

Redmine - Defect #29259

Attachment preview does not work for some source files such as JavaScript and Go

2018-07-22 20:16 - Stephan Wenzel

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Attachments	Estimated time:	0.00 hour
Target version:	4.0.4	Affected version:	
Resolution:	Fixed		
Description			
<p>Redmine squanders the capabilities of CodeRay for previewing code files in Redmine's preview pane. A very prominent squandering is the negligence of javascript files, for which Redmine only shows that no preview is available.</p> <p>AttachmentsController#show method only tests: @attachment.is_text? - The function Attachment::is_text? only relies on Redmine::MimeType.is_type?('text', filename). The mime type of javascript, however, is "application/javascript". Here Redmine misses what CodeRay can do.</p> <p>I propose the following patches:</p> <p>Add function 'is_code?' to Attachment.rb</p> <pre>def is_code? ::CodeRay::FileType[filename].present? end</pre> <p>Patch AttachmentsController.rb (Redmine 3.4.6)</p> <pre>- elsif @attachment.is_text? && @attachment.filesize <= Setting.file_max_size_displayed.to_i.kilobyte + elsif (@attachment.is_text? @attachments.is_code?) && @attachment.filesize <= Setting.file_max_size_displayed.to_i.kilobyte</pre> <p>These two patches add previews for cfc, cfm, clj, cpp, cu, cxx, c++, C, dpr, gemspec, go, groovy, gvy, haml, hpp, h++, html.erb, js, lua, mab, pas, phtml, prawn, py3, pyw, raydebug, rjs, rpdf, ru, rxml, sass, taskpaper, template, tmproj, xaml</p> <p>I have provided a plugin that applies these patches as a proof of concept:</p> <p>[[https://www.redmine.org/plugins/redmine_more_code]]</p>			
Related issues:			
Related to Redmine - Feature #24681: Syntax highlighter: replace CodeRay with...		Closed	
Blocked by Redmine - Defect #31285: Syntax highlighting does not work for att...		Closed	

Associated revisions

Revision 18122 - 2019-05-04 11:32 - Go MAEDA

Attachment preview does not work for some source files such as JavaScript and Go (#29259).

Patch by Go MAEDA with the help of Stephan Wenzel's contribution.

Revision 18123 - 2019-05-04 12:26 - Go MAEDA

Merged r18122 from trunk to 4.0-stable (#29259).

History

#1 - 2018-07-22 20:24 - Stephan Wenzel

Difference between is_text? based on Redmine::MimeType and proposed is_code? based on CodeRay::FileType

is_text?	is_code?
c	c

cc	cc
	cfc
	cfm
	clj
cpp	cpp
cs	
csb	
css	css
csv	
	cu
	cxx
	c++
	C
diff	diff
	dpr
erb	erb
	gemspec
	go
	groovy
	gvx
h	h
	haml
hh	hh
	hpp
	h++
htm	htm
html	html
	html.erb
ini	
install	
java	java
	js
	json
jsp	
	lua
	mab
mxml	
	pas
patch	patch
	phtml
php	php
php3	php3
php4	php4
php5	php5
pl	
pm	
po	
properties	

	prawn
py	py
	py3
	pyw
rake	rake
	raydebug
rb	rb
rbw	rbw
readme	
rhtml	rhtml
	rjs
	rpdf
	ru
ruby	
	rxml
	sass
sh	
sql	sql
	taskpaper
	template
	tmproj
tpl	
txt	
upgrade	
	xaml
xhtml	xhtml
xml	xml
xsd	
yaml	yaml
yml	yml

#2 - 2018-07-23 02:44 - Go MAEDA

Thank you for reporting this issue.

I think the following code can detect more kinds of text files. Redmine already uses mimemagic gem ([source:tags/3.4.6/lib/redmine/thumbnail.rb#L33](https://source.redmine.org/3.4.6/lib/redmine/thumbnail.rb#L33)).

```
Index: app/models/attachment.rb
=====
--- app/models/attachment.rb      (revision 17449)
+++ app/models/attachment.rb      (working copy)
@@ -235,7 +235,8 @@
   end

   def is_text?
-    Redmine::MimeType.is_type?('text', filename)
+    Redmine::MimeType.is_type?('text', filename) ||
+    File.open(diskfile) {|f| MimeMagic.by_magic(f).try(:text?)} rescue false
   end

   def is_image?
```

