

ステータス:	Reopen	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Custom fields_14	予定工数:	0.00時間
対象バージョン:		作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/13118	status_id:	8
category_id:	14	tracker_id:	1
version_id:	0	plus1:	0
issue_org_id:	13118	affected_version:	
author_id:	31712	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	5		

**説明**

This means "a\_issue.project.issue\_custom\_fields" and "Project.find\_by\_id(a\_issue.project.id).issue\_custom\_fields" give sometimes different results. This also affect the result of the method "available\_custom\_fields" in issue.rb. This is only sometimes the case - don't no why.

Especially in that case "a\_issue.project.issue\_custom\_fields.count" give also a wrong value (count like expected, but some expected entries are nil)

A simple detection is possible with:

```
(Project.find_by_id(a_issue.project.id).issue_custom_fields - a_issue.project.issue_custom_fields).empty?
```

The same for trackers:

```
(Tracker.find_by_id(a_issue.tracker.id).custom_fields - a_issue.tracker.custom_fields).empty?
```

journals

**Works for me on my development environment. Can you provide a failing test?**

I will try it. But for now I'm using this method for workaround and detection:

```
'# Overrides Redmine::Acts::Customizable::InstanceMethods#available_custom_fields
def available_custom_fields
  if !(Project.find_by_id(self.project.id).issue_custom_fields - self.project.issue_custom_fields).empty?
    debugger
    puts "Found this bug for project: http://www.redmine.org/issues/13118!"
  end
  if !(Tracker.find_by_id(self.tracker.id).custom_fields - self.tracker.custom_fields).empty?
    debugger
    puts "Found this bug for tracker: http://www.redmine.org/issues/13118!"
  end
  #workaround for this bug:
  this_project = Project.find_by_id(self.project.id)
  this_tracker = Tracker.find_by_id(self.tracker.id)
  retval= (this_project && this_tracker) ? (this_project.all_issue_custom_fields & this_tracker.custom_fields.all) : []
  return retval
end
```

**Now I have detected the problem also once for trackers.**

Thomas Kohler wrote:

This means "a\_issue.project.issue\_custom\_fields" and "Project.find\_by\_id(a\_issue.project.id).issue\_custom\_fields" give

sometimes different results.

@a\_issue.project@ and @Project.find\_by\_id(a\_issue.project.id)@ are not the same objects in memory. You can try:

```
a_issue.project.object_id == Project.find_by_id(a_issue.project.id).object_id
```

So if you update an association (eg. add a tracker or a custom field to a project), you may need to @#reload@ your issue in order to see the new association.

I'm closing it, please reopen if you have a valid failing test.

Yes, you are right! I have given that the a\_issue.save do that already, but unfortunately not.

I have finished a failing test now and also a test with your solution. In my opinion a #reload in the method would be a good idea. Or inside the Issue.save? What do you think about?

```
def available_custom_fields
  if !(Project.find_by_id(self.project.id).issue_custom_fields - self.project.issue_custom_fields).empty?
    self.reload
  end
  if !(Tracker.find_by_id(self.tracker.id).custom_fields - self.tracker.custom_fields).empty?
    self.reload
  end
  retval= (project && tracker) ? (project.all_issue_custom_fields & tracker.custom_fields.all) : []
  return retval
end
```

Please check my failing test and post your opinion about "reload in the #available\_custom\_fields or on/after #save of the issue".

---

#### 履歴

#1 - 2022/05/10 17:14 - Admin Redmine

- カテゴリ を Custom fields\_14 にセット