold''', ''italic'', '''''Wikipedia style'''''`
	`` `monospaced and ''nowiki''` ``
	`**bold**, //italic//, **//!WikiCreole style//**`
		bold, *italic*, ***Wikipedia style***
		`monospaced and nowiki`
		bold, *italic*, ***WikiCreole style***

Headings	`== Level 2 ==` `=== Level 3 ^([#hn note])^`	**Level 2** **Level 3** [note]

Paragraphs	`First paragraph`
	`on multiple lines.`
	
	`Second paragraph.`
		First paragraph on multiple lines.
		
		Second paragraph.

Lists	`* bullets list`
	`on multiple paragraphs`
	`1. nested list`
	`a. different numbering styles`
		• bullets list on multiple paragraphs
		1. nested list
		1. different numbering styles

Definition Lists	`term:: definition on multiple paragraphs`	term definition on multiple paragraphs

Preformatted Text	`{{{`
	`multiple lines, ''no wiki''`
	`  white space respected`
	`}}}`
		`multiple lines, ''no wiki''`
		`  white space respected`

Blockquotes	`  if there's some leading` `  space the text is quoted`	if there's some leading space the text is quoted
Discussion Citations	`>> ... (I said)` `> (he replied)`	... (I said) (he replied)
Tables	`||= Table Header =|| Cell ||` `||||  (details below)  ||`	**Table Header** Cell (details below)

Links	`http://trac.edgewall.org`
	`WikiFormatting (CamelCase)`
	`wiki:WikiFormatting, wiki:"WikiFormatting"`
		?http://trac.edgewall.org
		WikiFormatting (CamelCase)
		wiki:WikiFormatting, wiki:"WikiFormatting"

TracLinks	`#1 (ticket), [1] (changeset), {1} (report)`
	`ticket:1, ticket:1#comment:1`
	`Ticket [ticket:1], [ticket:1 ticket one]`
	`Ticket [[ticket:1]], [[ticket:1|ticket one]]`
		#1 (ticket), [1] (changeset), {1} (report)
		ticket:1, ticket:1#comment:1
		Ticket 1, ticket one
		Ticket 1, ticket one

Setting Anchors	`[=#point1 (1)] First...` `see [#point1 (1)]`	(1) First... see (1)
Escaping Markup	`!'' doubled quotes` `!wiki:WikiFormatting, !WikiFormatting`	'' doubled quotes wiki:WikiFormatting, WikiFormatting
Images	`[[Image(*link*)]]`	trac POWERED
Macros	`[[MacroList(*)]]` `[[Image?]]`	*(short list of all available macros)* *(help for the Image macro)*

Processors	`{{{`
	`#!div style="font-size: 80%"`
	`Code highlighting:`
	`  {{{#!python`
	`  hello = lambda: "world"`
	`  }}}`
	`}}}`
		Code highlighting:
		`hello = lambda: "world"`

Comments	`{{{#!comment`
	`Note to Editors: ...`
	`}}}`	

Miscellaneous	`Line [[br]] break`
	`Line \\ break`
	`----`
		Line
		break Line
		break
		---- 	

Font Styles

The Trac wiki supports the following font styles:

Wiki Markup	Display
`* '''bold''', ''' triple quotes !''' can be bold too if prefixed by ! ''',`	**bold**, **triple quotes ''' can be bold too if prefixed by !** ,
`* ''italic''`	*italic*
`* '''''bold italic''''' or ''italic and ''' italic bold ''' ''`	***bold italic*** or *italic and* ***italic bold***
`* __underline__`	underline
`* {{{monospace}}} or `monospace` (hence `{{{` or {{{`}}} quoting)`	`monospace` or `monospace` (hence `{{{` or `` ` `` quoting)
`* ~~strike-through~~`	~~strike-through~~
`* ^superscript^`	superscript
`* ,,subscript,,`	subscript
`* **also bold**, //italic as well//, and **'' bold italic **'' //(since 0.12)//`	**also bold**, *italic as well*, and ***bold italic*** *(since 0.12)*

Notes:

- `{{{...}}}` and `` `...` `` commands not only select a monospace font, but also treat their content as verbatim text, meaning that no further wiki processing is done on this text.
- `!` tells wiki parser to not take the following characters as wiki format, so pay attention to put a space after !, e.g. when ending bold.
- all the font styles marks have to be used in opening/closing pairs, and they must nest properly (in particular, an `''` italic can't be paired with a `//` one, and `'''` can't be paired with `**`)

Headings

You can create heading by starting a line with one up to six *equal* characters ("=") followed by a single space and the headline text.

