Redmine - Defect #33281

Totals of custom fields may not be sorted as configured

2020-04-08 17:05 - Go MAEDA

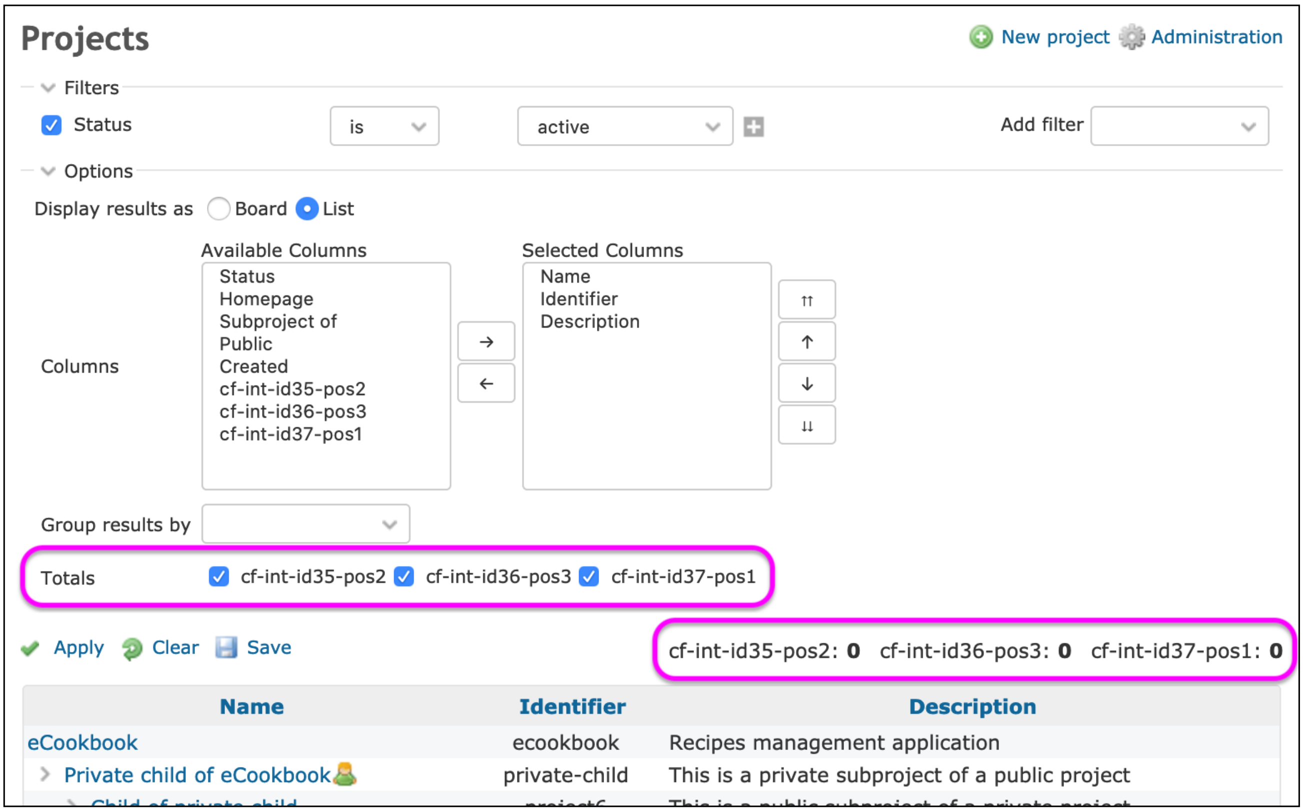
Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Filters	**Estimated time:**	0.00 hour
Target version:	4.1.2		
Resolution:	Fixed	**Affected version:**	4.1.0

Description

Totals for custom fields should be ordered by the position value of the fields, but actually they are ordered by id. They should be ordered according to the position value, like the issues list.

Assume that the positions of the project custom fields are set like this:
custom-fields-position.png

Totals for the custom fields are expected to be displayed in the same order as in the image above, but are actually sorted by id values in the database.



Projects

Filters

☑ Status is active

Options

Display results as ○ Board ● List

Columns

Available Columns: Status, Homepage, Subproject of, Public, Created, cf-int-id35-pos2, cf-int-id36-pos3, cf-int-id37-pos1

Selected Columns: Name, Identifier, Description

Group results by

Totals ☑ cf-int-id35-pos2 ☑ cf-int-id36-pos3 ☑ cf-int-id37-pos1

cf-int-id35-pos2: **0** cf-int-id36-pos3: **0** cf-int-id37-pos1: **0**

Name	Identifier	Description
eCookbook	ecookbook	Recipes management application
Private child of eCookbook	private-child	This is a private subproject of a public project

Associated revisions

Revision 19758 - 2020-05-08 06:30 - Go MAEDA

Fix that totals of custom fields may not be sorted as configured (#33281).

