Redmine - Defect #30001

CSV importer ignores shared version names of other projects

2018-11-21 01:05 - Go MAEDA

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Jean-Philippe Lang		**% Done:**	0%
Category:	Importers		**Estimated time:**	0.00 hour
Target version:	3.4.7			
Resolution:	Fixed		**Affected version:**	

Description

When importing issues from CSV, you cannot set a target version for an issue if the specified version name in the CSV is a shared version of another project.

This issue was reported to [redmine.tokyo community](#).

Associated revisions

Revision 17658 - 2018-11-29 17:06 - Jean-Philippe Lang

CSV importer ignores shared version names of other projects (#30001).

Patch by Yuuki NARA and Go MAEDA.

Revision 17659 - 2018-11-29 17:06 - Jean-Philippe Lang

Adds a test for #30001.

Patch by Marius BALTEANU.

Revision 17664 - 2018-11-29 20:12 - Jean-Philippe Lang

Merged r17658 and r17659 to 3.4-stable (#30001).

History

#1 - 2018-11-23 09:01 - Yuuki NARA

+1

#2 - 2018-11-23 11:14 - Yuuki NARA

- File 30001-34stable.patch added

I think that this defect can be addressed by correcting the following points.

app/models/issue_import.rb

def build_object(row, item)
-     if version = issue.project.versions.named(version_name).first
+   if version = issue.project.shared_versions.named(version_name).first

I made patch for 3.4-stable for Revision 17627

#3 - 2018-11-25 18:45 - Marius BĂLTEANU

- File test_for_30001.patch added

- Status changed from New to Confirmed

- Target version set to 3.4.7

Thanks Yuuki Nara for the patch, I can confirm that it works ok.

I'm attaching a unit test for this case, I think we can fix this issue in [3.4.7](#).

#4 - 2018-11-26 01:16 - Go MAEDA

I am in favor of this fix but my only concern is that the case when the current project and another project have versions with the same name. In that case, I think it is the natural behavior to pick the version in the current project. Probably a user may be confused if the version is picked.

Do you know how the patch handles that case? I have not deeply checked the current patch.

#5 - 2018-11-27 18:43 - Yuuki NARA

Thanks Marius BALTEANU for the unit test code.

> Do you know how the patch handles that case? I have not deeply checked the current patch.

It is defined in the following place.

model/project.rb
def shared_versions

I pulled out the relevant part.
It is not designed to output version definition of own PJ first.

```
    @shared_versions ||= begin
      r = root? ? self : root
      Version.
        joins(:project).
        preload(:project).
        where("#{Project.table_name}.id = #{id}" +
              " OR (#{Project.table_name}.status <> #{Project::STATUS_ARCHIVED} AND (" +
              " #{Version.table_name}.sharing = 'system'" +
              " OR (#{Project.table_name}.lft >= #{r.lft} AND #{Project.table_name}.rgt <= #{r.rgt} AND
#{Version.table_name}.sharing = 'tree')" +
              " OR (#{Project.table_name}.lft < #{lft} AND #{Project.table_name}.rgt > #{rgt} AND #{
Version.table_name}.sharing IN ('hierarchy', 'descendants'))" +
              " OR (#{Project.table_name}.lft > #{lft} AND #{Project.table_name}.rgt < #{rgt} AND #{
Version.table_name}.sharing = 'hierarchy')" +
              "))")
    end
```

Actually, in the hierarchical structure project, we defined version by sharing and displayed choices of fixed version.

Did not initially display the version defined in project.

Even with the CSV import function, if there is a version with the same name in the project hierarchy, I think that other than the self PJ will be selected.

Measures

In the process of shared_version, first output its own PJ, then exclude its own PJ when adding hierarchical structure.

Or declare which is output to the behavior when the same version name is defined as undefined.

If you define the same version name, you can not identify which version is defined in any PJ except when editing the ticket.
From this, it is considered inappropriate to define the same version name when sharing in the PJ hierarchy.

#6 - 2018-11-27 18:49 - Yuuki NARA

When using the project hierarchical relationship, it is difficult to intentionally set the same version name between projects.
It only leads to confusion.

When the same project name is defined, it seems mostly a mistake.

#7 - 2018-11-28 09:12 - Marius BĂLTEANU

I don't think that will be so many cases where the users want to import issues and will have versions with the same name. I'm in favour of fixing the current issue and discuss the fix for the second case (versions with the same name) in another ticket.

#8 - 2018-11-28 11:27 - Yuuki NARA

I agree with dealing with the issue of version display order when using the project hierarchical structure as a separate ticket.

The cause of this problem (in the CSV import function) is that the display order of the version choices for each project is not determined when using the version project hierarchy.
(Sometimes versions other than your own project are displayed first)

On the issue editing screen, I think that it is unnatural for users to not display their own projects as a choice of version first, which is very important.

It is difficult to intentionally use the same version name in the project hierarchy.

(There is a high possibility of erroneous setting)

However, I think it is necessary to clarify this problem as a known limit.

#9 - 2018-11-29 01:30 - Go MAEDA

What do you all think about the following implementation? It picks the version in the current project first when there are versions with the same name.

```
diff --git a/app/models/issue_import.rb b/app/models/issue_import.rb
index ad04c0be5..9fc4f5550 100644
--- a/app/models/issue_import.rb
+++ b/app/models/issue_import.rb
@@ -122,7 +122,10 @@ class IssueImport < Import
       end
     end
     if issue.project && version_name = row_value(row, 'fixed_version')
-      if version = issue.project.versions.named(version_name).first
+      version =
+        issue.project.versions.named(version_name).first ||
+        issue.project.shared_versions.named(version_name).first
+      if version
         attributes['fixed_version_id'] = version.id
       elsif create_versions?
         version = issue.project.versions.build
```

#10 - 2018-11-29 09:16 - Marius BĂLTEANU

It is fine for me.

#11 - 2018-11-29 17:07 - Jean-Philippe Lang

- Status changed from Confirmed to Resolved

- Assignee set to Jean-Philippe Lang

- Resolution set to Fixed

Fix and test committed, thanks.

#12 - 2018-11-29 20:13 - Jean-Philippe Lang

- Status changed from Resolved to Closed

Files

30001-34stable.patch	595 Bytes	2018-11-23	Yuuki NARA
test_for_30001.patch	741 Bytes	2018-11-25	Marius BĂLTEANU