Redmine - Feature #9703

Define repositories independently from projects

2011-11-30 16:15 - Hugues De Keyzer

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	SCM	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

It is common for multiple projects to use the same repository. When a repository is set up in a parent project, references to and from children work, although no “Repository” link appears in the children’s primary links. I don’t know whether this is supposed to work or not, since the feature request #1657 about this is not resolved.

I think that it would be more interesting if repositories were defined independently from projects. A project would then select the repository it uses from the list of defined repositories. This could also allow for multiple repositories per project.

Related issues:

Related to Redmine - Feature #779: Multiple SCM per project	**Closed**	**2008-03-04**
Related to Redmine - Patch #9359: invert project <-> repository relationship	**New**	**2011-09-30**
Related to Redmine - Feature #23079: Same Repository to differents projects	**Closed**	

History

#1 - 2011-11-30 16:47 - Hugues De Keyzer

The subject of this feature request should be “Define repositories independently from projects”.

#2 - 2011-11-30 17:23 - Etienne Massip

- Subject changed from Define repositories indepently from projects to Define repositories independently from projects

- Category set to SCM

I think it's a dupe too.

#3 - 2011-11-30 17:45 - Hugues De Keyzer

The following issues are related to this one, but I wasn’t able to find one proposing a similar solution: #779, #1657, #2255, #3087, #3169, #3346, #3687.

#4 - 2011-12-05 00:01 - Mischa The Evil

Hugues De Keyzer wrote:

> The following issues are related to this one, but I wasn’t able to find one proposing a similar solution: #779, #1657, #2255, #3087, #3169, #3346, #3687.

Indeed. I've added a relation to #779.

#5 - 2011-12-21 16:59 - Colin Mollenhour

This sounds like the best all-around solution, better than #779. It solves duplicate commit messages and the whole parent-child-sibling issue. For me the biggest issue is I have multiple git repos by creating dummy projects under the parent project, but then if there is some commit that references an issue of a sibling repository the connection is not made. With svn I could work around this because I would organize the repo around my sub-project layout but with git having multiple repositories is the only way to control access to specific submodules.

#6 - 2011-12-21 19:52 - Andy Bolstridge

I'm not convinced. While it does sound 'clean' to refer to repositories separately, I think most projects will still use unique repositories themselves. Ie, the code for each project does not reside in the same place, either because they are in distinct repositories, or in distinct sub-sections of a single repo.

As a result, we'd end up defining repository links in one place, only to refer to each in the projects - the same could be achieved by placing the repo links directly in the project.

This does not mean that we could take all the repo links for all projects and have them displayed in a report elsewhere that allows the admin to change them (eg if a repo moves).

#7 - 2011-12-21 20:10 - William Baum

I really like this idea, and actually considered writing up something similar myself.

I don't see it as inconsistent with #779. It would be great to have available repositories and associate them with projects as need be.. Multiple repositories per project and multiple projects per repository.

Colin, the cross-project revision references thing is easy.. See #3087

#8 - 2011-12-22 08:27 - Pedro Gutierrez

I like the idea very much. It would make our life simpler and more flexible.
Mainly because our problem is not that we need multiple repositories for a project, but the other way around, i.e. we'd need multiple projects per repository.

So I'd prefer this solution to the one proposed in #779 . However,I see a number of open issues:

Issue1: *Any suggestion on how to grant access to the repositories?*

The easiest way, I'd say, would be that repositories had a list of associated projects.
And keep the rest as it is at the moment, i.e. the access to the repository granted at role level.
So if I want to push a change to a repository I'd need:

- to be involved in a project with a role that provides access to the repository
- that the project itself be associated to that repository

Issue2: *If I'm involved in two projects (with the appriate role). And both projects are sharing a repository how do I associate each commit with the project?*

#9 - 2012-05-17 17:26 - Holger Winkelmann

We also suggest to have repos independently of projects and assign them to one or more projects,

#10 - 2013-05-18 02:13 - R. Steve McKown

+1

This might greatly help larger team use of redmine. We have about a dozen shared code git repos, used by several dozen unique applications, each of which is built from its own application level git repo and git submodule like links to the relevant shared git repos. Since we develop apps for customers, the number of apps is growing, and the feature set made available by the shared code is also growing. It's a massive N:N relationship between app code and shared code.

We want to enter feature and defect issues against the project that triggers the issue. But our shared code base is getting so comprehensive now that about half of any project's issues end up generating commits against shared code. We don't want to use #779 to attach the shared repos to all the relevant projects, as this would result in massive commit data duplication in redmine. The concept of this #9703 would look to solve this challenge for us.

Edit: I just found the global setting "Allow issues of all the other projects to be referenced and fixed". Enabling this checkbox seems to be a good solution to use issue-commit relations in our shop.

#11 - 2013-05-23 05:31 - Bruno Medeiros

R. Steve McKown, don't you have a separated redmine project for each shared code library (repo)?
I have a similar setup of yours, but I create a project for each library (shared code repo), and every time a new commit is needed on the shared code, I create a new issue on the specific project of the shared code.
PS.: I still would like to see this ticket implemented, but not for this specific case.

#12 - 2016-06-14 12:24 - Zer Guz

+1. In our company we have many repository modules that are used in multiple projects. It will help if a repository can be associated with multiple projects.

#13 - 2016-07-05 07:38 - Toshi MARUYAMA

- Related to Feature #23079: Same Repository to differents projects added

#14 - 2021-09-26 21:45 - Tolga Uzun

We are using a main repositoyory for a product and managing the different versions of the project in the same repository but differentiating in folder hierarchy but not at root level, inside other folders (literature, design etc and than the version folders). And also there are different projects in Redmine that is used to track the issues. This way, I have to add the main repository to all of the parent and child projects that causes fetching all repository changesets to the redmine database multiple times which takes enormous time for a big old repository.

I think this feature would be very useful and the solution is already provided: Just make the repositories independent of the projects and implement a relation between them.

By the way, I am new using the repository module so I may be wrong about the solution since I only use it for listing the changesets and relating them with the issues.

In short: +1