Redmine - Feature #25538

Drop support for Ruby 2.2.1 and ealier, 2.2.2+ is now required

2017-04-06 21:51 - Toshi MARUYAMA

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Jean-Philippe Lang	**% Done:**	0%
Category:	Ruby support	**Estimated time:**	0.00 hour
Target version:	4.0.0		
Resolution:	Fixed		

Description

Support for Ruby 1.9.3 and 2.0 has ended by Ruby community.
https://www.ruby-lang.org/en/news/2015/02/23/support-for-ruby-1-9-3-has-ended/
https://www.ruby-lang.org/en/news/2016/02/24/support-plan-of-ruby-2-0-0-and-2-1/

Related issues:

Related to Redmine - Feature #25807: Upgrade to roadie-rails 1.2	**Closed**
Related to Redmine - Defect #25794: Mass-deleted attachments are not represen...	**Closed**
Related to Redmine - Defect #26183: Use Nokogiri 1.7.2	**Closed**
Related to Redmine - Feature #29947: Update roadie gem to 3.4.0	**Closed**
Related to Redmine - Patch #29999: Update rdoc gem	**Closed**
Related to Redmine - Defect #30161: Incorrect supported Ruby version in doc/I...	**Closed**
Blocks Redmine - Feature #23630: Migrate to Rails 5.2	**Closed**
Blocks Redmine - Patch #26503: Update nokogiri gem (~> 1.8.0)	**Closed**

Associated revisions

Revision 16778 - 2017-07-08 18:33 - Jean-Philippe Lang

Ruby 1.9.3 no longer supported (#25538).

History

#1 - 2017-04-06 21:54 - Toshi MARUYAMA

- Blocks Defect #24271: htmlentities warning added

#2 - 2017-04-06 21:55 - Toshi MARUYAMA

- Blocks Defect #22335: Images with non-ASCII file names are not shown in PDF added

#3 - 2017-04-06 21:56 - Toshi MARUYAMA

- Description updated

#4 - 2017-04-07 02:56 - Go MAEDA

+1

- Ruby 2.1 and earlier should no longer be used. (https://www.ruby-lang.org/en/news/2017/04/01/support-of-ruby-2-1-has-ended/)
- Supporting ancient versions of Ruby is big burden for plugin developers.

#5 - 2017-04-07 10:31 - Go MAEDA

- File 0001-Dropped-Ruby-2.1-and-earlier-support.patch added

This is a patch to drop ruby 2.1 and earlier support by removing RUBY_VERSION condition expressions.

#6 - 2017-04-07 13:49 - Fernando Hartmann

Hi, this will block my upgrade to 3.4.0 in my Ubuntu 14.04 LTS and OpenSuSE 42.2# and probably other distros.

Shouldn't this be put in 4.0.0 like #23630 and #19755 ?
Given some more time to users to plan system upgrades ?

#7 - 2017-04-07 14:33 - Holger Just

I agree, dropping support for Ruby versions should probably be done in a major version instead.

That said, dropping Ruby 1.9.3 support is probably a good idea in the short-to-middle term since it is seriously old and not supported by the Ruby language team for quite some time. For other versions, I'd be actually more careful.

- Ubuntu 14.04 LTS defaults to Ruby 1.9.3, but also ships with Ruby 2.0. Ubuntu 16.04 LTS ships with Ruby 2.3 by default.
- Debian stable (Jessie) currently ships with Ruby 2.1. The next stable (which is just around the corner) will ship with Ruby 2.3
- RHEL/CentOS 7 ships with Ruby 2.0 currently, RHEL/CentOS 6 still ships Ruby 1.8.7.
- OpenSuSE 42 ships with Ruby 2.1

Given this landscape, in order to provide the broadest possible support for existing users, I'd propose to only drop Ruby 1.9.3 support in the short term (if at all, see below). I understand that is is desirable to also drop at least Ruby 2.0 but this would seriously inconvenience many people who have to use older Ruby versions from their upstream vendors. Thus, I'd propose to only follow this approach now if the gain of not supporting older Ruby 2.x versions is actually significant.

