

## Redmine - Defect #3087

### Revision referring to issues across all projects

2009-03-31 23:54 - Zarooba Rozruba

<b>Status:</b>	Closed	<b>Start date:</b>	2009-03-31
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Jean-Philippe Lang	<b>% Done:</b>	80%
<b>Category:</b>	SCM	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.4.0	<b>Affected version:</b>	0.8.2
<b>Resolution:</b>	Fixed		
<b>Description</b>			
Hello			
<b>Short summary</b>			
I am trying to get a commit from a sub project close/reffer to an issue on a main project.			
Nothing is being insert into changesets_issues table.			
Further more, if I insert into changesets_issues table myself, the url to repositories/revision/ is generated to main project (where issue exists) instead of pointing to sub project (where the revision exists).			
Redmine 0.8.2			
<b>Long winded example</b>			
I have a parent project on which users can create issues.			
However, developers have split this project into distinct components, each with own repository (background tasks, project website, nightly processes). For each component I have create a sub project.			
Users do not care about these components.			
I have made such that:			
<ul style="list-style-type: none"><li>- main project has issues module enabled, but repository module disabled</li><li>- sub projects have issues module disabled, but repository module enabled</li></ul>			
This way all user activity within redmine happends on main project.			
Redmine sees repository updates from subprojects and shows them on main project screen.			
However, revision to issue (changesets_issues table) is not working.			
There are two projects			
<ul style="list-style-type: none"><li>- Main project : url web id 'main'</li></ul>			
<a href="http://example/projects/overview/main">http://example/projects/overview/main</a>			
<ul style="list-style-type: none"><li>- sub project (nightly task) : url web id 'sub'</li></ul>			
<a href="http://example/projects/overview/sub">http://example/projects/overview/sub</a>			
issue created ( # 20 ) on main project ( url web id 'main' )			
url : <a href="http://example/issues/show/20">http://example/issues/show/20</a>			
Developer fixes code ( revision 123 on repository id 11 with changeset_id 1234)			
url : <a href="http://example/repositories/revision/sub/123">http://example/repositories/revision/sub/123</a>			
Developers commit comment is : "" commit fixes # 20 ""			
When I visit revision from redmine			
url : <a href="http://example/repositories/revision/sub/123">http://example/repositories/revision/sub/123</a>			
the string # 20 links properly to my issue:			
url : <a href="http://example/issues/show/20">http://example/issues/show/20</a>			
However, There is nothing inserted into changesets_issues			

I have tried, to create a link by hand:

```
INSERT INTO changesets_issues ( changeset_id , issue_id )  
VALUES ( 1234 , 20 ) ;
```

This worked, now I see ""Associated Revisions"" under my issue # 20. However, the url is :

url : <http://example/repositories/revision/main/123>

instead of

url : <http://example/repositories/revision/sub/123>

### closing notes

So two problems are :

- 1) comment from commit does not close/refer to an issue, if commit is different project than issue
- 2) once there is cross project entry in changesets\_issues table, the issue page generates wrong url to repository revision (ignores the possibility that commit revision on an issue is from different project)

Thank you for reading this, and thank you for looking if this is something to update.

Kind regards

### Related issues:

Related to Redmine - Defect #4163: Allow cross-project references in commits	Closed	2009-11-04
Related to Redmine - Defect #3376: Moving an issue to another project copies ...	Closed	2009-05-17
Related to Redmine - Feature #3346: Support for cross-project revision links	Closed	2009-05-12
Related to Redmine - Feature #779: Multiple SCM per project	Closed	2008-03-04
Has duplicate Redmine - Feature #12726: Display the associated revisions with...	Closed	

### Associated revisions

#### Revision 8749 - 2012-02-02 18:43 - Jean-Philippe Lang

Fixed: associated changesets from other projects are not visible if the current project doesn't have the repository module enabled (#3087).

#### Revision 8750 - 2012-02-02 19:09 - Jean-Philippe Lang

Adds a setting that allows to reference/fix issues of all projects (#3087).

### History

#### #1 - 2009-04-06 10:07 - Mohan Varma

+1

We are having the same issue.

Regards,  
Mohan.

#### #2 - 2009-05-11 01:29 - Lincoln Stoll

Isn't this the same as issue [#2255](#) , and possibly implemented by the patch in [#2256](#)?

#### #3 - 2009-05-11 02:02 - Lincoln Stoll

- *File commit\_update\_parent\_project.patch added*

This is a combination of the patch in [#2256](#) and an update to the issue page to make it link to the correct project in the associated revisions.

