

Redmine - Defect #5354

Updating custom fields does not trigger update to "updated_on" field in the customized object

2010-04-20 00:11 - Stuart Cianos

Status:	Closed	Start date:	2010-04-20
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Custom fields	Estimated time:	0.00 hour
Target version:	4.2.0	Affected version:	0.9.3
Resolution:	Fixed		
Description			
When updating a custom field (which has been defined within the Redmine settings), the updated_on field in the Projects table is not updated to reflect the date of modification to the project's settings.			

Associated revisions

Revision 19510 - 2020-02-11 05:06 - Go MAEDA

Updating custom fields does not trigger update to "updated_on" field in the customized object (#5354).

Patch by Yuichi HARADA.

Revision 19511 - 2020-02-11 05:56 - Go MAEDA

Reverts r19510 that breaks IssueTest#test_closed_on_should_be_set_when_closing_an_open_issue (#5354).

Revision 19541 - 2020-02-29 07:42 - Go MAEDA

Updating custom fields does not trigger update to "updated_on" field in the customized object (#5354).

Patch by Yuichi HARADA.

History

#1 - 2010-05-13 01:15 - Stuart Cianos

- File redmine_patch_5354.diff added

Patch which corrects the behavior is attached.

#2 - 2010-08-08 17:28 - Jean-Baptiste Barth

Sorry I don't see the point : I can confirm updated_on field is not updated when you just change a custom field, but why do you need it to be updated ? I don't see any place where we display or use this information, so it doesn't seem to be a problem to me. Can you explain your use-case please ?

#3 - 2010-08-09 18:46 - Stuart Cianos

The field is used by plugins and reports at our site to determine the last date of activity on the project's general information. It could potentially be used by other plugins or reporting tools as well.

Not updating the field is not the expected behavior to end users, and would be considered a bug if a user is expecting the updated_on value to actually reflect the last date of update for the project's general information (whether native or a custom field).

- Stu

Jean-Baptiste Barth wrote:

Sorry I don't see the point : I can confirm updated_on field is not updated when you just change a custom field, but why do you need it to be updated ? I don't see any place where we display or use this information, so it doesn't seem to be a problem to me. Can you explain your use-case please ?

#4 - 2010-09-03 19:30 - Stuart Cianos

- File redmine_patch_5354_r2.diff added

New patch revision against Redmine 1.0.1

#5 - 2010-09-15 23:22 - Stuart Cianos

- Assignee set to Eric Davis

#6 - 2010-09-15 23:32 - Holger Just

- Assignee deleted (Eric Davis)

Please don't assign issues to people without them telling you to do so explicitly.

Also, your patch has some issues.

- You save projects two times and don't check the second time.
- Patches always have to be against trunk to be accepted.
- You are missing some tests.

#7 - 2010-09-16 00:22 - Stuart Cianos

Bullet points 1 and 2 fixed:

```
# Force a save of the project record with the correct updated_on time
# to cover corner cases of custom fields, etc. being updated by user
# which do not natively trigger the base project save method
@project.updated_on = Time.now
if validate_parent_id && @project.save
  @project.set_allowed_parent!(params[:project]['parent_id']) if params[:project].has_key?('parent_id')
  flash[:notice] = l(:notice_successful_update)
```

Before submitting a new patch against Trunk, what specific tests need to be wrapped around setting project.updated_on in order to get this accepted upstream?

#8 - 2010-09-16 03:08 - Eric Davis

Thanks for the patch. You shouldn't set updated_on yourself though. Rails provides a :touch option that will automatically update updated_on when any custom field changes.

<http://github.com/rails/rails/commit/abb899c54e8777428b7a607774370ba29a5573bd>

Also, your proposed patch would work for editing a project through the web only. If a project is updated any other way, it wouldn't be updated.

#9 - 2010-09-17 02:45 - Stuart Cianos

- File scvmc_project_cf_updated_on_r3.diff added

New patch which handles the update within the custom_value model is attached. Note that I attempted to use the :touch property within the belongs_to in custom_value, but ran into what may be a bug or undefined behavior for polymorphic fields in ActiveRecord (though did not have time to try and debug activerecord itself). I found a way around it by using a callback and then checking if the parent relation was a project.

If there is a better methodology to address this within Redmine's model, please do suggest so that it can be implemented and submitted for inclusion.

#10 - 2010-09-18 05:03 - Stuart Cianos

Quick question: Does the development team prefer that a patch issue be opened up w/ the diff for the refactored patch, or just keep the fix attached to this defect?

#11 - 2011-03-24 08:53 - Toshi MARUYAMA

- Category set to Custom fields

#12 - 2020-01-13 19:22 - shawn freeman

Bump - I have this problem as well.

