Support Platforms - Task #1264

Updating Redmine issue statuses from Gerrit

05/24/2013 01:56 PM - Rossen Apostolov

<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>Rossen Apostolov</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

Description

Moving the discussion about gerrit-redmine interaction away from [#694](#694).

The gerrit hook at [https://github.com/tru/redmine-gerrit-scripts](https://github.com/tru/redmine-gerrit-scripts) is able to post comments and update the status of redmine issues but as Teemu pointed out:

It won't do without heavy modification. It has several issues in handling either multiple issue references in one commit, or cases where multiple commits are necessary to resolve an issue, or links like "Related to #xyz". A separate post for each patchset may also be excessive.

We need to decide how exactly such hook should behave and I'll work on extending the functionality.

History

#1  - 05/24/2013 01:57 PM - Rossen Apostolov

- Description updated

#2  - 05/24/2013 02:27 PM - Teemu Murtola

One question is which credentials the plugin should use to get into Redmine. Does the Redmine REST API allow for impersonating someone without knowing the API key? This would allow mimicking the behavior of Redmine when it puts issues into the Resolved state. If not, we probably want to create a separate login for the plugin (something like "Gerrit Code Review"). Will write other thoughts later.

#3  - 05/24/2013 04:31 PM - Teemu Murtola

At minimum, the plugin should do the same distinction between issue links as Redmine currently does: if a certain keyword appears, then that issue is going to resolve the issue, otherwise it is only related. And it should be able to handle multiple issue references in a single commit message. It should also take into account draft changes (which didn't exist at the time the original plugin was written).

One possible minimal (in the sense of implementation complexity and impact on Redmine post count) behavior could be:

- For each new patch set (drafts excluded, but publishing a draft patch set included), change the state of those issues to "Fix uploaded" that are linked such that Redmine would change them to Resolved, and the issue is not yet in that state.
  - At minimum, the post should include the Change-Id, link to the gerrit page, and the name of the author and/or committer of the issue (more or less what the hook currently does).

Additional, possible nice-to-have features (I'm not suggesting we have this initially, but instead collecting some thoughts here):

- If impersonation is possible through the REST API, then the author could be indicated also as the user posting the comment, like Redmine currently does for issues it sets to Resolved.
- If for each patch set, post a link to each issue that is mentioned in the commit message (not sure whether this would generate excessive number of posts on Redmine without the first point below).
  - Don't post the link if a previous patch set is already linked in the issue.
  - If a draft patch set is added, also post something (perhaps not a link, since people not involved in the draft can't access it), but don't yet change the status (or change it to In Progress if in New or Accepted).
  - Also post a link if a change gets abandoned or restored.
  - Possibly also post something if a linked change gets merged.
- Possibly make the same plugin responsible of changing the issue state to Resolved to make sure both transitions follow the same rules.
Teemu's analysis seems good.
I'd clarify that "Don't post the link if a previous patch set is already linked" means don't keep linking for successive patches of the same gerrit issue mini-branch (unless it's been abandoned and resumed, or changed from draft to public, or such). If there's two Gerrit issues that reference the same Redmine issue, the plugin should treat them as independent equals.

---

- Status changed from Accepted to Feedback wanted

Here is how it works at the moment:

- It looks for "#12345" issue IDs in the commit messages and if found, changes the status to "Fix uploaded" for all of them. It doesn't distinguish whether the issue ids are referred to as "related to #1234" or "fixes #1234" etc. "Fixes #1234" will change the status to "Resolved" once the change passes the gerrit review, and this is done by redmine itself.
- The hook processes also drafts. In that case the status is changed to "In progress".
- The script updates the status and posts a message only after the first patchset. Subsequent ones are ignored.
- In addition to a change in the status, the script posts a message in the redmine issue as user "Gerrit Code Review Bot" with the following example info (e.g. in http://redmine.gromacs.org/issues/1403):

  ---
  Updated by Gerrit Code Review Bot less than a minute ago
  Gerrit received patchset '1' for Issue #1403.
  Uploader: Rossen Apostolov (rossen@kth.se)
  Change-Id: Id8c4f9b3b812ce89e9f201cb5daeaf899f46c01ab
  Gerrit URL: https://gerrit.gromacs.org/2877
  ---

---

#6 - 12/12/2013 03:25 PM - Szilárd Páll
Sounds good as a first step.

Rossen Apostolov wrote:

Here is how it works at the moment:

- It looks for "#12345" issue IDs in the commit messages and if found, changes the status to "Fix uploaded" for all of them. It doesn't distinguish whether the issue ids are referred to as "related to #1234" or "fixes #1234" etc. "Fixes #1234" will change the status to "Resolved" once the change passes the gerrit review, and this is done by redmine itself.
Not all issues are bugs, we have been using tasks and features quite extensively. In previous discussions it has been emphasized extensively that both issue referencing, statuses, as well as automated updates need to just make sense without getting in the way. Hence, I think it would be beneficial to make sure that referencing is done correctly and fine-grained enough.

