

Redmine - Defect #25726

Issue details page shows default values for custom fields that aren't actually set

2017-04-28 17:13 - Jens Krämer

Status:	Closed	Start date:	
Priority:	High	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Custom fields	Estimated time:	0.00 hour
Target version:	4.0.0	Affected version:	3.2.0
Resolution:	Fixed		
Description			
<p>I posted this already as a comment in #21074 as this is where this behavior was most certainly introduced but it's probably a good idea to open a new issue for this.</p>			
The problem			
<p>When creating an issue custom field with a default value, the issue details page of issues that have existed before shows the field with this default value. However since the value is not really set in the DB you cannot see it in the issues list, nor can you find any of these issues when searching for it through the filters.</p>			
Steps to reproduce			
<ul style="list-style-type: none">• Create a new issue.• Create a new issue custom field (I tried with boolean and text types) with a default value• look at the previously created issue in issues list (have the custom field displayed) and detail view• note the different values being shown for the custom field - issue details page shows the configured default, while the issue list shows the field as unset (empty).• When filtering the issue list for the field's value (or simply looking at the database) it turns out that the field is indeed unset (neither filtering for the field being 'yes' or 'no' turns up the issue). So clearly the details view is lying about the field's true value by replacing <i>unset</i> with the configured default.			
Desired Behavior			
<p>Imho the default value for unset custom fields (where no corresponding custom_value exists) should only be preset in the edit form, but not when merely showing the issue details. Under all circumstances the details view should be consistent with the issues list in this regard.</p> <p>I came up with a patch that checks if there is actually any custom value present in the database for the given field before calling show_value, but it feels more like a workaround than a solution.</p> <p>Another related thing I observed is that, when editing such an issue and setting the custom field's value for the first time and to something other than the default, a journal detail is created saying "field xy was changed from <default> to <new value>", though the default was never really set on the field.</p> <p>Finally, it might be a useful feature to offer the option of applying the default to all issues (matching tracker / project selection) when saving a custom field with a default set.</p>			
Related issues:			
Related to Redmine - Defect #9849: Custom field values of several issue custo...			New
Related to Redmine - Defect #21074: When changing the tracker of an existing ...			Closed

Associated revisions

Revision 17499 - 2018-09-23 10:25 - Jean-Philippe Lang

Fixed that issue details page shows default values for custom fields that aren't actually set (#25726).

History

#1 - 2017-05-25 13:14 - Toshi MARUYAMA

- Related to Defect #9849: Custom field values of several issue custom field formats don't get set/updated properly during email issue creation/reply added

#2 - 2017-06-02 12:14 - Mischa The Evil

- Related to Defect #21074: When changing the tracker of an existing issue, new custom fields are not initialized with their default value added

#3 - 2017-06-02 12:14 - Mischa The Evil

- Status changed from New to Confirmed
- Priority changed from Normal to High
- Target version set to 3.2.7
- Affected version changed from 3.3.3 to 3.2.0

I'll give my feedback below each of the relevant, quoted parts:

Jens Krämer wrote:

I posted this already as a comment in [#21074](#) as this is where this behavior was most certainly introduced but it's probably a good idea to open a new issue for this.

Thanks for this...

The problem

When creating an issue custom field with a default value, the issue details page of issues that have existed before shows the field with this default value. However since the value is not really set in the DB you cannot see it in the issues list, nor can you find any of these issues when searching for it through the filters.

I confirmed this behaviour on <source/trunk@16580>.

[...] So clearly the details view is lying about the field's true value by replacing *unset* with the configured default.

Indeed, and I don't like this either.

Desired Behavior

Imho the default value for unset custom fields (where no corresponding custom_value exists) should only be preset in the edit form, but not when merely showing the issue details. Under all circumstances the details view should be consistent with the issues list in this regard.

Agreed.

I came up with a patch that checks if there is actually any custom value present in the database for the given field before calling `show_value`, but it feels more like a workaround than a solution.

The patch does the job, though it's looking like a workaround indeed.

Another related thing I observed is that, when editing such an issue and setting the custom field's value for the first time and to something other than the default, a journal detail is created saying "field xy was changed from <default> to <new value>", though the default was never really set on the field.

IMHO this too shouldn't be happening either. It should reflect from <unset> to <new value> instead.

Finally, it might be a useful feature to offer the option of applying the default to all issues (matching tracker / project selection) when saving a custom field with a default set.

This might indeed be a useful addition. It seems out-of-scope for this issue though.

I'll target the issue to [3.2.7](#) and set the affected version to [3.2.0](#) (where this defect got introduced). I'll also set status and priority accordingly.

[Jens Priesen](#): thanks for digging into this...

#4 - 2017-06-02 13:01 - Mischa The Evil

I just noticed that the impact of this might be worse and more pervasive than earlier expected. I'll elaborate based on TS's STR:

- Create a new issues tree like:

```
issue1
issue2
-> issue3
-> issue4
-> issue5
    -> issue6
```

- Create a new issue custom field (I tried with boolean type) with a default value
- ...several steps that are the same as in the initial description and which are not relevant here...
- Edit issue6, setting the custom field's value for the first time and to something other than the default

Observed behaviour

The custom value is saved correctly for issue6 (and the wrong journal is created as discussed earlier). But besides this, also issue5 and issue2 (the propagated parents of issue6) get the default value of the custom field set (without creating respective journals and without updating their 'updated' timestamps).