Environment:
Redmine version 4.0.2.stable
Ruby version 2.4.5-p335 (2018-10-18) [x86_64-linux]
Rails version 5.2.2
Environment production
Database adapter MySQL2
Mailer queue ActiveJob::QueueAdapters::AsyncAdapter
Mailer delivery sendmail
SCM:

Subversion 1.7.14
Git 2.19.1
Filesystem
Redmine plugins:
redmine_agile 1.4.11
redmine_checklists 3.1.16
redmine_default_custom_query 1.4.0
redmine_image_clipboard_paste 3.3.0
redmineup_tags 2.0.4

#13 - 2020-01-20 07:01 - Yuichi HARADA

- File 5354_fixed.patch added

Update updated_on of the class (e.g. Project) that uses 'acts_as_customizable' with the following patch.

```
diff --git a/lib/plugins/acts_as_customizable/lib/acts_as_customizable.rb b/lib/plugins/acts_as_customizable/lib/acts_as_customizable.rb
index 96fdbf8b6..3703a5e65 100644
--- a/lib/plugins/acts_as_customizable/lib/acts_as_customizable.rb
+++ b/lib/plugins/acts_as_customizable/lib/acts_as_customizable.rb
@@ -145,7 +145,9 @@ module Redmine
   end
   end
   self.custom_values = target_custom_values
+  custom_values_changed = custom_values.any?(&:changed?)
   custom_values.each(&:save)
+  touch if persisted? && custom_values_changed
   @custom_field_values_changed = false
   true
 end
```

#14 - 2020-01-22 11:08 - Go MAEDA

- Target version set to Candidate for next major release

#15 - 2020-01-30 08:47 - Go MAEDA

- Target version changed from Candidate for next major release to 4.2.0

Setting the target version to 4.2.0.

#16 - 2020-02-04 06:19 - Go MAEDA

- Subject changed from Updating custom fields does not trigger update to "updated_on" field in the Project model to Updating custom fields does not trigger update to "updated_on" field in the customized object

#17 - 2020-02-11 05:06 - Go MAEDA

- Status changed from New to Closed
- Assignee set to Go MAEDA
- Resolution set to Fixed

Committed the patch. Thank you.

#18 - 2020-02-11 05:56 - Go MAEDA

- Status changed from Closed to Reopened

The patch breaks an existing test.

Failure:

```
IssueTest#test_closed_on_should_be_set_when_closing_an_open_issue [/var/lib/jenkins/workspace/trunk/DATABASE_ADAPTER/postgresql/RUBY_VER/ruby-2.6/test/unit/issue_test.rb:3071]:
No visible difference in the ActiveSupport::TimeWithZone#inspect output.
You should look at the implementation of #== on ActiveSupport::TimeWithZone or its members.
Tue, 11 Feb 2020 04:40:14 UTC +00:00
```

```
bin/rails test test/unit/issue_test.rb:3064
```

#19 - 2020-02-12 04:05 - Yuichi HARADA

- File 5354_fixed_v2.patch added

Go MAEDA wrote:

The patch breaks an existing test.

[...]

The test passes with the following patch. The saved_changes? method is available in Rails 5.1.0 and later.

https://github.com/rails/rails/blob/v5.1.0/activerecord/lib/active_record/attribute_methods/dirty.rb#L170

<https://github.com/rails/rails/pull/25337>

```
diff --git a/lib/plugins/acts_as_customizable/lib/acts_as_customizable.rb b/lib/plugins/acts_as_customizable/lib/acts_as_customizable.rb
index 3703a5e65..73b2ae847 100644
--- a/lib/plugins/acts_as_customizable/lib/acts_as_customizable.rb
+++ b/lib/plugins/acts_as_customizable/lib/acts_as_customizable.rb
@@ -145,9 +145,8 @@ module Redmine
   end
   end
   self.custom_values = target_custom_values
-  custom_values_changed = custom_values.any?(&:changed?)
  custom_values.each(&:save)
-  touch if persisted? && custom_values_changed
+  touch if !saved_changes? && custom_values.any?(&:saved_changes?)
  @custom_field_values_changed = false
  true
end
```

#20 - 2020-02-29 07:42 - Go MAEDA

- Status changed from Reopened to Closed

Committed the new patch. Thank you.

Files

redmine_patch_5354.diff	1.09 KB	2010-05-12	Stuart Cianos
redmine_patch_5354_r2.diff	1.22 KB	2010-09-03	Stuart Cianos
scvmc_project_cf_updated_on_r3.diff	1017 Bytes	2010-09-17	Stuart Cianos
5354_fixed.patch	1.89 KB	2020-01-20	Yuichi HARADA
5354_fixed_v2.patch	1.83 KB	2020-02-12	Yuichi HARADA