The headline text can be followed by the same number of "=" characters, but this is no longer mandatory.

Finally, the heading might optionally be followed by an explicit id. If not, an implicit but nevertheless readable id will be generated.

Wiki Markup	Display
`= Heading =`	# Subheading
`== Subheading`	## About *this*
`=== About ''this'' ===`	### Explicit id
`=== Explicit id === #using-explicit-id-in-heading`	## Subheading
`== Subheading #sub2`	

Paragraphs

A new text paragraph is created whenever two blocks of text are separated by one or more empty lines.

A forced line break can also be inserted, using:

Wiki Markup	Display
`Line 1[[BR]]Line 2`	Line 1 Line 2

`Paragraph one`
 	
`Paragraph two`
	Paragraph one
	
	Paragraph two

Lists

The wiki supports both ordered/numbered and unordered lists.

Example:

Wiki Markup	Display

`* Item 1`
`  * Item 1.1`
`  * Item 1.1.1`
`  * Item 1.1.2`
`  * Item 1.1.3`
`  * Item 1.2`
`* Item 2`
	• Item 1
	  ♦ Item 1.1
	  ◊ Item 1.1.1
	  ◊ Item 1.1.2
	  ◊ Item 1.1.3
	  ♦ Item 1.2
	• Item 2

`- items can start at the beginning of a line and they can span multiple lines` `  - be careful though to continue the line with the appropriate indentation, otherwise that will start a new paragraph...`	• items can start at the beginning of a line and they can span multiple lines ♦ be careful though to continue the line with the appropriate indentation, otherwise that will start a new paragraph...

`1. Item 1`
`  a. Item 1.a`
`  a. Item 1.b`
`  i. Item 1.b.i`
`  i. Item 1.b.ii`
`1. Item 2`
	1. Item 1
	  1. Item 1.a
	  2. Item 1.b
	  1. Item 1.b.i
	  2. Item 1.b.ii
	2. Item 2

`And numbered lists can also be restarted with an explicit number:` `3. Item 3`	And numbered lists can also be restarted with an explicit number: 3. Item 3

Definition Lists

The wiki also supports definition lists.

Wiki Markup	Display
`llama::` `  some kind of mammal, with hair`	

```
 ppython::
   some kind of reptile, without hair
   (can you spot the typo?)
```

llama

some kind of mammal, with hair

ppython

some kind of reptile, without hair (can you spot the typo?)

Note that you need a space in front of the defined term.

Preformatted Text

Block containing preformatted text are suitable for source code snippets, notes and examples. Use three *curly braces* wrapped around the text to define a block quote. The curly braces need to be on a separate line.

Wiki Markup	Display

`{{{`
`def HelloWorld():`
`  print '''Hello World'''`
`}}}`
	`def HelloWorld():`
	`  print '''Hello World'''`

Note that this kind of block is also used for selecting lines that should be processed through WikiProcessors.

Blockquotes

In order to mark a paragraph as blockquote, indent that paragraph with two spaces.

Wiki Markup	Display
`Paragraph` `  This text is a quote from someone else.`	Paragraph This text is a quote from someone else.

Discussion Citations

To delineate a citation in an ongoing discussion thread, such as the ticket comment area, e-mail-like citation marks (">", ">>", etc.) may be used.

Wiki Markup	Display

`>> Someone's original text`
`> Someone else's reply text`
`>  - which can be any kind of Wiki markup`
`My reply text`
	Someone's original text
	Someone else's reply text
	♦ which can be any kind of Wiki markup
	My reply text

Tables

Simple Tables

Simple tables can be created like this:

Wiki Markup	Display		
`||Cell 1||Cell 2||Cell 3||`	Cell 1	Cell 2	Cell 3
`||Cell 4||Cell 5||Cell 6||`	Cell 4	Cell 5	Cell 6

Cell headings can be specified by wrapping the content in a pair of '=' characters. Note that the '=' characters have to stick to the cell separators, like this:

Wiki Markup

```
||         ||= stable =||= latest =||
||= 0.10 =||  0.10.5  || 0.10.6dev||
||= 0.11 =||  0.11.6  || 0.11.7dev||
```

Display

	stable	latest
0.10	0.10.5	0.10.6dev
0.11	0.11.6	0.11.7dev

Finally, specifying an empty cell means that the next non empty cell will span the empty cells. For example:

Wiki Markup

```
|| 1 || 2 || 3 ||
|||| 1-2 || 3 ||
|| 1 |||| 2-3 ||
|||||| 1-2-3 ||
```

