Redmine - Defect #35345

Redmine::Plugin::compare_versions fails with stable and devel branches

2021-06-02 14:16 - Massimo Rossello

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Plugin API	**Estimated time:**	0.00 hour
Target version:			
Resolution:	Invalid	**Affected version:**	

Description

A plugin using Redine::Plugin::require_redmine always fails against stable and development branches. In fact, a version like e.g. 4.0.9.stable (from the latest 4.0.9 tag) can't be matched either against 4.0.9 or 4.0.9.0.

Attached a workaround based on trunk, which is not satisfactory as it isn't using collect(&:to_i) in compare_versions. But it works.

History

#1 - 2021-06-22 05:27 - Go MAEDA

- Tracker changed from Patch to Defect

- Status changed from New to Confirmed

- Target version set to Candidate for next minor release

I have confirmed the issue with the following minimum plugin.

plugins/minimum_plugin/init.rb:

```
Redmine::Plugin.register :minimum_plugin do
  name 'Minimum plugin'
  author 'John Smith'
  version '0.0.1'
  requires_redmine :version => '4.2.1.devel'
end
```

Redmine::PluginRequirementError is raised during startup if you give a version string with "devel" or "stable" to @requires_redmine.

```
$ bin/rails c
/path/to/redmine/lib/redmine/plugin.rb:256:in `block in requires_redmine': minimum_plugin plugin requires one the following Redmine versions: 4.2.1.devel but current is 4.2.1.devel (Redmine::PluginRequirementError)
```

This is because trailing keywords such as "devel" and "stable" in the required version value will be converted to zero but keywords in the current version are retained as-is. As the result, compare_versions will compare two values [4, 2, 1, 0] and [4, 2, 1, "stable"] and fail.

To fix the error, the trailing "devel" and "stable" in the current version should be converted to zero as well, as done in the proposed patch.

#2 - 2021-06-24 16:23 - Go MAEDA

- File 35345-v2.patch added

- Target version changed from Candidate for next minor release to 4.1.4

Added tests for the original patch.

Setting the target version to 4.1.4.

#3 - 2021-06-25 09:04 - Go MAEDA

- Target version changed from 4.1.4 to 5.0.0

I think requires_redmine is not supposed to be given a version number with a suffix such as "4.2.1.stable" or "4.2.1.devel", so plugins should not pass such a value to the method.

However, I think it's better to fix this issue in order to avoid an error message that does not make sense like "plugin requires one the following

Redmine versions: 4.2.1.devel but current is 4.2.1.devel".

#4 - 2021-06-25 10:31 - Massimo Rossello

Agreed that a plugin requiring e.g. 4.0.9 (having version 4.0.9.stable) should not bother to know that the version was tagged 'stable'.
However there's currently no way to target version 4.0.9.stable so far. It could be acceptable to solve this on live branches (4.1 and later) only, but IMO 5.0.0 is a bit late unless the versioning policy takes the problem in consideration.

#5 - 2021-06-25 11:45 - Go MAEDA

Massimo Rossello wrote:

> However there's currently no way to target version 4.0.9.stable so far. It could be acceptable to solve this on live branches (4.1 and later) only, but IMO 5.0.0 is a bit late unless the versioning policy takes the problem in consideration.

I don't understand why this fix needs to be applied to 4.1-stable and 4.2-stable.

Even after applying the patch, plugins cannot target only stable versions. The patch just drops "stable" or "devel" suffixes. So, for example, Redmine 4.2.1.devel does not reject a plugin even if the plugin passes the value "4.2.1.stable" to requires_redmine method. And passing a version number with a "devel" or "stable" suffix is a wrong use of the requires_redmine method. Plugins should not do that.

Applying the change to 4.2-stable and 4.1-stable does not provide any other benefit than allowing incorrect usage of the requires_redmine method.

#6 - 2021-06-25 13:08 - Massimo Rossello

Let's assume that some day we'll have a 4.1.x.stable version (just like the current 4.0.9.stable).
How do you suggest to target that version without applying the patch and (I may agree, but it's arbitrary) violating the intended usage of requires_redmine?

Some fix needs to be applied because there's no other option currently available. I anticipated that my patch was not satisfactory; then let's apply some cleaner solution, but having no clean option does not remove the need.
Unless, as I said, versioning policy prevents 4.1 and 4.2 branches to ever tag any version as stable.

