Support Platforms - Bug #836
Linking to Redmine issues from commits is inconsistent
11/12/2011 02:48 PM - Teemu Murtola

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Szilárd Páll</td>
</tr>
<tr>
<td>Category:</td>
<td>Redmine</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Affected version:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

In different contexts, links from commit messages to Redmine issues behave a bit inconsistently. In Gerrit and in Redmine when viewing a commit, anything like #XYZ is treated as a link to an issue in Redmine. But links from the issue page in Redmine to related commits only works for commits that are specifically tagged with some keywords before the #XYZ. And the actual parsing rules for these keywords are not explained anywhere (how should one, for example, link to more than one issues for both of them to get recognized?). I think it would be best to configure Redmine to add commits to issue pages simply based on the #XYZ tags.

Currently, because of #646, this may result in commits appearing more than once for the issues, but I don't think that's a major problem (and #646 should be solved anyway).

**Related issues:**
Related to Support Platforms - Feature #694: Write instructions/policy for is... Feedback wanted 02/01/2011

**History**

#1 - 11/17/2011 08:01 PM - Teemu Murtola
- Category set to Redmine

#2 - 01/13/2012 09:02 PM - Szilárd Páll
There is a formatting help, the link to it is above the upper right corner of comment box. The only issues is that currently the redmine git plugin is quite messed up and sometimes it takes hours for the link to appear.

#3 - 01/13/2012 09:30 PM - Teemu Murtola
I wasn't referring to comments in issues, they seem to work fine. Currently, in every context in Redmine and in Gerrit, if you write just #836 or similar, it will show up as a link to the Redmine issue. But what doesn't work is that if you just write #836 in a commit message, that commit does not appear on the issue page (like, e.g., in #641). Instead, you need to write "IssueID #836", or some other magic keyword for this link to materialize (just take a look, for example, at https://gerrit.gromacs.org/#change,372 that was merged ages ago and doesn't appear on the relevant issue page). And it isn't very clear how this works, e.g., if you want to link to multiple issues. Should you prefix each hash sign separately with a keyword, or what is sufficient? At a minimum, this should be figured out and documented. These keywords can be configured globally in Redmine, and I think it would be clearest to just enable that linking without any keywords.

#4 - 04/12/2012 08:56 PM - Szilárd Páll
Indeed, it is possible to configure the referencing and it's also possible to have any keyword-less #XYZ in git commit messages to be considered a reference. By default a keyword is required which in fact has been documented on the wiki, just that it was under the gerrit review guide.

Earlier today I moved the description of how to reference issues in git commits to the FAQ on Gerrit which used to contain partial/incorrect information.

Do we want to allow no keywords? It might be good for consistency, but not for clarity: having clearly stated whether the ID is a ref or fix is important. Even more important is that issues actually do get referenced appropriately (and filed whenever required).
#5 - 04/12/2012 09:05 PM - Teemu Murtola

I think that the current situation is not very good. The set of keywords is quite limited, and it isn't easy to use them correctly while trying to write a fluent description of the change.

As for clarity, I think that it will not make things any worse if we remove the keyword requirement for the reference. I think there are very few if any situations where you want to mention a Redmine issue in a commit, but don't want the link to appear, so the requirement on the keywords is just a nuisance. There will still be those keywords for marking a commit as a fix.

#6 - 04/13/2012 04:43 AM - Mark Abraham

Szilárd Páll wrote:

Do we want to allow no keywords? It might be good for consistency, but not for clarity: having clearly stated whether the ID is a ref or fix is important. Even more important is that issues actually do get referenced appropriately (and filed whenever required).

Allowing no keywords seems good to me. Anything that makes cross-referencing easier has to be good for a distributed project. Lowering the barrier for new contributors, and the memory requirements for existing contributors is all good. Context will likely make any spurious links ("this is our #1 priority") obviously spurious.

#7 - 04/13/2012 06:44 AM - Teemu Murtola

Just an addition to my previous comments about what is not clear and/or easy:

1. The word "references" isn't very useful when trying to explain in more detail how a commit relates to an issue that it does not fix, and IssueID in the middle of text is a bit awkward as well.
2. It's not clear how Redmine treats, e.g., "references #123 and #234", or "reference #123", or "These changes fix #123", or "IssueIDs: #123, #234 ", etc. All of these (or some other variations) would be reasonable things to include in a commit message if trying to write proper English instead of some jargon.

#8 - 04/18/2012 11:28 PM - Szilárd Páll

Mark Abraham wrote:

Allowing no keywords seems good to me. Anything that makes cross-referencing easier has to be good for a distributed project. Lowering the barrier for new contributors, and the memory requirements for existing contributors is all good. Context will likely make any spurious links ("this is our #1 priority") obviously spurious.

I'm not sure that allowing now keywords will improve the quality of referencing. What I would like to ideally see is the proper use of referencing so that e.g all important bugs get filed and appropriately referenced in the commit so that the status of the respective issue can be automatically changed to "Resolved" (which we'd need to add). However, by enabling keyword-less referencing, I fear that this might not work as most will stick to what requires the least effort: never using keywords (if referencing issues at all).
Just an addition to my previous comments about what is not clear and/or easy:

1. The word “references” isn’t very useful when trying to explain in more detail how a commit relates to an issue that it does not fix, and IssueID in the middle of text is a bit awkward as well.

I see no problem with adding/removing keywords, I can do that anytime. Suggest better keywords and I’ll add them in Redmine and document them on the wiki.

1. It’s not clear how Redmine treats, e.g., “references #123 and #234”, or “reference #123”, or “These changes fix #123”, or “IssueIDs: #123, #234”, etc. All of these (or some other variations) would be reasonable things to include in a commit message if trying to write proper English instead of some jargon.

