

## Redmine - Feature #2503

### History of Ticket-Description

2009-01-13 22:28 - Thomas Rekitke

<b>Status:</b>	Closed	<b>Start date:</b>	2009-01-13
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Issues	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Resolution:</b>			

#### Description

It is a pity that changes of the description of an issue are nowhere stored. So you can never be sure, the description is what it was or what it should be :-)

I built a workaround for myself, by copying the description to a shallow wiki-page, where I can see the history. Obviously more a bad hack, than a good solution. I paste the patch anyhow to illustrate what could be a solution. Using the wiki as "storage" for the history is maybe not such a bad idea after all.

Index: /usr/local/workspace/redmine/app/models/issue.rb

```
-----
--- /usr/local/workspace/redmine/app/models/issue.rb      (revision 2263)
+++ /usr/local/workspace/redmine/app/models/issue.rb      (working copy)
@@ -152,7 +152,21 @@
                                     :prop_key => c.custom_field_id,
                                     :old_value => @custom_values_before_change[
c.custom_field_id],
                                     :value => c.value)
-   }
+   }
+   # edit corresponding wiki page if of type use case
+   if self.tracker_id == 6 and @issue_before_change.description != self.description # only tr
acker 6. hard coded magig number, brrr :-(
+     wiki_page = self.project.wiki.find_or_new_page( subject )
+     wiki_page.content = WikiContent.new(:page => wiki_page) if wiki_page.new_record?
+     content = wiki_page.content_for_version( nil )
+     content.comments = nil
+     content.text = self.description
+     content.author = User.current
+     wiki_page.new_record? ? wiki_page.save : content.save
+   end
+   @current_journal.save
end
# Save the issue even if the journal is not saved (because empty)
```

#### Related issues:

Related to Redmine - Patch #1717: Show diff for issue description change	<b>Closed</b>	<b>2008-07-30</b>
Has duplicate Redmine - Defect #3612: Changing original content of the issue ...	<b>Closed</b>	<b>2009-07-13</b>
Is duplicate of Redmine - Feature #746: Versioned issue descriptions	<b>Closed</b>	<b>2008-02-27</b>

#### History

##### #1 - 2009-01-13 22:54 - Mischa The Evil

See patch [#1717](#) which holds a working patch to adress this issue. It also provides an idea of implementing it in the core.

I'll relate the issues...

##### #2 - 2009-01-14 11:46 - Thomas Rekitke

- Status changed from New to Resolved

##### #3 - 2009-04-22 10:44 - Robert Pollak

In which Redmine release is or will this be contained?

#### #4 - 2009-07-13 22:42 - Mischa The Evil

- Status changed from Resolved to New

Wrapped the code into pre-tags and corrected the issue-status.

#### #5 - 2009-07-17 13:25 - Martin Schneider

Thomas Rekitke wrote:

It is a pity that changes of the description of an issue are nowhere stored. So you can never be sure, the description is what it was or what it should be :-)

Independent from the implementation, this is a great feature! Please make this feature part of the core. Additionally it would be nice if the comments would have a history too. So the "can edit comments" rule gets a new aspect.

BTW:

Thanks for Redmine! It is pretty cool!

#### #6 - 2009-08-03 11:25 - Marek Kreft

Hi,

When can we expect this feature (or its alternative) to be implemented in Redmine core?

#### #7 - 2009-08-03 14:28 - Marek Kreft

Hello again.

This is what I figured out today. I think that this small fix improves things quite well.

In place of

```
if self.tracker_id == 6 and @issue_before_change.description != self.description # only tracker 6. hard code
d magig number, brrr :-(
+   wiki_page = self.project.wiki.find_or_new_page( subject )
+   wiki_page.content = WikiContent.new(:page => wiki_page) if wiki_page.new_record?
+   content = wiki_page.content_for_version( nil )
+   content.comments = nil
+   content.text = self.description
+   content.author = User.current
+   wiki_page.new_record? ? wiki_page.save : content.save
+ end
```

I've added this

```
if @issue_before_change.description != self.description
  @current_journal.details << JournalDetail.new(:property => 'cf',
                                                :prop_key => 'description',
                                                :old_value => @issue_before_change.description,
                                                :value => self.description)
```

It was tested on trunk version and works like charm.

#### #8 - 2010-05-16 01:18 - Felix Schäfer

- Status changed from New to Closed

Closing this issue in favor of [#746](#).