Redmine - Defect #24979

Email "keyword" lines deleted even if the sender doesn't have permission to edit that field

2017-02-02 16:53 - Felix Schäfer

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Email receiving	**Estimated time:**	0.00 hour
Target version:	Candidate for next major release		
Resolution:		**Affected version:**	

Description

When receiving an email with keyword lines (RedmineReceivingEmails) Redmine will destructively remove these lines, even when the sender doesn't have the permission to edit those attributes. The value of the issue's attributes will not be changed as per the field permission (that is correct) but the line with that value will be removed nonetheless (this is surprising).

One step further might even be to only remove lines with keywords if the value is a possible value for the attribute in question, but that's not quite the scope of my current issue.

History

#1 - 2017-02-02 16:59 - Felix Schäfer

Basically this could probably be done by calling issue.safe_attribute? for each attribute before get_keyword-ing it from the mail body. To illustrate this please see the following diff, this is not tested though and is only for illustration purposes!

The issue.safe_attributes = {} call upfront was to ensure that in the case of an email creating a new issue the safe attributes list was set properly. Tests in the console showed that before that call issue.safe_attribute? will return true for every attribute even when initialising an issue with a project and an author, I'm not quite sure if this is by design or a bug though.

```
--- a/app/models/mail_handler.rb
+++ b/app/models/mail_handler.rb
@@ -531,17 +531,18 @@ class MailHandler < ActionMailer::Base

   # Returns a Hash of issue attributes extracted from keywords in the email body
   def issue_attributes_from_keywords(issue)
+    issue.safe_attributes = {}
     attrs = {
-      'tracker_id' => (k = get_keyword(:tracker)) && issue.project.trackers.named(k).first.try(:id),
-      'status_id' =>  (k = get_keyword(:status)) && IssueStatus.named(k).first.try(:id),
-      'priority_id' => (k = get_keyword(:priority)) && IssuePriority.named(k).first.try(:id),
-      'category_id' => (k = get_keyword(:category)) && issue.project.issue_categories.named(k).first.try(:id)
,
-      'assigned_to_id' => (k = get_keyword(:assigned_to)) && find_assignee_from_keyword(k, issue).try(:id),
-      'fixed_version_id' => (k = get_keyword(:fixed_version)) && issue.project.shared_versions.named(k).first
.try(:id),
-      'start_date' => get_keyword(:start_date, :format => '\d{4}-\d{2}-\d{2}'),
-      'due_date' => get_keyword(:due_date, :format => '\d{4}-\d{2}-\d{2}'),
-      'estimated_hours' => get_keyword(:estimated_hours),
-      'done_ratio' => get_keyword(:done_ratio, :format => '(\d|10)?0')
+      'tracker_id' => issue.safe_attribute?('tracker_id') && (k = get_keyword(:tracker)) && issue.project.tra
ckers.named(k).first.try(:id),
+      'status_id' =>  issue.safe_attribute?('status_id') && (k = get_keyword(:status)) && IssueStatus.named(k
).first.try(:id),
+      'priority_id' => issue.safe_attribute?('priority_id') && (k = get_keyword(:priority)) && IssuePriority.
named(k).first.try(:id),
+      'category_id' => issue.safe_attribute?('category_id') && (k = get_keyword(:category)) && issue.project.
issue_categories.named(k).first.try(:id),
+      'assigned_to_id' => issue.safe_attribute?('assigned_to_id') && (k = get_keyword(:assigned_to)) && find_
assignee_from_keyword(k, issue).try(:id),
+      'fixed_version_id' => issue.safe_attribute?('fixed_version_id') && (k = get_keyword(:fixed_version)) &&
 issue.project.shared_versions.named(k).first.try(:id),
+      'start_date' => issue.safe_attribute?('start_date') && get_keyword(:start_date, :format => '\d{4}-\d{2}
-\d{2}'),
+      'due_date' => issue.safe_attribute?('due_date') && get_keyword(:due_date, :format => '\d{4}-\d{2}-\d{2}
'),
+      'estimated_hours' => issue.safe_attribute?('estimated_hours') && get_keyword(:estimated_hours),
+      'done_ratio' => issue.safe_attribute?('done_ratio') && get_keyword(:done_ratio, :format => '(\d|10)?0')
```

```
     }.delete_if {|k, v| v.blank? }

     if issue.new_record? && attrs['tracker_id'].nil?
@@ -553,7 +554,8 @@ class MailHandler < ActionMailer::Base

   # Returns a Hash of issue custom field values extracted from keywords in the email body
   def custom_field_values_from_keywords(customized)
-    customized.custom_field_values.inject({}) do |h, v|
+    return {} unless customized.safe_attribute? 'custom_field_values'
+    customized.editable_custom_field_values.inject({}) do |h, v|
       if keyword = get_keyword(v.custom_field.name)
         h[v.custom_field.id.to_s] = v.custom_field.value_from_keyword(keyword, customized)
       end
```

#2 - 2017-02-03 19:17 - Jean-Philippe Lang

Felix Schäfer wrote:

> The issue.safe_attributes = {} call upfront was to ensure that in the case of an email creating a new issue the safe attributes list was set properly. Tests in the console showed that before that call issue.safe_attribute? will return true for every attribute even when initialising an issue with a project and an author, I'm not quite sure if this is by design or a bug though.

Indeed, issue safe attributes depend on the tracker and #safe_attribute? returns true for pretty much all attributes if the tracker is not set. issue.safe_attributes = {} sets the default tracker but after that, #safe_attribute? will consider this default tracker and not the tracker submitted in the email. So we would have to set the tracker first, then use #safe_attribute? for other attributes.

#3 - 2017-02-13 05:20 - ravy ouerm

- File ist_000000135856 (1).jpg added

- File base_image.jpg added

#4 - 2017-03-04 04:35 - Toshi MARUYAMA

- File deleted (ist_000000135856 (1).jpg)

#5 - 2017-03-04 04:35 - Toshi MARUYAMA

- File deleted (base_image.jpg)

#6 - 2018-08-14 10:35 - Go MAEDA

- Target version set to Candidate for next major release