The instructions on the wiki are based on the documentation and on the settings we have in Redmine (copying from the admin page):

- Referencing keywords: refs, references, IssueID
- Fixing keywords: fixes, closes (Applied status: none)

As the docs are not very clear we’d have to test which conjunctions work (“and”, “,” “.”).

There are three different things here:

1. Redmine bugs/tasks/whatever are not always created, or if they exist, they are not always referenced in commit messages.
2. In order to reference an issue, one currently has to use some keyword, and the alternatives are not flexible. Even without any keyword, a hyperlink appears from the commit message, both in Gerrit and in Redmine. But the reverse link from the issue only appears with the keywords, which is simply confusing.
3. Another set of keywords is needed to change the status of the related issue directly from the commit message.

I simply don’t see how changing 2. implies that 1. will somehow get worse than it is now. To me, these are completely unrelated. And I’m not proposing to change 3. Also, if the automatic status change doesn’t work for some reason, it will be easier later if the link from the issue is still created. And the most foolproof way of ensuring that the link exists is to allow any keywords.

I see no problem with adding/removing keywords, I can do that anytime. Suggest better keywords and I’ll add them in Redmine and document them on the wiki.

Well, if you are so fervently against removing the keywords for referencing, then I suggest that we add "issue", "task", "feature", and "bug" (and their plurals if required) to the list. IssueID is already there, so this would just make the language more natural. But I still don’t understand how this will then differ from not requiring any keyword at all, except for the extra typing. There is a possibility for unintended links, but that doesn’t seem to be a
concern to you.

1. It's not clear how Redmine treats, e.g., "references #123 and #234", or "reference #123", or "These changes fix #123", or "IssueIDs: #123, #234", etc. All of these (or some other variations) would be reasonable things to include in a commit message if trying to write proper English instead of some jargon.

The instructions on the wiki are based on the documentation and on the settings we have in Redmine (copying from the admin page):

- Referencing keywords: refs, references, IssueID
- Fixing keywords: fixes, closes (Applied status: none)

As the docs are not very clear we'd have to test which conjunctions work ("and", ";", ":").

Yes, I noticed that the instructions are directly from the redmine documentation, which I also find inadequate. The list to test should also include plural forms and verb suffixes (it might be easier to simply find the location in the Redmine code where this is done and read it from there...). Capitalization at least doesn't seem to matter.

#11 - 04/19/2012 06:37 AM - Teemu Murtola

Just for reference, a bit of history why did I bring this issue up?

When Redmine was taken into use, I read the documentation about this linking, found it lacking sufficient detail to understand its limitations, and decided to simply use a separate IssueID #NNN line at the end of my commit message to do the linking.

When Gerrit came, the original announcement (and the wiki page also for a long time) said that to link to an issue, you only need to use #NNN. I suspected that this only related to Gerrit configuration, but there was a bit of hope that it would have actually been thought out also from the Redmine side. So I started writing more useful information about the link in the commit message, only using #NNN. But it started to be a bit awkward to manage the issues without the commits automatically linked, so I brought this up.

I can of course go back to my original way, but for review purposes, I think it makes a lot more sense to actually explain in the commit message how is the commit related to the linked issue. For a simple bug, "fixes #NNN" is sufficient, but for a more complex task, it may not be that apparent if the commit only partially implements something or resolves some prerequisite issues. With the current linking rules, it is inconvenient to actually write such an explanation: either you need to prefix everything with IssueID, or use 'references', which doesn't add any useful information.

#12 - 04/26/2012 11:47 PM - Szilárd Páll

I/we gave this issue a thought and we concluded that the most useful setup would be the following.
Enabling referencing without keywords.

Introducing a new issue status and automatically switching to this when a fix is committed. I would suggest “Fix submitted”. We might want extend the list of fixing keywords, currently we have: fixes, closes.

Ensure while reviewing that fixes for issues present in Redmine get properly referenced.

Additionally, it has been discussed that in order to make compiling release notes easier, in case of severe bugs, we should always file a bug report and reference it in the commit appropriately.

We might want to move the discussion to the dev list to get more feedback or comments.

---

#13 - 04/27/2012 09:41 AM - Teemu Murtola

The proposed setup seems good to me. I’ve been missing such a “Fix submitted” state ever since “Resolved” was removed (have been using “Feedback” for that, but that state is quite vague, and different people probably see it differently).

---

#14 - 06/14/2012 05:06 PM - Szilárd Pálí

Szilárd Pálí wrote:

I/we gave this issue a thought and we concluded that the most useful setup would be the following.

- Enabling referencing without keywords.

Should work with just #YXZ now, sorry for the delay.

- Introducing a new issue status and automatically switching to this when a fix is committed. I would suggest “Fix submitted”. We might want extend the list of fixing keywords, currently we have: fixes, closes.

Rossen: could you please look into restoring this functionality, many of us want it!

- Ensure while reviewing that fixes for issues present in Redmine get properly referenced.

Additionally, it has been discussed that in order to make compiling release notes easier, in case of severe bugs, we should always file a bug report and reference it in the commit appropriately.

We might want to move the discussion to the dev list to get more feedback or comments.
Szilárd Páll wrote:

Should work with just #YXZ now, sorry for the delay.

Great, thanks. I think this issue can now be closed, as the problem in the description is solved. Assigned to you, Szilard, because you fixed it. :)  

- Introducing a new issue status and automatically switching to this when a fix is committed. I would suggest "Fix submitted". We might want extend the list of fixing keywords, currently we have: fixes, closes.

Rossen: could you please look into restoring this functionality, many of us want it!

I propose continuing the discussion on issue states in [#694](#694). Will try to find some time to write my thoughts there.