In general, I think that the removal of supported Ruby versions should be done in a major version bump only. With the release of Redmine 4, I'd be quite happy if only Ruby >= 2.2 would be supported. Given that Rails 5 requires Ruby >= 2.2.2 anyway, this is probably a given requirement. Until then, I'd propose we still support the existing Ruby versions in all existing and future 3.x-stable branches.

#8 - 2017-04-08 09:35 - Jean-Philippe Lang

- Target version deleted (3.4.0)

#9 - 2017-04-08 09:51 - Toshi MARUYAMA

- Blocks deleted (Defect #24271: htmlentities warning)

#10 - 2017-04-08 09:51 - Toshi MARUYAMA

- Blocks deleted (Defect #22335: Images with non-ASCII file names are not shown in PDF)

#11 - 2017-04-09 05:31 - Go MAEDA

- Subject changed from Drop support Ruby 1.9.3 and 2.0 to Drop support Ruby 2.2.1 and ealier

- Target version set to 4.0.0

Rails 5 requires Ruby 2.2.2 or newer.
http://edgeguides.rubyonrails.org/upgrading_ruby_on_rails.html#ruby-versions

#12 - 2017-05-12 04:38 - Go MAEDA

- Related to Feature #25807: Upgrade to roadie-rails 1.2 added

#13 - 2017-06-06 08:31 - Go MAEDA

We can unpin test_after_commit gem 0.4 if Ruby 1.9.3 support is dropped (#25794, r16582).

#14 - 2017-06-06 08:31 - Go MAEDA

- Related to Defect #25794: Mass-deleted attachments are not represented correctly in email notifications added

#15 - 2017-06-19 08:25 - Toshi MARUYAMA

- Related to Defect #26183: Use Nokogiri 1.7.2 added

#16 - 2017-06-19 08:26 - Toshi MARUYAMA

https://github.com/sparklemotion/nokogiri/pull/1640#issuecomment-301805320

> The 1.6.8.x release isn't supported anymore, and we're encouraging teams to upgrade to get security patches.

#17 - 2017-06-19 13:21 - Toshi MARUYAMA

- Target version changed from 4.0.0 to 3.4.0

Nokogiri team refused to maintain old release for old Ruby.
https://github.com/sparklemotion/nokogiri/pull/1640#issuecomment-309409944

#18 - 2017-06-19 15:14 - Holger Just

As a concrete motion from my side:

- I think it's a god idea to require at least Ruby 2.2.2 (or whatever Rails depends on at this time) from Redmine 4 on.
- Removing support for 1.9.3 *could* be sensible in Redmine 3.4 if that aligns with the the way Redmine want's to support Rubies in general. Ruby 1.9.3 is not maintained by the upstream Ruby for quite some time now and no halfway modern OS ships with Ruby 1.9.3 anymore.
- I don't think it is a good idea to remove support for Ruby 2.0 in Redmine 3.4 as this would alienate many RHEL / CentOS users. I would be fine with showing a deprecation warning in Redmine 3.4 for anything <= Ruby 2.1 though. Redmine should still work as far as possible even on Ruby 2.0.

In general, I think it is worth it to define how Redmine plans to handle support to external dependencies like Ruby, Rails or nokogiri (e.g. with #26183) in general. If there is a commitment from the Redmine project to ensure that all dependencies are ensured to be secure, the Redmine project probably has to follow the support cycles of these dependencies too. That would mean to regularly issue patch releases with updated dependencies and to either EOL Redmine versions depending on not supported dependencies or provide monkey patches for them. This issue of how we define our policy was also discussed in https://github.com/rails/rails/pull/27814.

Policy wise, I think this would significantly increase the burden on the project for too little gain. I think we could follow our current best-effort way to provide updates while still striving to be as compatible as possible to a wide version range of external dependencies. As for Ruby compatibility, Redmine has historically attempted to follow the requirements of its respective Rails version. This has allowed Redmine to be used on a wide array of configurations and systems.

Even more so: at Planio we often notice that many Redmine installation are not (or only very seldom) updated. Making it harder to update Redmine by introducing stricter environment dependencies (like a Ruby version) could likely result on even poorer update rates which effectively has a negative effect on overall security (Yes, this is a hard problem!)

