

Redmine - Defect #4353

Changing Tracker toggles presence of custom field

2009-12-07 09:27 - Felix Schäfer

Status:	Closed	Start date:	2009-12-07
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Custom fields	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:	Fixed		
Description			
(I haven't tested this one on trunk yet, but the problem is reproducible on a r3110 instance, and I don't think there's been much change to issues or custom fields since then. I'll have a look at it later on my dev install.)			
We have a custom field (a list) defined for all three default trackers (bug, feature, support), and changing from one tracker to another, no matter from which to which other, toggles the presence of the custom field.			

Associated revisions

Revision 3139 - 2009-12-09 10:12 - Jean-Philippe Lang

Fixes tracker_id and custom_field_values assignment order for issues (#4353).

History

#1 - 2009-12-07 11:16 - Felix Schäfer

So I tested this on my devel installation, running current git (commit 84bf891bb5bc71edfed3f5da75f4af98ddfc1a41), which is [r3129](#). To pin it down some more: this happens only on the form for new issues, changing the tracker on an existing issue doesn't exhibit this behavior.

I have a work-around for this one, in that you have to delete the custom_field from the params[:issue] before passing them to the newly instantiated issue, which works when adding @params[:issue].delete(:custom_field_values) in the first branch of the if-statement after [source:trunk/app/controllers/issues_controller.rb#L441](#), but I think the problem rather lies in the Issue model.

#2 - 2009-12-07 11:49 - Felix Schäfer

So I think the ultimate culprit is Redmine::Acts::Customizable::InstanceMethods#custom_field_values ([source:trunk/vendor/plugins/acts_as_customizable/lib/acts_as_customizable.rb#L61](#)), because it will return @custom_field_values which seems to get initialized by @issue.attributes = params[:issue] in [source:trunk/app/controllers/issues_controller.rb#L445](#), which for whatever reason seems to set @custom_field_values directly rather than through #custom_field_values= (or #save_custom_field_values gets called at some point, but that would be very weird).

Anyway, I removed the modification mentioned in the previous post from my issues_controller.rb, and got it working by removing @custom_field_values ||= from [source:trunk/vendor/plugins/acts_as_customizable/lib/acts_as_customizable.rb#L62](#), which kills the caching, but gets me the custom field every time. I can see different solutions for this (attr_protected, cleaning params[:issue] as in the previous post, killing the caching as mentioned above ...), but I'm not sure which one is preferred or would be considered "clean".

#3 - 2009-12-07 12:26 - Felix Schäfer

In my first response, Textile mangled the text in ways it wasn't supposed to, which leaves an @ hanging in front of params. The whole sentence should read:

I have a work-around for this one, in that you have to delete the custom_field from the params[:issue] before passing them to the newly instantiated @issue, which works when adding params[:issue].delete(:custom_field_values) in the first branch of the if-statement after [source:trunk/app/controllers/issues_controller.rb#L441](#), but I think the problem rather lies in the Issue model.

#4 - 2009-12-08 00:25 - Mischa The Evil

- Status changed from New to Resolved
- Affected version (unused) set to devel
- Resolution set to Fixed

Felix,

It seems this is fixed already in [r3125](#) for issue [#4345](#). Can you test on a up-to-date trunk and report back?

#5 - 2009-12-08 07:06 - Felix Schäfer

Mischa The Evil wrote:

Felix,

It seems this is fixed already in [r3125](#) for issue [#4345](#). Can you test on a up-to-date trunk and report back?

Well, all my testing and proposed solutions are for [r3129](#) (see my first answer), so no, it wasn't fixed in [r3125](#).

#6 - 2009-12-08 14:16 - Jean-Philippe Lang

- Status changed from Resolved to New

Felix, I can't reproduce.

Can you give some more details: ruby and rails versions and where it happens (new or edit issue form)?

#7 - 2009-12-08 15:30 - Felix Schäfer

I can reproduce this consistently on both these installations (both are [r3129](#) or git equivalent):

```
felix@arael:~/Code/redmine [master]$ ruby script/about
About your application's environment
Ruby version      1.8.7 (universal-darwin10.0)
RubyGems version  1.3.5
Rack version      1.0
Rails version     2.3.4
Active Record version 2.3.4
Active Resource version 2.3.4
Action Mailer version 2.3.4
Active Support version 2.3.4
Application root  /Users/felix/Code/redmine
Environment       development
Database adapter  sqlite3
Database schema version 20091205124427
```

```
About your Redmine plugins
Redmine Doodles plugin      0.1
Redmine Newissuealerts plugin 0.0.2
```

```
# RAILS_ENV=production ruby script/about
About your application's environment
Ruby version      1.8.7 (x86_64-linux)
RubyGems version  1.3.5
Rack version      1.0
Rails version     2.3.4
Active Record version 2.3.4
Active Resource version 2.3.4
Action Mailer version 2.3.4
Active Support version 2.3.4
Application root  /xxx
Environment       production
Database adapter  mysql
Database schema version 20091205124427
```

```
About your Redmine plugins
Issue Due Date  0.1.0
```