#7 - 2021-06-26 02:23 - Go MAEDA

Massimo Rossello wrote:

> Let's assume that some day we'll have a 4.1.x.stable version (just like the current 4.0.9.stable).
> How do you suggest to target that version without applying the patch and (I may agree, but it's arbitrary) violating the intended usage of requires_redmine?

Write as follows and Redmine 4.1.x.stable accepts the plugin. Adding ".stable" is not necessary and should not be done. Plugins don't need to consider ".devel" versions of Redmine. This is because it is for development and not for production environment.

```
  requires_redmine :version => '4.1'
```

#8 - 2021-06-26 14:01 - Massimo Rossello

I am more concerned with "stable" than "devel" versions. Targeting 4.1 would include also 4.1.0, whereas one would like to target 4.1.x.stable ONLY, as the version having all security patches.

For example to target 4.0.9.stable, using

```
requires_redmine :version_or_higher => '4.0.8'
```

would be preferable to

```
requires_redmine :version => '4.0'
```

albeit this still includes an extra unwanted version 4.0.8.

#9 - 2021-07-22 04:00 - Go MAEDA

- Target version deleted (5.0.0)

I think it is harmful to allow the wrong usage of Redmine::Plugin.requires_redmine by committing the patch 35345-v2.patch.

Giving a version number with a suffix such as "4.2.1.stable" and "4.2.1.devel" to Redmine::Plugin.requires_redmine is wrong. Even after applying the patch, suffixes will not have any effect. Allowing such wrong values takes away the chance for users to notice that they are using the method incorrectly.

So, I don't think the patch 35345-v2.patch should be committed.

#10 - 2021-07-22 09:27 - Massimo Rossello

I agree that the patch is not perfect. However, my main point is that IF the 4.2 branch will end up to tag a final version 4.2.x.stable, THEN there will be a problem irrespective of this patch.
So my concern is basically to move the resolution of this problem to 5.0.0, not about using my patch.

However, I also do not understand why using Redmine::Plugin.requires_redmine with versions like "4.2.1.stable" would be wrong. You define such a version scheme, so it should be a legit target.
Finally, that would not introduce a bad habit, since a restricted set of plugin versions would stick to target that stable version only. Plugins targeted to 5.0 would use different values.

#11 - 2021-08-15 14:20 - Massimo Rossello

In order to properly address version 4.2.2.stable with no fix and no patch applied on 4.2 branch, I am using this workaround:

```
  requires_redmine :version => '4.2.2.stable' unless Redmine::VERSION.to_s == '4.2.2.stable'
```

Actually all versions fail the match, including 4.2.2.stable for *requires_redmine* is not able to manage it, but the *unless* statement skips the check when version is exactly 4.2.2.stable. This is probably worse than allowing to address an exact match with the true version 4.2.2.stable, which is considered harmful.

How shall we manage this issue? Shall we make it a bug instead of a patch? At the moment the problem is just being ignored.

#12 - 2021-08-15 14:32 - Go MAEDA

Massimo Rossello wrote:

> How shall we manage this issue? Shall we make it a bug instead of a patch? At the moment the problem is just being ignored.

Just pass a version number without the ".stable" suffix as documented in source:tags/4.2.2/lib/redmine/plugin.rb#L215.

```
requires_redmine :version => '4.2.2'
```

#13 - 2021-08-15 15:30 - Massimo Rossello

Gosh, looks like that one of my early understandings (i.e. can't match using version => 4.0.9) was a false assumption. Probably I encountered a false negative test.

My fault, sorry. Targeting 4.2.2 (and 4.0.9) works perfectly, and now all your reasoning takes a different light.

You can close this issue, since I am not allowed to.
Thanks!

#14 - 2021-08-22 12:22 - Mischa The Evil

- Status changed from Confirmed to Closed

- Resolution set to Invalid

Al Bert, thanks for your feedback.

I'm closing this issue as requested. I've added a note about (and link back to) this issue on Plugin_FAQ.

Files

fix_compare_versions_for_stable_branches.patch	468 Bytes	2021-06-02	Massimo Rossello
35345-v2.patch	1.11 KB	2021-06-24	Go MAEDA