redmineorg-copy202205 - Vote #69784

notes field is not propagated during issue copy

2022/05/09 15:33 - Admin Redmine

ステータス:	Closed	**開始日:**	2011/04/27
優先度:	通常	**期日:**	
担当者:		**進捗率:**	0%
カテゴリ:	Issues_2	**予定工数:**	0.00時間
対象バージョン:	1.2.0_27	**作業時間:**	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/8239	**status_id:**	5
category_id:	2	**tracker_id:**	1
version_id:	27	**plus1:**	0
issue_org_id:	8239	**affected_version:**	
author_id:	32759	**closed_on:**	
assigned_to_id:	0	**affected_version_id:**	31
comments:	7		

説明

scenario (vanilla installation of redmine):

- open existing issue
- click copy
- add notes
- click copy button

notes are not visible, they are lost

journals

Confirmed with demo site, I'll try to investigate further.

Had a quick look yesterday evening, @Issue#new.copy_from@ does not copy current journal.

Maybe that call to @Issue#init_journal@ in @IssueMovesController@ should be moved to model after call to @#copy_from@ and only for a move action (do we really need a journal entry in a copy context if no notes are filled in) ?

This change prevents double @#save@ :

```
Index: app/models/issue.rb
===================================================================
--- app/models/issue.rb (revision 5543)
+++ app/models/issue.rb (working copy)
@@ -147,10 +147,18 @@
             end ||= false
         end

-    def move_to_project_without_transaction(new_project, new_tracker = nil, options = {})
+    def move_to_project_without_transaction(new_project, notes = nil, new_tracker = nil, options = {})
             options ||= {}
-        issue = options[:copy] ? self.class.new.copy_from(self) : self
-
+
+        if options[:copy]
+            issue = self.class.new.copy_from(self)
+            issue.instance_variable_set :@current_journal, Journal.new(:journalized => issue, :user => User.current, :notes =>
otes.present?
+        else
+            issue = self
+            issue.init_journal(User.current)
```

```
+               issue.current_journal.notes  =  notes  if  notes.present?
+           end
+
            if  new_project  &&  issue.project_id  !=  new_project.id
                #  delete  issue  relations
                unless  Setting.cross_project_issue_relations?
@@  -867,7  +875,8  @@
                                                                            :prop_key  =>  c
                                                                            :old_value  =>  (
                                                                            :value  =>  send
-           }
+           }  if  @issue_before_change
+
                #  custom  fields  changes
                custom_values.each  {|c|
                    next  if  (@custom_values_before_change[c.custom_field_id]==c.value  ||
@@  -876,7  +885,8  @@
                                                                            :prop_key  =>  c
                                                                            :old_value  =>  (
                                                                            :value  =>  c.val
-           }
+           }  if  @custom_values_before_change
+
                @current_journal.save
                #  reset  current  journal
                init_journal  @current_journal.user,  @current_journal.notes
Index:  app/controllers/issue_moves_controller.rb
===================================================================
---  app/controllers/issue_moves_controller.rb       (revision  5543)
+++  app/controllers/issue_moves_controller.rb       (working  copy)
@@  -17,10  +17,8  @@
                moved_issues  =  []
                @issues.each  do  |issue|
                    issue.reload
-                   issue.init_journal(User.current)
-                   issue.current_journal.notes  =  @notes  if  @notes.present?
                    call_hook(:controller_issues_move_before_save,  {  :params  =>  params,  :issue  =>  issue,  :target_project  =>  @
copy  =>  !!@copy  })
-                   if  r  =  issue.move_to_project(@target_project,  new_tracker,  {:copy  =>  @copy,  :attributes  =>  extract_changed
move(params)})
+                   if  r  =  issue.move_to_project(@target_project,  @notes,  new_tracker,  {:copy  =>  @copy,  :attributes  =>  extrac
utes_for_move(params)})
                        moved_issues  <<  r
                    else
                        unsaved_issue_ids  <<  issue.id
```

Less intrusive based on trunk ?

This was fixed in r5602. Anything wrong with it?

related_issues

relates,Closed,6901,Copy/Move an issue does not give any history of who actually did the action.

履歴

#1 - 2022/05/10 17:19 - Admin Redmine

- カテゴリ を Issues_2 にセット

- 対象バージョン を 1.2.0_27 にセット