#3 - 2018-07-23 02:57 - Go MAEDA

The is_text? method should be called last for performance reason.

```
Index: app/models/attachment.rb
=====
```

```

--- app/models/attachment.rb      (revision 17452)
+++ app/models/attachment.rb      (working copy)
@@ -235,7 +235,8 @@
     end

     def is_text?
-       Redmine::MimeType.is_type?('text', filename)
+       Redmine::MimeType.is_type?('text', filename) ||
+       File.open(diskfile) {|f| MimeMagic.by_magic(f).try(:text?)} rescue false
     end

     def is_image?
@@ -259,7 +260,7 @@
     end

     def previewable?
-       is_text? || is_image? || is_video? || is_audio?
+       is_image? || is_video? || is_audio? || is_text?
     end

     # Returns true if the file is readable

```

#4 - 2018-07-23 05:49 - Go MAEDA

- Category set to Attachments
- Target version set to Candidate for next major release

#5 - 2018-07-28 09:29 - Go MAEDA

- Target version changed from Candidate for next major release to 4.1.0

#6 - 2018-08-05 09:30 - Go MAEDA

- File allow-preview-for-any-text-files.diff added

Updated the patch.

- Changed not to use MimeMagic gem
- Added a test.

#7 - 2018-08-05 09:31 - Go MAEDA

- File deleted (allow-preview-for-any-text-files.diff)

#8 - 2018-08-05 09:31 - Go MAEDA

- File allow-preview-for-any-text-files.diff added

#9 - 2018-08-10 10:06 - Go MAEDA

- File allow-preview-for-any-text-files-v2.diff added

Updated the patch. Simplified Attachment#is_text?.

#10 - 2018-08-10 12:49 - Marius BĂLTEANU

Go MAEDA wrote:

Updated the patch. Simplified Attachment#is_text?.

Go Maeda, the patch contains some undesired changes.

#11 - 2018-08-10 23:21 - Pavel Rosický

```
! Redmine::Utils.binary?(File.read(diskfile, 4096)) rescue false
```

see

<http://blog.honeybadger.io/benchmarking-exceptions-in-ruby-yep-theyre-slow/>
<https://robots.thoughtbot.com/don-t-inline-rescue-in-ruby>

IMO text-type detection should be evaluated by mime only, reading the content this way is just guessing.

#12 - 2018-08-11 00:24 - Stephan Wenzel

Why not stick with a function „is_code?“ as I already proposed?

A negative test of a file on it's binary property may be a too powerful „catch all“. A „catch all“ based on it's non-binary character may raise questions how f.i. an „.html“-file should be treated in the preview pane - it (the html-file) has a non binary character (thus, is „text“) but a non-textual representation may be appropriate. The same is true for non-binary .pdf or .svg files. Just because a file's content is not binary, does not necessarily mean that a text-representation is most appropriate.

The example „is_code?“ function based on CodeRay, which does the preview coloring of code, would leave other plugins room for further previewers for files in text form, for which a non-textual representation is appropriate.

I propose to always leave the mime test to the respective previewer, else there will always be a mismatch.

Eventually, it might be useful to implement a hook in the format block of attachments_controller#show method, that allows an easy integration for highly specialised previewers.

#13 - 2018-09-20 01:31 - Go MAEDA

- Related to Feature #24681: Syntax highlighter: replace CodeRay with Rouge added

#14 - 2018-09-30 06:01 - Go MAEDA

- Target version changed from 4.1.0 to Candidate for next major release

Since we have moved from CodeRay to Rouge, we cannot use the proposed is_code method.

#15 - 2019-05-02 08:39 - Go MAEDA

- File 0002-Attachment-preview-does-not-support-source-code-in-s.patch added

I completely rewrote the patch to support the new syntax highlighter Rouge introduced in Redmine 4.0. The patch has to be applied after the patch attached to [#31285](#).

After applying the patch, files supported by the syntax highlighter are treated as text.

#16 - 2019-05-02 08:39 - Go MAEDA

- Blocked by Defect #31285: Syntax highlighting does not work for attachments with .pl extension added

#17 - 2019-05-03 09:56 - Go MAEDA

- Tracker changed from Patch to Defect

- Subject changed from Attachments Controller squanders CodeRay's capabilities to Attachment preview does not support source code in some languages

- Target version changed from Candidate for next major release to 4.0.4

Setting the target version to 4.0.4.

#18 - 2019-05-04 11:32 - Go MAEDA

- Subject changed from Attachment preview does not support source code in some languages to Attachment preview does not work for some source files such as JavaScript and Go

- Status changed from New to Resolved

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the fix.

#19 - 2019-05-04 12:29 - Go MAEDA

- Status changed from Resolved to Closed

Files

allow-preview-for-any-text-files.diff	2.54 KB	2018-08-05	Go MAEDA
allow-preview-for-any-text-files-v2.diff	5.38 KB	2018-08-10	Go MAEDA
0002-Attachment-preview-does-not-support-source-code-in-s.patch	4.02 KB	2019-05-02	Go MAEDA