Desired Behavior

The default value for unset custom fields (where no corresponding custom_value exists) should not be set — silently — for parent issues (or any other issue) when updating a child issue (with an initial custom field value other than the default value).

A correction of this specific behavior is obviously not included in the proposed patch. ~~This whole specific issue reminds me of issue #25806. I'll dig into that one to see if it's related in a way~~ (see edit note).

Edit by Mischa The Evil: after triaging #25806, I've come to the conclusion that, while it is a valid defect, it does not seem to be related to this issue. For more details see #25806#note-1.

#5 - 2017-06-06 23:50 - Jean-Philippe Lang

Imho the default value for unset custom fields (where no corresponding custom_value exists) should only be preset in the edit form, but not when merely showing the issue details. Under all circumstances the details view should be consistent with the issues list in this regard.

I agree that the details view should be consistent with the issue list. On the other hand, keeping the default value in the edit form for existing issues is more debatable as it raises the same issue (form content not consistent with the details view) and would cause an unexpected behaviour: a user who just enters a note will also generate a journal saying "custom field set to X".

Here is a simple fix for this that should also address the problem reported by Mischa for the parent issues (not tested).

```
Index: app/models/custom_value.rb
-----
--- app/models/custom_value.rb      (revision 16595)
+++ app/models/custom_value.rb      (working copy)
@@ -24,7 +24,7 @@
 
 def initialize(attributes=nil, *args)
   super
-   if new_record? && custom_field && !attributes.key?(:value)
+   if new_record? && (customized.nil? || customized.new_record?) && custom_field && !attributes.key?(:value)
     self.value ||= custom_field.default_value
   end
 end
```

We should probably apply this fix and give the option of applying the default to all issues as proposed by Jens.

#6 - 2017-06-07 02:27 - Jens Krämer

Looks like the best way to solve this.

#7 - 2017-06-07 21:38 - Jean-Philippe Lang

- Target version changed from 3.2.7 to 4.1.0

#8 - 2017-06-09 05:57 - Mischa The Evil

Jean-Philippe Lang wrote:

Imho the default value for unset custom fields (where no corresponding custom_value exists) should only be preset in the edit form, but not when merely showing the issue details. Under all circumstances the details view should be consistent with the issues list in this regard.

I agree that the details view should be consistent with the issue list. On the other hand, keeping the default value in the edit form for existing issues is more debatable as it raises the same issue (form content not consistent with the details view) and would cause an unexpected behaviour: a user who just enters a note will also generate a journal saying "custom field set to X".

Good point. I had read this expectation as if it was about the new form, my mistake. So the conclusion of this could be that it's not viable to have [#21074](#) fixed without side-effects mentioned in that issue and/or the ones described in this issue.

Jean-Philippe Lang wrote:

Here is a simple fix for this that should also address the problem reported by Mischa for the parent issues (not tested).

Confirmed that this patch fixes both problems. It is however (also) reverting the solution ([r14773](#)) which was added to fix the initial [#21074](#). This is confirmed by the failing test:

```
1) Failure:
Redmine::ApiTest::IssuesTest#test_PUT_/issues/:id.json_with_tracker_change_and_omitted_custom_field_specific_t
o_that_tracker_should_set_default_value [test/integration/api_test/issues_test.rb:752]:
Expected: "Default"
Actual: nil
```

Jean-Philippe Lang wrote:

We should probably apply this fix and give the option of applying the default to all issues as proposed by Jens.

[...]

- **Target version** changed from 3.2.7 to 3.5.0

Does this scheduling also cover delivery of an intermediate fix (like your proposed patch)? I'm asking because IMHO, the side-effects of the changes fixing [#21074](#) are more disruptive than the initial problem they are supposed to fix (which already leads to users patching their instances/data: eg. [Migration 3.0.3 -> 3.3.3 : Custom Fields and default values](#)). Based on that, I would suggest to include the proposed (albeit [#21074](#)-reintroducing) patch in the next 3.2 and 3.3 maintenance releases and in the upcoming 3.4 stable release line until we have a solution (that removes the root cause of [#21074](#) altogether) of handling custom field default values, like as proposed by Jens, that can be included in a future 3.5 feature release. What do you think?

#9 - 2018-06-30 03:39 - Go MAEDA

- Assignee set to Jean-Philippe Lang
- Target version changed from 4.1.0 to 4.0.0

#10 - 2018-09-23 10:27 - Jean-Philippe Lang

- Status changed from Confirmed to Closed
- Resolution set to Fixed

I've committed a fix in [r17499](#) that does not revert the fix for [#21074](#). Maybe it should be merged in 3.x branches, let me know if you think so.

Files

0001-do-not-display-default-values-that-aren-t-actually-s.patch	1.86 KB	2017-04-28	Jens Krämer
---	---------	------------	-------------