This should fix this issue - it allows you to commit to a subprojects repository, and update the parent project.

The patch is against rev 2731 (0.8.3)

Does anything else need to be done to make this visible and get it accepted?

#### #4 - 2009-05-12 02:46 - Lincoln Stoll

Note - you will need to have the repository module enabled in the master project (even if it doesn't have a repository) to get the associated revisions details in the issue

#### #5 - 2009-05-12 04:24 - Zarooba Rozruba

The patch to the file `app/views/issues/_changesets.rhtml` should be accepted no matter what, after all, the db schema allows for linking repository changesets across projects.

I am testing the changes to `app/models/changeset.rb`.

This sounds amazingly promising. I will keep you up to date.

#### #6 - 2009-05-19 19:11 - Zarooba Rozruba

- % Done changed from 0 to 30

Jean-Philippe Lang committed part of the fix through ticket [#3376](#)

#### #7 - 2009-05-19 22:59 - Bob Roberts

Zarooba Rozruba wrote:

Jean-Philippe Lang committed part of the fix through ticket [#3376](#)

I've applied the attached patch and amended line no.83 in `changeset.rb` so that any revisions that contain issue IDs for sibling projects are also properly linked in those projects. This has been working great for us for a while now. The recent commit above also fixes the case where tickets are moved between projects and ensure the revision association is always correct.

```
project_list = [repository.project.self_and_siblings] + repository.project.ancestors
```

The use case for us was simple, we have a master project with 10 sub projects and some issues are fixed that affect say 3 projects in one go so are committed together in one revision. In this case it would be useful to retain the revision association for all issues across these projects (they all have the same root repository path anyway)

To get this patch committed into trunk, would it be sensible to add some config options per project under the repository module that can define the expected behaviour of this? i.e. Revisions committed against this project that reference tickets in [self],[self + siblings], [self + all ancestors, self and parent] are allowed?

As a setup note, for any parent/ancestor projects, make sure you have enable the repository module to allow revision linking to work correctly across projects (you do not have to configure it)

#### #8 - 2009-09-03 20:31 - William Baum

This is awesome, but I took it further with a simple change..

The ability to associate issues from ancestor projects is a start, but I found it still way too limiting. We have different repositories by the type of stuff (database objects vs. visual studio projects, requirements documents, etc.) Issues need to be associated with all of those.. So why limit it at all?

```
#project_list = [repository.project] + repository.project.ancestors
project_list = Project.find(:all)
```

This allows any change in any repository to be associated with any issue. If the user doesn't have access to the other repository, they get an error message.. perfect. Furthermore, you can create dummy projects just to hold repositories and now you can have as many repositories as you want..

I have this running in redmine 0.8.4 as well as a recent trunk, and it's working great!

My experience with ruby is extremely limited, so if there's a better syntax or something, please let me know.

--Bill

#### #9 - 2009-09-09 00:13 - Roman Musin

+1

It is real need.

#### #10 - 2009-12-14 13:13 - Thomas Pihl

+1

#### #11 - 2009-12-16 12:53 - Markus Meier

+1

#### #12 - 2009-12-24 12:39 - Lalit Roul

make it +1 from me too..

#### #13 - 2009-12-24 12:46 - Lalit Roul

forgot to mention an important thing if it can be considered, our requirement is not strictly a parent-child requirement, so we have kept project all at top level. But would still like to have a cross project issue-revision relation.

#### #14 - 2010-03-24 11:23 - Geordee Naliyath

I need this too. We have different subprojects created for different modules and there is a user acceptance testing subproject. The commits in the module-specific subproject is not linked to an issue in user acceptance testing subproject. I usually hold myself back from applying patches. Hope to get this issue resolved in the trunk sometime soon.

#### #15 - 2010-03-24 16:12 - William Baum

- File 100324-cross-project-revisions.patch added
- Status changed from New to Resolved
- % Done changed from 30 to 80

I too would like to see this in the trunk, perhaps controlled by an "Allow cross-project revisions" admin setting.