Steps to reproduce:

1. Create a custom field for issues, in my example it is named "Client" and is a list for all trackers, used as a filter and searchable,
2. Activate the custom field in the project,
3. Go to the new issue page, and change the tracker (for a new ticket, before creating the ticket!), and watch the custom field appearing and disappearing.

If you wish, I can also send you my dev sqlite, but I don't think it's related to that. Looking at the development.log, I can also see that custom fields are looked up on loading the "New Issue" page, but not after changing the tracker once, again after changing it twice, and so on.

#8 - 2009-12-08 21:54 - Jean-Philippe Lang

Are you able to try this patch ? It needs a restart.

Index: app/models/issue.rb

```
=====
--- app/models/issue.rb      (revision 3123)
+++ app/models/issue.rb      (working copy)
@@ -141,9 +141,22 @@

    def tracker_id=(tid)
      self.tracker = nil
-     write_attribute(:tracker_id, tid)
+     result = write_attribute(:tracker_id, tid)
+     @custom_field_values = nil
+     result
    end

+   # Overrides attributes= so that tracker_id gets assigned first
+   def attributes_with_tracker_first=(new_attributes, *args)
+     return if new_attributes.nil?
+     new_tracker_id = new_attributes['tracker_id'] || new_attributes[:tracker_id]
+     if new_tracker_id
+       self.tracker_id = new_tracker_id
+     end
+     self.attributes_without_tracker_first = new_attributes, *args
+   end
+   alias_method_chain :attributes=, :tracker_first
+
+   def estimated_hours=(h)
+     write_attribute :estimated_hours, (h.is_a?(String) ? h.to_hours : h)
+   end
end
```

#9 - 2009-12-08 23:06 - Felix Schäfer

I tried that on my devel install, and everything works as expected. From what I can tell, your code should thwart the problem, so that works for me.

#10 - 2009-12-08 23:24 - Jean-Philippe Lang

Thanks for the feedback. I'll commit soon.

#11 - 2009-12-08 23:45 - Mischa The Evil

- Resolution deleted (Fixed)

Felix Schäfer wrote:

Mischa The Evil wrote:

It seems this is fixed already in [r3125](#) for issue [#4345](#). Can you test on a up-to-date trunk and report back?

Well, all my testing and proposed solutions are for [r3129](#) (see my first answer), so no, it wasn't fixed in [r3125](#).

Sorry, I indeed have missed the "version-update" in [#note-1](#)... Was under the assumption that you were still testing with [source:/trunk@3110](#).

Jean-Philippe Lang wrote:

Felix, I can't reproduce.

I am not able to reproduce the error neither.

Jean-Philippe Lang wrote:

Can you give some more details: ruby and rails versions and where it happens (new or edit issue form)?

- Output of `./script/about`:

```
[mischa@server xxx]# ./script/about
About your application's environment
Ruby version      1.8.6 (i686-linux)
RubyGems version  1.3.5
Rack version      1.0
Rails version     2.3.4
Active Record version 2.3.4
Active Resource version 2.3.4
```

```
Action Mailer version    2.3.4
Active Support version   2.3.4
Application root         /xxx
Environment              production
Database adapter         mysql
Database schema version  20091205124427
```

- MySQL 4.1.20
- Not able to reproduce the issue following the steps to reproduce given by Felix in [#note-7](#). Not on the new issue form neither on the edit issue form.

#12 - 2009-12-08 23:59 - Mischa The Evil

Jean-Philippe Lang wrote:

Are you able to try this patch ? It needs a restart.
[...]

Tested this on [source/trunk@3137](#) too and the feature keeps working as expected without any side-effects. No changes in functionality for me here. Seems it might (have) be (been) a Ruby 1.8.7 issue only?

[...] I'll commit soon.

Thanks for your quick heads-up again ;)

#13 - 2009-12-09 10:08 - Jean-Philippe Lang

- *Status changed from New to Closed*

- *Resolution set to Fixed*

Fix committed in [r3139](#).

Mischa, thanks for investigating on this but I've also tested with ruby 1.8.7 and couldn't reproduce.

Anyway, this had to be fixed because we should not rely on params order for issue attributes assignment.