

Redmine - Feature #3061

Let macros optionally match over multiple lines and ignore single curly braces

2009-03-27 12:36 - Nils Israel

Status:	Closed	Start date:	2009-03-27
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Text formatting	Estimated time:	0.00 hour
Target version:	2.1.0		
Resolution:	Fixed		

Description

Hi,
please modify the regex to match the wiki macros in [source:trunk/lib/redmine/wiki_formatting/textile/formatter.rb](https://source.trunk/lib/redmine/wiki_formatting/textile/formatter.rb). With this version the regex matches options with multiple lines and closing curly braces in it. I use this in the wiki_latex_plugin.

Thx,
Nils

```
MACROS_RE = /
    (!)?           # escaping
    (
    \{\{
    ([\w]+)        # opening tag
    ([\w]+)        # macro name
    (\{.*?\})?    # optional arguments
    \}\}          # closing tag
    )
    /xm unless const_defined?(:MACROS_RE)
```

Associated revisions

Revision 10209 - 2012-08-17 16:46 - Jean-Philippe Lang

Macros processing overhaul (#3061, #11633).

- macro arguments are no longer parsed by text formatters
- macro output is escaped unless it's html safe

Revision 10210 - 2012-08-18 10:25 - Jean-Philippe Lang

Let macros optionally accept a block of text (#3061).

History

#1 - 2010-08-04 23:04 - Marc Mengel

Note that this is now in:
lib/redmine/wiki_formatting.rb and not down in the textile specific code...

#2 - 2011-04-26 11:52 - Michael Kussmaul

It seems in current trunk it is still not included? Is there anything wrong with the proposed regex?

#3 - 2011-04-26 12:31 - Etienne Massip

- Target version set to Candidate for next minor release

Looks ok, but why do the actual RE has such a limitation preventing a closing curly brace to be present in the argument list ?

[source:trunk/lib/redmine/wiki_formatting.rb#L74](https://source.trunk/lib/redmine/wiki_formatting.rb#L74)

#4 - 2011-04-26 13:41 - Michael Kussmaul

I'm not sure, if this is the main "issue" - but there are quite some macro-plugins out there, which need support for multiline macros (e.g. <http://www.redmine.org/boards/3/topics/10649> or also the graphviz-plugin could benefit from it).

Currently they only work by patching multiple redmine files (if I understood the docu of those plugins correctly). They usually work all the same: an

external tool generates an image. The "recipe" how to bake this image is inside a multi-line macro. I like this approach, as it allows to extend redmine in a very flexible way.

#5 - 2011-04-28 20:34 - Etienne Massip

- Category set to *Text formatting*

#6 - 2012-04-01 23:33 - Thimios Dimopoulos

There is no `MACROS_RE =` line on Redmine 1.3. Has anyone managed to make the plugin work with redmine 1.3?

thanks!

#7 - 2012-04-10 13:04 - Thimios Dimopoulos

An alternative to using this plugin, is using MathJax: https://github.com/process91/redmine_latex_mathjax

Just installed it on Redmine 1.3 stable and it works fine.

#8 - 2012-05-02 07:47 - Oscar Dekkers

+1

#9 - 2012-06-13 18:01 - Simon Delamare

In Redmine 1.3, `MACROS_RE` is defined in `redmine/app/helpers/application_helper.rb`

#10 - 2012-08-08 23:37 - Etienne Massip

- Status changed from *New* to *Resolved*

- Target version changed from *Candidate for next minor release* to *2.1.0*

- Resolution set to *Fixed*

Looks like it has been fixed with [r10178](#), could you please confirm?

#11 - 2012-08-09 16:19 - Jean-Philippe Lang

Etienne Massip wrote:

Looks like it has been fixed with [r10178](#), could you please confirm?

No, the regexp does not match multiple lines (no `m` modifier). The problem with passing a block of text to a macro is that it would be processed by all previous parsers (textile, redmine links...) and has a lot of chances to be altered. To make this feature work nicely, we should prevent that.

#12 - 2012-08-09 19:45 - Etienne Massip

- Target version changed from *2.1.0* to *Candidate for next major release*

Indeed, I missed the modifier (you committed the "allow curly braces as argument" part, still).

Wouldn't it make sense to parse macro tags first?

#13 - 2012-08-09 19:56 - Etienne Massip

- Status changed from *Resolved* to *New*

- Resolution deleted (*Fixed*)

#14 - 2012-08-13 16:07 - Jean-Philippe Lang

- Target version changed from *Candidate for next major release* to *2.2.0*

Etienne Massip wrote:

Wouldn't it make sense to parse macro tags first?

Sure, but that's not that simple. Formatted text is cached and macro output must not be parsed by textile and redmine links. The solution would be to "catch" macro at the beginning then substitute with their output at the very end of the process.

#15 - 2012-08-13 16:08 - Jean-Philippe Lang

- Subject changed from *Modify MACRO_RE to match over multiple lines and ignore single curly braces* to *Let macros optionally match over multiple lines and ignore single curly braces*

#16 - 2012-08-18 10:41 - Jean-Philippe Lang

- Status changed from *New* to *Closed*

- Assignee set to *Jean-Philippe Lang*

- Target version changed from *2.2.0* to *2.1.0*

- Resolution set to *Fixed*

Implemented in [r10209](#), [r10210](#). Macros now optionally accept a block of text as a third argument. Examples from the doc:

```
Redmine::WikiFormatting::Macros.register do
  # Declaring a macro that do not accept a block of text
  macro :my_macro do |obj, args|
    "My macro output"
  end

  # Declaring a macro that accepts a block of text (as third argument)
  macro :my_macro2 do |obj, args, text|
    "My macro output"
  end
end
```

```
# Macros are invoked in formatted text using the following
# syntax:
#
# No arguments:
# {{my_macro}}
#
# With arguments:
# {{my_macro(arg1, arg2)}}
#
# With a block of text:
# {{my_macro2
# multiple lines
# of text
# }}
#
# With arguments and a block of text
# {{my_macro2(arg1, arg2)
# multiple lines
# of text
# }}
#
# Note that the closing tag }} must be at the start of a new line
```

With [r10209](#), the output of macros is automatically escaped. Use `#html_safe` if the generated output contains html tags that should not be escaped. Examples:

```
Redmine::WikiFormatting::Macros.register do
  macro :my_macro do |obj, args|
    "Unsafe <tag>"
  end

  macro :my_macro2 do |obj, args|
    "Safe <tag>".html_safe
  end
end

# Then:
#
# {{my_macro}}
# produces: Unsafe &lt;tag&gt;
#
# {{my_macro2}}
# produces: Safe <tag>
```