- A change that references an issue can do so for multiple reasons; very often referencing does not indicate that the issue is solved. Instead it often means that a partial fix/implementation or a loosely related change has been uploaded, e.g. required in order to solve the referenced issue, etc. Hence, uploading a change related to an issue should **not** trigger an automated issue state change to "Fix uploaded." I'd even suggest avoiding automated changes to the issue status without contextual/keyword parsing this will often be simply incorrect. E.g. we often use such referencing: "This is required in order to solve (or implement) #1234" which indicates neither that a "fix is uploaded for bug #1234" nor "that solution for issue #1234" is in the respective change. To conclude, if a change is a "Fix"/"Implementation" (identifiable through a referencing keyword), a status change is warranted and useful! Otherwise, I suggest only keyword-based status changes, otherwise the gerrit hook should only post a comment with a text e.g. "Related gerrit change uploaded: https://the.change.url".

- I suggest that with the new freshly installed redmine versions' new features, we should consider fine-tuning referencing keywords and issue statuses for the different trackers. IMHO the effort of distinguishing between bugs and features/tasks is very small. (Note that suggestions - by Teemu and me, see #694, note 50 - on using a more suitable status wording have been ignored. Please consider them again.)
  - The hook processes also drafts. In that case the status is changed to "In progress".

As explained above above, a simple referencing of an issue from a change does not mean that the respective issue is being worked on, to make such conclusions, keyword parsing is necessary. Without that, I suggest only posting a message e.g. "Related gerrit draft change uploaded: https://the.change.url" (note, that you may not have access to view the draft change)"

However, as drafts are by default private, posting the full issue header violates visibility rules set up in gerrit and the URL is only useful for those with explicit access granted for the draft.

- The script updates the status and posts a message only after the first patchset. Subsequent ones are ignored.

Sounds good - although depending on how we treat drafts, it may be worth posting a message only or also after a change becomes non-draft.

- In addition to a change in the status, the script posts a message in the redmine issue as user "Gerrit Code Review Bot" with the following example info (e.g. in http://redmine.gromacs.org/issues/1403):

---

Updated by Gerrit Code Review Bot less than a minute ago

Gerrit received patchset '1' for Issue #1403.
Uploader: Rossen Apostolov (rossen@kth.se)
Change-Id: Id8c4f9b3b812ce89e9f201cb5dae899f46c01ab
Gerrit URL: https://gerrit.gromacs.org/2877
---

Looks good, but see my comments on drafts above.
#7 - 12/13/2013 02:09 PM - Rossen Apostolov
It will be difficult indeed to do proper parsing of the commit message and choose the correct status. So for now I disabled the status updating, and only a message will be printed out. The message will also mention whether it's a draft.

#8 - 12/19/2013 06:14 PM - Szilárd Páll
So how about adding some nicer keywords for tasks and features (implemented, solved)? Additionally, as we have a required keyword for auto-resolving issues (after merge), in such cases we know what the reference means, so upon matching the referencing keyword we could update the status to e.g. "Implemented" (features) or "Fix uploaded" (bugs).

#9 - 01/17/2014 08:38 PM - Rossen Apostolov
that's good but then, when writing the commit message, the developers need to remember whether the issue they were working on was defined in redmine as "task" or "feature", and what keyword to use. I'd say lets keep it this way?

#10 - 06/10/2014 01:43 PM - Rossen Apostolov
- Status changed from Feedback wanted to Closed

#11 - 06/19/2014 10:09 PM - Szilárd Páll
Note that AFAIR drafts still trigger a redmine message which is not correct as drafts will most of the time not be visible to the public, typically not even to the author of the issue or watchers.

#12 - 06/20/2014 09:52 AM - Rossen Apostolov
drafts trigger a message that explicitly says "drafts", capitalized. Have a look at the last message in http://redmine.gromacs.org/issues/1403

and the point is only to notify that someone is working on the issue, and who that person is, if you want to look at the draft or test it, you can message the person or put a comment in the redmine discussion, otherwise, how would you prefer it to work?

#13 - 06/20/2014 12:20 PM - Szilárd Páll
I see little reason in notifying every watcher of an issue about a draft because a draft is anyway restricted in access to those whom the owner of the change explicitly shared it with. Not a big deal, it's only a minor annoyance.

#14 - 06/20/2014 12:47 PM - Rossen Apostolov
the benefit is that the watchers are aware of ongoing work. otherwise there's a risk of duplicating efforts

#15 - 06/20/2014 02:28 PM - Szilárd Páll
For that setting the assignee and the status to "In progress" should suffice, right? Anyway, as I said, I don't think it's a big deal, it's just somewhat confusing to get a link to the ongoing private work posted publicly.

#16 - 07/07/2014 06:35 PM - Roland Schulz
It isn't ideal that we only analyze the first patchset. The main problem: If the first patchset is missing the reference then we never update redmine. As an example see: https://gerrit.gromacs.org/#/c/3750, I just updated the commit message but no link was added to redmine.

#17 - 07/07/2014 07:18 PM - Rossen Apostolov
Maybe this happens if you edit the commit message directly from gerrit. It probably works if you do "git commit --amend".

#18 - 07/07/2014 07:20 PM - Roland Schulz
I don't think so. I thought the script we did was simply checking whether patchset is 1. Thus it wouldn't make a difference with amend.
yes, you're right. but if we remove that condition then every new patchset will add a new message to the redmine issue and flood the discussion with mostly identical messages.

It isn't rocket science (but certainly more complex than the current script) to add some state to the script by either

1. keep a local db.
2. parse previous patch sets from git, or
3. query redmine for comments on the associated issues and check those posted by the script

Any of these would allow avoiding excessive posts.