

Redmine - Defect #3843

Better issue identifiers

2009-09-10 13:34 - XXX XXX

Status:	Closed	Start date:	2009-09-10
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:		Affected version:	0.8.4
Description			
The problem at the moment is that numbering is cross project. For example: I have 2 projects: PHM and Alfresco. I make my first 2 issues in the project Liferay: #1 and #2 . Then I make my first issue in Alfresco but this issue gets #3 . If you have a lot of projects (like in our company) you can start your issues in project 10 with #10989 which isn't logical from a project perspective. It is also harder to remember the number of an issue this way. I suggest that numbering starts from #1 at each project. Then the url to see an issue could be something like this: issues/alfresco/4. Another option is that the project identifier is prefixed to the id and it still starts from #1 at each project. Then the url could be something like this: issues/phm-4. Probably this incorporates some database changes but I think it really is a good improvement to the program. (the URL changes would probably relate to issue #1901)			
Related issues: Related to Redmine - Feature #5765: Issue Numbers per Project Is duplicate of Redmine - Feature #282: Ability to specify a project code			
Closed 2010-06-27			
Closed			

History

#1 - 2009-09-10 13:35 - XXX XXX

Maybe it would take a little more time to develop cause I haven't taken into account links to issues in textfields. # + issue number would not suffice anymore, except when the project identifier would be integrated with the id (or only using that syntax when linking in the textfield)

#2 - 2009-09-10 16:26 - Adam Piotr Żochowski

How do you link issues between projects? (aka: issue # 35 from project A blocks issue # 35 from project B)

How do you link issues in messages?

How do you move an issue? (aka: issue#35 is moved from project A into project B)

How do you link commit messages to issues (especially with [#3087](#) a commit to project A can close a ticket in parent/child project B, how does one handle this message?)

It is simplest to keep the issue numbers as they are.

Kind regards

Adam Żochowski

#3 - 2009-09-10 16:36 - XXX XXX

Adam Piotr Żochowski wrote:

How do you link issues between projects? (aka: issue # 35 from project A blocks issue # 35 from project B)

How do you link issues in messages?

Like I said, we would need to rethink that. It is possible (JIRA does it as well using a project prefix and number, e.g. RAN-34 and RED-34) by just adding more information. For example, linking could be done this way for project A: #35-A and for B: #35-B.

How do you move an issue? (aka: issue#35 is moved from project A into project B)

If you move an issue to a project where its number is already taken, then you just take the highest number used + 1. So issue 35 in project A could become issue 63 in project B.

How do you link commit messages to issues (especially with [#3087](#) a commit to project A can close a ticket in parent/child project B, how does one handle this message?)

Same syntax as with messages (see above).

It is simplest to keep the issue numbers as they are.

Yeah it would probably be simpler to develop as it is now, but towards an end-user experience I think my proposition (the JIRA way) is better. In our company we have more than 20 projects with every project more than 500 issues (stories, bugs, features, support, ...) so it is easy to have issue numbers like 10943 and this just seems messier and more difficult to remember than 35-RED.

Just my two cents.

Thanks for sharing your vision!

Mano Swerts

#4 - 2009-09-12 11:59 - Jean-Philippe Lang

- Status changed from New to Closed

- Resolution set to Wont fix

This was already discussed in [#282](#), [#538](#), [#1926](#).

Using such identifiers when you can move issues is too problematic.

#5 - 2009-09-14 08:04 - XXX XXX

Jean-Philippe Lang wrote:

This was already discussed in [#282](#), [#538](#), [#1926](#).

Using such identifiers when you can move issues is too problematic.

It sure is more difficult to implement than what is implemented now, but I don't think that it is that problematic when it is programmed well. And, in my honest opinion, I do not see the need for moving issues between projects. Projects are all very different to each other so the chance that you will want to move an issue from one project to another is fairly small.

#6 - 2009-09-14 08:16 - Eric Davis

And, in my honest opinion, I do not see the need for moving issues between projects. Projects are all very different to each other so the chance that you will want to move an issue from one project to another is fairly small.

I disagree. I move at least 3 issues every day; either they are entered on the wrong project or after a discussion it's decided that the issue should be filed somewhere else.

I agree with the won't fix resolution.

#7 - 2009-09-14 08:34 - Andrew Chaika

I agree too with this resolution.

But described functionality (additional issue project-related numeration) might be implemented by plugin.

#8 - 2009-09-14 09:25 - Ernad Husremovic

Eric Davis wrote:

I agree with the won't fix resolution.

+1

#9 - 2009-09-17 11:09 - Yonghwan SO

I agree with this behavior now, (because the number is really not important, especially for [#3843](#) or more big number, and I also move some issues between projects.) but at the first sight, I feel strange with it.

anyway, the only bad time is when I communicate with customer. They also confuse with this hopping number and issue [#47](#) of 35 issues, and I do not want to manage map of redmine#47-to-customer#26. so I think, at least for some cases, we need both issue_id(for system) and issue_num(for communication). One existing solution is using custom integer field. but it's very annoying and there is no option like auto-increasing.

how about your side?

#10 - 2010-03-21 23:19 - Kirk Stork

To me, the original reason for this issue is not of concern, however, there are at least two use-cases that per-project issue numbers would help with:

1. Importing from other issue tracking systems. The original issue numbers could be preserved because the imported project is its own new thing.
2. It would allow Redmine to have a capability for exporting and importing projects. Say, to move a project from one installation to another, or to mothball a project to DVD and keep the ability to un-mothball it years later without conflict.

These are both features I would find useful...

Eric Hulser: I also agree with others who want to retain the ability to move issues without them changing their identity. In my organization we use hierarchical project structures, and frequently move issues between sub-projects of an overarching project.

So.. I want it both ways -- not sure what the "real" solution should be. :/

#11 - 2010-11-25 06:29 - James Selvakumar

How about giving issue aliases? I agree with all the benefits of a unique issue id. Yes it helps in many ways. To get the best of both world, we can retain the unique issue identifiers and provide a issue alias. So a issue [#3843](#) can be referred also as #REDMINE-xyz something like JIRA does. Have a look at the [Apache issue tracker](#)

#12 - 2014-08-23 19:03 - Andriy Lesyuk

I implemented the "another option", but instead of the project identifier I used the special project key. See [this page](#).