Still, it is probably a good idea to nudge users in the direction of a preferred Ruby version and updated dependencies if possible. That way, people with specific requirements can still use other versions but the general population is still nudged into the sweet spot. Hard dependency updates *of the environment* should however be limited to major version updates if at all possible. This also helps packagers who maintain e.g. the redmine package in Debian (and by extension Ubuntu).

#19 - 2017-06-20 11:49 - Go MAEDA

+1 for Holger Just.

#20 - 2017-06-20 14:22 - Fernando Hartmann

Go MAEDA wrote:

> +1 for Holger Just.

I agree.

#21 - 2017-06-20 21:40 - Vasili Korol

Debian 8, CentOS 6 and Ubuntu 14.04 have Ruby 1.9.3 installed by default (although ruby 2.* is also available from the repos). These operating systems are installed on a lot of servers. Switching to ruby2 as default is not always possible there.

#22 - 2017-06-21 00:15 - Holger Just

Vasili Korol wrote:

> Debian 8, CentOS 6 and Ubuntu 14.04 have Ruby 1.9.3 installed by default (although ruby 2.* is also available from the repos). These operating systems are installed on a lot of servers. Switching to ruby2 as default is not always possible there.

- Debian 8 (Jessie) uses Ruby 2,1 by default, Debian Stretch (just released) uses 2.3. Only Debian 7 (Wheezy aka. oldoldstable) used Ruby 1.9.3
- While Ubuntu 14.04 (Trusty) installs Ruby 1.9.3 by default, you can install ruby2.0 alongside with it.
- I personally don't care about CentOS/RHEL 6 anymore. It was released 6 years ago and about everything on there is heavily outdated. CentOS 7 is available since three years.

By continuing to support such extremely old (and mostly unmaintained) software, we would enter a world of pain and would continuously force us to not use any of new features available in later versions and work with older (often also unsupported) versions of gems. If users want to continue using outdated base systems, they can always install a custom Ruby with RVM, ruby-install or rbenv/ruby-build when upgrading Redmine.

Again, support for external dependencies (most importantly Ruby, but also other gems) is always a tradeoff between the required effort for the maintenance and compatibility with existing systems. Redmine currently supports a lot of even very old systems which is increasingly becoming painful. Cutting support for the most outdated systems first in a controlled manner helps (in lieu of saying: is required) for the Redmine project to be able to continuously maintain and modernize its codebase. Cutting support for old dependencies with newer releases is a standard practice followed by about any software package.

#23 - 2017-06-25 10:57 - Jean-Philippe Lang

- Target version changed from 3.4.0 to 4.0.0

3.4 is almost ready for release and we have a code base that support older ruby versions. Current trunk will be upgraded to Rails 5.1 soon and

support for ruby < 2.2.1 will be dropped.

#24 - 2017-07-03 09:49 - Go MAEDA

- Category set to Ruby support

#25 - 2017-07-04 03:45 - Go MAEDA

- Related to Feature #26334: Update Redmine gems to latest stable versions added

#26 - 2017-07-04 05:41 - Toshi MARUYAMA

- Related to deleted (Feature #26334: Update Redmine gems to latest stable versions)

#27 - 2017-07-09 06:29 - Go MAEDA

- Blocks Feature #23630: Migrate to Rails 5.2 added

#28 - 2017-07-23 13:38 - Jean-Philippe Lang

- Subject changed from Drop support Ruby 2.2.1 and ealier to Drop support for Ruby 2.2.1 and ealier, 2.2.2+ is now required

- Status changed from New to Closed

- Assignee set to Jean-Philippe Lang

- Resolution set to Fixed

#29 - 2017-07-23 23:53 - Go MAEDA

- Blocks Patch #26503: Update nokogiri gem (~> 1.8.0) added

#30 - 2018-11-12 14:36 - Go MAEDA

- Related to Feature #29947: Update roadie gem to 3.4.0 added

#31 - 2018-11-25 10:58 - Go MAEDA

- Related to Patch #29999: Update rdoc gem added

#32 - 2018-12-09 13:58 - Go MAEDA

- Related to Defect #30161: Incorrect supported Ruby version in doc/INSTALL added

Files

0001-Dropped-Ruby-2.1-and-earlier-support.patch	3.26 KB	2017-04-07	Go MAEDA