Macros with empty text parameter are rendered differently in preview mode vs. non preview mode.

The preview mode seems to parse the the closing double curly brackets as text parameter and does not stop until the next macros double curly brackets, while the non preview mode parses an empty text parameter.

The sample code below shows demonstrates this with the collapse macro, but it happens for any macro which allows a text parameter and the two attached images.

This might be an issue with a greedy regular expression.

Test code for the bug in Redmine 4.x

```ruby
{{collapse(Collapse 1)}}
```

Above is a empty collapse macro which renders correctly in preview and in wiki view.

```ruby
{{collapse(Collapse 2)}}
```

Above is a empty collapse macro which renders correctly in wiki view but this text (and the following macro up to its closing curly brackets) will not be visible in preview mode.

```ruby
{{collapse(Collapse 3)}}
This is a block of text that is collapsed by default. It can be expanded by clicking a link.
```

Above is a non empty collapse macro which renders correctly in preview and in wiki view.

Test code for the bug in Redmine 4.x

[Collapse 1](Collapse 1)

[Collapse 2](Collapse 2)

[Collapse 3](Collapse 3)
Environment:
Redmine version                4.1.1.stable
Ruby version                   2.6.6-p146 (2020-03-31) [x86_64-linux]
Rails version                  5.2.4.2
Environment adapter production
Database adapter               PostgreSQL
Mailer queue                   ActiveJob::QueueAdapters::AsyncAdapter
Mailer delivery                smtp
SCM:
Subversion                     1.10.4
Git                            2.20.1
Filesystem                     Xitolite 2.20.1
Redmine plugins:
- additionals                    2.0.24-master
- redmine_advancedimage          0.0.4
- redmine_agile                   1.5.3
- redmine_checklists              3.1.16
- redmine_filelinks               0.0.0
- redmine_git_hosting             4.0.2
- redmine_latex_mathjax           0.4.0
- redmine_lightbox2               0.5.1
- redmine_local_avatars           1.0.0
- redmine_mentions                1.0.0
- redmine_postgresql_search       1.0.6
- redmine_recurring_tasks         0.3.4
- redmine_reporting               2.0.6
- redmine_wiki_lists              0.0.9
- redmine_wikiapproval            0.0.0
- redmine_wikiforms               0.0.0
- redmineup_tags                  2.0.8
- sidebar_hide                    0.0.8-fix

History
#1 - 2020-05-16 05:28 - Go MAEDA
- Category changed from Wiki to Text formatting
- Status changed from New to Confirmed

#2 - 2020-05-21 05:00 - Yuichi HARADA
- File 33432.patch added

It seems that the line break code is different between preview mode and non-preview mode. So preview mode doesn't seem to match the regular expression ApplicationHelper::MACROS_RE.

preview mode(Linebreak: "\n"). Since the input value is posted via JavaScript, the line break code is replaced from "\n" to "\n".

"Test code for the bug in Redmine 4.x
Above is a empty collapse macro which renders correctly in preview and in wiki view.\nThis is a block of text that is collapsed by default.\nIt can be expanded by clicking a link.\nThis is a block of text that is collapsed by default.\nIt can be expanded by clicking a link.
"
non-preview mode(Linebreak: "\r\n")

"Test code for the bug in Redmine 4.x
Above is a empty collapse macro which renders correctly in preview and in wiki view.\nThis is a block of text that is collapsed by default.\nIt can be expanded by clicking a link.\nThis is a block of text that is collapsed by default.\nIt can be expanded by clicking a link.
"