Display

1	2	3
1-2		3
1	2-3	
1-2-3		

Note that if the content of a cell "sticks" to one side of the cell and only one, then the text will be aligned on that side. Example:

Wiki Markup

```
||=Text =||= Numbers =||
||left align    ||        1.0||
||  center      ||        4.5||
||      right align||     4.5||
|| default alignment ||   2.5||
||default||         2.5||
||  default ||      2.5||
|| default ||       2.5||
```

Display

Text	Numbers
left align	1.0
center	4.5
right align	4.5
default alignment	2.5
default	2.5
default	2.5
default	2.5

If contrary to the example above, the cells in your table contain more text, it might be convenient to spread a table row over multiple lines of markup. The \ character placed at the end of a line after a cell separator tells Trac to not start a new row for the cells on the next line.

Wiki Markup

```
|| this is column 1 [http://trac.edgewall.org/newticket new ticket] || \
|| this is column 2 [http://trac.edgewall.org/roadmap the road ahead] || \
|| that's column 3 and last one ||
```

Display

this is column 1 ?new ticket this is column 2 ?the road ahead that's column 3 and last one

Complex Tables

If the possibilities offered by the simple "pipe"-based markup for tables described above are not enough for your needs, you can create more elaborated tables by using WikiProcessor based tables.

Links

Hyperlinks are automatically created for WikiPageNames and URLs. WikiPageLinks can be disabled by prepending an exclamation mark "!" character, such as `!WikiPageLink`.

Wiki Markup	Display
`TitleIndex, http://www.edgewall.com/, !NotAlink`	TitleIndex, ?http://www.edgewall.com/, NotAlink

Links can be given a more descriptive title by writing the link followed by a space and a title and all this inside square brackets. If the descriptive title is omitted, then the explicit prefix is discarded, unless the link is an external link. This can be useful for wiki pages not adhering to the WikiPageNames convention.

Wiki Markup	Display

`* [http://www.edgewall.com Edgewall Software]`
`* [wiki:TitleIndex Title Index]`
`* [wiki:TitleIndex]`
`* [wiki:ISO9000]`
	• ?Edgewall Software
	• Title Index
	• TitleIndex
	• ISO9000?

Following the ?WikiCreole trend, the descriptive title can also be specified by writing the link followed by a pipe ('|') and a title and all this inside *double* square brackets.

`* [[http://www.edgewall.com|Edgewall Software]]`
`* [[wiki:TitleIndex|Title Index]]`
`  or even [[TitleIndex|Title Index]]`
`* [[wiki:TitleIndex]]`
`  ''' but not ![[TitleIndex]]! '''`
`* [[ISO9000]]`
	• ?Edgewall Software
	• Title Index or even Title Index
	• TitleIndex **but not [[TitleIndex]]!**
	• ISO9000?

Note: the ?WikiCreole style for links is quick to type and certainly looks familiar as it's the one used on Wikipedia and in many other wikis. Unfortunately it conflicts with the syntax for macros. So in the rare case when you need to refer to a page which is named after a macro (typical examples being TitleIndex, InterTrac and InterWiki), by writing `[[TitleIndex]]` you will actually call the macro instead of linking to the page.

Trac Links

Wiki pages can link directly to other parts of the Trac system. Pages can refer to tickets, reports, changesets, milestones, source files and other Wiki pages using the following notations:

Wiki Markup	Display

`* Tickets: #1 or ticket:1`
`* Reports: {1} or report:1`
`* Changesets: r1, [1] or changeset:1`
`* ...`
`* targeting other Trac instances,`
`  so called InterTrac links:`
`  - Tickets: #Trac1 or Trac:ticket:1`
`  - Changesets: [Trac1] or Trac:changeset:1`
	• Tickets: #1 or ticket:1
	• Reports: {1} or report:1
	• Changesets: r1, [1] or changeset:1
	• ...
	• targeting other Trac instances, so called InterTrac links:

- Tickets: #Trac1 or Trac:ticket:1
- Changesets: [Trac1] or Trac:changeset:1

There are many more flavors of Trac links, see TracLinks for more in-depth information and a reference for all the default link resolvers.

Setting Anchors

An anchor, or more correctly speaking, an ?anchor name can be added explicitly at any place in the Wiki page, in order to uniquely identify a position in the document:

```
[=#point1]
```

This syntax was chosen to match the format for explicitly naming the header id documented above. For example:

```
== Long title == #title
```

It's also very close to the syntax for the corresponding link to that anchor:

```
[#point1]
```

Optionally, a label can be given to the anchor:

```
[[=#point1 '''Point 1''']]
```

Wiki Markup	Display

`[#point2 jump to the second point]`
`...`
`Point2:  [=#point2] Jump here`
	jump to the second point
	...
	Point2: Jump here

For more complex anchors (e.g. when a custom title is wanted), one can use the Span macro, e.g. `[[span(id=point2, class=wikianchor, title=Point 2, ^(2)^)]]`.