In recent versions, this has become even easier to achieve, as the trunk allows revisions to refer to different projects in the same tree. It does this by checking all of the referenced issues in the revision to see if they refer to the current project or its ancestors or descendants in `app/models/changeset.rb`:

```
# Finds issues that can be referenced by the commit message
# i.e. issues that belong to the repository project, a subproject or a parent project
def find_referenced_issues_by_id(ids)
  return [] if ids.compact.empty?
  Issue.find_all_by_id(ids, :include => :project).select {|issue|
    project == issue.project || project.is_ancestor_of?(issue.project) || project.is_descendant_of?(issue.pr
object)
  }
end
```

To get cross-project revisions, we just need to skip those checks:

```
# Finds issues that can be referenced by the commit message
# i.e. issues that belong to the repository project, a subproject or a parent project
def find_referenced_issues_by_id(ids)
  return [] if ids.compact.empty?
  Issue.find_all_by_id(ids, :include => :project).select {|issue|
    project == project
    ###project == issue.project || project.is_ancestor_of?(issue.project) || project.is_descendant_of?(issue
.project)
  }
end
```

I'm attaching an svn formatted patch against [r3606](#), which is still current, as `changeset.rb` is currently at [r3473](#). While patches are cumbersome to maintain, this one's really a one-liner in one file.

#### #16 - 2010-06-19 23:45 - Felix Schäfer

- Status changed from Resolved to Closed

As far as I can tell, something similar (or that exactly) exists in current trunk, probably 0.9-stable too, closing this issue.

#### #17 - 2010-06-21 16:03 - William Baum

- Status changed from Closed to Reopened

This is still valid, as the trunk allows revisions only to issues in related projects (parents, children). This change allows revisions in any repository to be associated with an issue in any project.

This is an essential part of the way we use Redmine in our organization, and I personally see no benefit in preventing revisions from being associated with issues, but perhaps this should be controlled by a permission setting?

#### #18 - 2010-11-26 12:35 - John Bart

+1 here. It could be controlled by the existing admin setting to allow cross project issue linkage. If you need that, chances are more than well that you'd require this patch here as well. Would be great to see such a change in the trunk.

**#19 - 2011-03-07 07:07 - Bob Bottle**

+1 to William Baum's comment 17. Would be good if issues were related to any project in redmine, not just parent/child projects.

**#20 - 2011-03-07 07:26 - Toshi MARUYAMA**

- *Category changed from Issues to SCM*

**#21 - 2011-07-21 07:10 - Martin Denizet (redmine.org team member)**

Felix Schäfer wrote:

As far as I can tell, something similar (or that exactly) exists in current trunk, probably 0.9-stable too, closing this issue.

I tested in 1.2.0, using commit messages, you can indeed create references to issues in ancestor or descendant project. However the ancestor or descendant project **must** have repository configured.

I don't see the reason for it to be this way.

In my organization, we have a project with a repository for the product development and sub-projects for customers implementations (With Gantt for deployment planning and specific issues). We are not able yet to link revisions to customers issues.

**#22 - 2011-07-22 10:31 - Artur Wasilewski**

+1

**#23 - 2011-09-05 03:34 - David Chiew**

+1 please!

**#24 - 2011-12-21 20:08 - William Baum**

My apologies, I intended to update this months ago..

In app/models/changeset.rb you'll find:

```
def find_referenced_issue_by_id(id)
  return nil if id.blank?
  issue = Issue.find_by_id(id.to_i, :include => :project)
  if issue
    unless issue.project &&
      (project == issue.project || project.is_ancestor_of?(issue.project) ||
       project.is_descendant_of?(issue.project))
      issue = nil
    end
  end
  issue
end
```

It's that unless business that's preventing the links unless the projects are related.

Just comment out that whole if block or the change the unless to:

```
unless project == project
```

This will enable changesets in any repository to reference issues in any project.

**#25 - 2012-01-16 21:35 - William Roush**

+1, should be in trunk, commits from other projects may influence related projects.

**#26 - 2012-02-02 19:10 - Jean-Philippe Lang**

- *Subject changed from Revision referring to issues across projects to Revision referring to issues across all projects*

- *Status changed from Reopened to Closed*

- *Assignee set to Jean-Philippe Lang*

- *Target version set to 1.4.0*

- *Resolution set to Fixed*

This is now fixed:

- related changesets are displayed on the issue even if the repository module is disabled on the issue's project ([r8749](#))
- a new setting is added in [r8750](#) to allow issues from any project to be referenced/fixed (by default, only issues of parent and subprojects can be referenced/fixed)

## Files

commit_update_parent_project.patch	2.58 KB	2009-05-11	Lincoln Stoll
100324-cross-project-revisions.patch	641 Bytes	2010-03-24	William Baum