

## Redmine - Feature #4585

### Move a Version from one project to another

2010-01-14 17:10 - Martin Lindhe

<b>Status:</b>	New	<b>Start date:</b>	2010-01-14
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Roadmap	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Resolution:</b>			
<b>Description</b> Background: After our company started using redmine we have made some restructuring with regards to projects/subprojects. Initially we created our first "milestone" version attached to something that was a top-level project. After reorganization it is now a child project. We now would like to move the association of the created version field to the top level project. The database is populated with issues so I rather avoid to simply delete the version, create a new and re-associate all issues with the new one.			
<b>Related issues:</b>			
Related to Redmine - Feature #8991: Nested versions for projects with sub-pro...			New 2011-08-05

#### History

##### #1 - 2010-01-14 17:11 - Martin Lindhe

Forgot to mention we are using svn trunk of redmine.

##### #2 - 2010-01-26 17:36 - Martin Lindhe

I solved our issue by editing redmine database, version table and change project\_id to the correct one

##### #3 - 2013-04-03 00:44 - yac yac

+1

##### #4 - 2013-05-24 16:44 - Joel SCHAAL

+1

##### #5 - 2013-05-25 13:14 - Toshi MARUYAMA

- Category set to Roadmap

##### #6 - 2013-12-12 06:29 - Dipan Mehta

+1. I think it is a high time that version should become first class citizen which can be moved, journal'd and also have inheritance/hierarchy as issues and project.

##### #7 - 2014-05-14 11:52 - Asaf H

+1. Related (versions as first class citizens): [#8991](#)

##### #8 - 2014-06-12 15:10 - Toshi MARUYAMA

- Related to Feature #8991: Nested versions for projects with sub-projects added

##### #9 - 2015-10-19 21:21 - Tyler Rose

There is a fairly easy work around without having to go into the database ...

1. Create a new target version in the new project.
2. Browse to the old target version
3. Move the issues to the new project if necessary (batch operation by shift click and right click)
4. Assign new target version
5. Delete old target version

**#10 - 2016-06-06 16:57 - Dipan Mehta**

Tyler Rose wrote:

There is a fairly easy work around without having to go into the database ...

1. Create a new target version in the new project.
2. Browse to the old target version
3. Move the issues to the new project if necessary (batch operation by shift click and right click)
4. Assign new target version
5. Delete old target version

Not at all! First off, there are often a lot of custom fields associated with versions that won't be classified automatically. The versions may be closed, open and so on that will have to be manually checked in re-done. If the versions have related dates which can't be replicated.

Another big thing is the version-id. Once you recreate new version it will have new version-id. So in all places - in text fields, wikis, etc. where you have written version#id that auto expands its title and attaches link - now you have to edit!  
The users may not even know where all it has been mentioned.

The most critical part where this applies is, that when project grows bigger - you simply try to create sub-project for some parts. Now if work is partially done with some versions - and some open, it becomes impossible!

This feature is very very very very important!

**#11 - 2018-12-26 17:24 - Darko Palic**

having exactly the same issue like Dipan Mehta.

Surely I could recreate the versions, but sadly I have additional attributes in a custom plugin, which would drop all informations

**#12 - 2024-10-09 22:21 - Mario Gleißberger**

We do a lot of project management by the versions/milestones and reorganize some milestones to. Because of this milestones not only have tickets connected to one project/team. We will try to manage the milestones by the teams and milestones have to change into an other project/teams.

I will have a lock to provide a patch. To changes this in the database every time is not a solution.

The connection between milestone and project could be changes by edit the milestone?

The possible selections should be filtered by the user permissions. Otherwise the user can move a milestone to a false project and could not move it again. That's right?