Escaping Links and WikiPageNames

You may avoid making hyperlinks out of TracLinks by preceding an expression with a single "!" (exclamation mark).

Wiki Markup	Display
`!NoHyperLink` `!#42 is not a link`	NoHyperLink #42 is not a link

Images

Urls ending with `.png`, `.gif` or `.jpg` are no longer automatically interpreted as image links, and converted to `<img>` tags.

You now have to use the [[Image]] macro. The simplest way to include an image is to upload it as attachment to the current page, and put the filename in a macro call like `[[Image(picture.gif)]]`.

In addition to the current page, it is possible to refer to other resources:

- `[[Image(wiki:WikiFormatting:picture.gif)]]` (referring to attachment on another page)
- `[[Image(ticket:1:picture.gif)]]` (file attached to a ticket)
- `[[Image(htdocs:picture.gif)]]` (referring to a file inside project htdocs)
- `[[Image(source:/trunk/trac/htdocs/trac_logo_mini.png)]]` (a file in repository)

Wiki Markup	Display
`[[Image(htdocs:../common/trac_logo_mini.png)]]`	trac POWERED

See WikiMacros for further documentation on the `[[Image()]]` macro.

Macros

Macros are *custom functions* to insert dynamic content in a page.

Wiki Markup	Display
`[[RecentChanges(Trac,3)]]`	**Feb 23, 2013** • TracRepositoryAdmin (diff) • TracWorkflow (diff) • TracInterfaceCustomization (diff)

See WikiMacros for more information, and a list of installed macros.

The detailed help for a specific macro can also be obtained more directly by appending a "?" to the macro name.

Wiki Markup	Display
`[[MacroList?]]`	**`[[MacroList]]`** Display a list of all installed Wiki macros, including documentation if available. Optionally, the name of a specific macro can be provided as an argument. In that case, only the documentation for that macro will be rendered. Note that this macro will not be able to display the documentation of macros if the `PythonOptimize` option is enabled for mod_python!

Processors

Trac supports alternative markup formats using WikiProcessors. For example, processors are used to write pages in reStructuredText or HTML.

Wiki Markup	Display
Example 1: HTML	

```
{{{
#!html
<h1 style="text-align: right; color: blue">
 HTML Test
</h1>
```

```
}}}
```

HTML Test

Example 2: Code Highlighting

```
{{{
#!python
class Test:

    def __init__(self):
        print "Hello World"
if __name__ == '__main__':
   Test()
}}}
```

```
class Test:
    def __init__(self):
        print "Hello World"
if __name__ == '__main__':
   Test()
```

Example 3: Complex Tables

```
{{{#!th rowspan=4 align=justify
With the `#td` and `#th` processors,
table cells can contain any content:
}}}
|----------------
{{{#!td
  - lists
  - embedded tables
  - simple multiline content
}}}
|----------------
{{{#!td
As processors can be easily nested,
so can be tables:
  {{{#!th
  Example:
  }}}
  {{{#!td style="background: #eef"
  || must be at the third level now... ||
  }}}
}}}
|----------------
{{{#!td
Even when you don't have complex markup,
this form of table cells can be convenient
to write content on multiple lines.
}}}
```

With the `#td` and `#th` processors, table cells can contain any content:

- lists
- embedded tables
- simple multiline content

As processors can be easily nested, so can be tables:

Example: must be at the third level now...

Even when you don't have complex markup, this form of table cells can be convenient to write content on multiple lines.

See WikiProcessors for more information.

Comments

Comments can be added to the plain text. These will not be rendered and will not display in any other format than plain text.

Wiki Markup	Display

`Nothing to`
`{{{`
`#!comment`
`Your comment for editors here`
`}}}`
`see ;-)`
	Nothing to
	
	see 🙂

Miscellaneous

An horizontal line can be used to separated different parts of your page:

Wiki Markup	Display

`Four or more dashes will be replaced by an horizontal line (<HR>)`
`----`
`See?`
	Four or more dashes will be replaced by an horizontal line (<HR>)
	--- 	
	See?

`"macro" style [[br]] line break`	"macro" style line break
`!WikiCreole style \\ line\\break`	WikiCreole style line break