Fixed the regular expression. I think this will solve the problem.
diff --git a/app/helpers/application_helper.rb b/app/helpers/application_helper.rb
index bf55e05a5..5bcdca29f 100644
--- a/app/helpers/application_helper.rb
+++ b/app/helpers/application_helper.rb
@@ -1277,8 +1277,8 @@
 module ApplicationHelper
 (\{\{\# opening tag
 ![\w]+\# macro name
-\((\[[^\n\r]*?]\)?\# optional arguments
-\(\[[\n\r].*?\[\n\r]\)?\# optional block of text
+\(\(([^\R]*?)\)?\# optional arguments
+\(\R.*?\R?\)?\# optional block of text
 \}\}\# closing tag
 )\))/mx unless const_defined?(:MACROS_RE)

#3 - 2020-05-24 09:33 - Go MAEDA

Yuichi HARADA wrote:

It seems that the line break code is different between preview mode and non-preview mode. So preview mode doesn't seem to match the regular expression ApplicationHelper::MACROS_RE.

Thank you for posting the patch. But I think the patch needs some fix.

I think the following line does not work as expected because the metacharacter \R cannot be put into []. [^R] is actually equivalent to [^R].

\(\(([^R]*?)\)?\) # optional arguments

See the test result below.

```
irb(main):001:0> /^\n\r$/ =~ "\n"
=> nil
irb(main):002:0> /^\R$/ =~ "\n"
=> 0
irb(main):003:0> /[^\R]/ =~ "\n"
=> nil
```

#4 - 2020-05-26 02:39 - Yuichi HARADA

Go MAEDA wrote:

I think the following line does not work as expected because the metacharacter \R cannot be put into []. [^R] is actually equivalent to [^R].

Sequence does not work.

I confirmed that the negative match /[^[R]/ doesn't work with the line break code. Fixed the patch.

#5 - 2020-06-02 15:09 - Go MAEDA

Target version set to Candidate for next minor release

#6 - 2020-06-06 08:32 - Go MAEDA

File 33432-test.patch added

Attaching test code for the patch.

#7 - 2020-06-09 07:58 - Go MAEDA

Target version changed from Candidate for next minor release to 4.0.8

Affected version changed from 4.1.1 to 4.0.0
Setting the target version to 4.0.8.

#8 - 2020-06-14 06:40 - Go MAEDA
- Target version changed from 4.0.8 to Candidate for next minor release

The actual cause of the problem is that the regular expression expects that there are two line terminators between "macroname[" and "]". The reason why there appears to be no problem when a line terminator is "\v\n" is that "\v" is parsed as the first line terminator and "\n" is parsed as the second line terminator.

So, even after applying the patch 33432-v2.patch, the macro in the following text is not detected in the Preview tab.

collapse{}
}

The code below can test it.

Index: test/helpers/application_helper_test.rb
===================================================================
--- test/helpers/application_helper_test.rb (Revision 19814)
+++ test/helpers/application_helper_test.rb (working copy)
@@ -1937,6 +1937,13 @@
  end
 end
+  def test_catch_macros_should_allow_empty_text_block
+    macros = catch_macros("{{collapse\r\n}}")
+    assert_equal "{{collapse\r\n}}", macros[0]
+    macros = catch_macros("{{collapse\n}}")
+    assert_equal "{{collapse\n}}", macros[0]
+  end
+
+private

  def wiki_links_with_special_characters

Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redmine 4.x preview render.PNG</td>
<td>26.4 KB</td>
<td>2020-05-11</td>
<td>Holger Mößinger</td>
</tr>
<tr>
<td>Redmine 4.x wiki render.PNG</td>
<td>27.6 KB</td>
<td>2020-05-11</td>
<td>Holger Mößinger</td>
</tr>
<tr>
<td>33432.patch</td>
<td>785 Bytes</td>
<td>2020-05-21</td>
<td>Yuichi HARADA</td>
</tr>
<tr>
<td>33432-v2.patch</td>
<td>700 Bytes</td>
<td>2020-05-26</td>
<td>Yuichi HARADA</td>
</tr>
<tr>
<td>33432-test.patch</td>
<td>735 Bytes</td>
<td>2020-06-06</td>
<td>Go MAEDA</td>
</tr>
</tbody>
</table>