Patch by Yuichi HARADA.

Revision 19759 - 2020-05-08 06:46 - Go MAEDA

Merged r19758 from trunk to 4.1-stable (#33281).

History

#1 - 2020-04-22 07:58 - Yuichi HARADA

- File 33281.patch added

Fixed the Totals for custom fields to be sorted by field position order. I attached a patch.

```
diff --git a/app/models/query.rb b/app/models/query.rb
index 100728cff..7f1169023 100644
--- a/app/models/query.rb
+++ b/app/models/query.rb
@@ -755,7 +755,13 @@ class Query < ActiveRecord::Base
   end

   def available_totalable_columns
-    available_columns.select(&:totalable)
+    available_columns.select(&:totalable).sort do |a, b|
+      if a.is_a?(QueryCustomFieldColumn) && b.is_a?(QueryCustomFieldColumn)
+        a.custom_field <=> b.custom_field
+      else
+        0
+      end
+    end
   end

   def default_columns_names
```

#2 - 2020-04-22 08:28 - Go MAEDA

- Category changed from Projects to Filters

- Target version set to 4.1.2

Setting the target version to 4.1.2.

#3 - 2020-04-24 09:13 - Yuichi HARADA

- File 33281-v2.patch added

Custom fields that are normally stored in "Query#available_columns" are in position order, but the cross-project's projects, issues, and spent time tabs do not store them in "Query#available_columns" in position order.
I fixed the patch as follows.

```
diff --git a/app/models/project_query.rb b/app/models/project_query.rb
index 72a0922bd..b8c79072f 100644
--- a/app/models/project_query.rb
+++ b/app/models/project_query.rb
@@ -69,7 +69,7 @@ class ProjectQuery < Query
   def available_columns
     return @available_columns if @available_columns
     @available_columns = self.class.available_columns.dup
-    @available_columns += ProjectCustomField.visible.
+    @available_columns += project_custom_fields.visible.
                             map {|cf| QueryCustomFieldColumn.new(cf) }
     @available_columns
   end
diff --git a/app/models/query.rb b/app/models/query.rb
index 100728cff..31c5c911b 100644
--- a/app/models/query.rb
+++ b/app/models/query.rb
@@ -609,13 +609,13 @@ class Query < ActiveRecord::Base
     if project
       project.rolled_up_custom_fields
     else
-      IssueCustomField.all
+      IssueCustomField.sorted
     end
   end

   # Returns a scope of project custom fields that are available as columns or filters
   def project_custom_fields
-    ProjectCustomField.all
+    ProjectCustomField.sorted
   end

   # Returns a scope of project statuses that are available as columns or filters
```

```
diff --git a/app/models/time_entry_query.rb b/app/models/time_entry_query.rb
index 610e7628e..39c931d4a 100644
--- a/app/models/time_entry_query.rb
+++ b/app/models/time_entry_query.rb
@@ -113,7 +113,7 @@ class TimeEntryQuery < Query
                              map {|cf| QueryCustomFieldColumn.new(cf) }
     @available_columns += issue_custom_fields.visible.
                              map {|cf| QueryAssociationCustomFieldColumn.new(:issue, cf, :totalable => false)
}
-    @available_columns += ProjectCustomField.visible.
+    @available_columns += project_custom_fields.visible.
                              map {|cf| QueryAssociationCustomFieldColumn.new(:project, cf) }
     @available_columns
   end
```

#4 - 2020-04-28 03:49 - Go MAEDA

Yuichi HARADA wrote:

> Custom fields that are normally stored in "Query#available_columns" are in position order, but the cross-project's projects, issues, and spent time tabs do not store them in "Query#available_columns" in position order.
> I fixed the patch as follows.
> [...]

Thank you for deeply inspecting the issue. But I found that the order of totals on Spent time page still ignores the position value of custom fields even after applying 33281-v2.patch.

Could you look into it?

#5 - 2020-05-07 07:31 - Yuichi HARADA

- File 33281-v3.patch added

Go MAEDA wrote:

> Thank you for deeply inspecting the issue. But I found that the order of totals on Spent time page still ignores the position value of custom fields even after applying 33281-v2.patch.
>
> Could you look into it?

Thank you for pointing out.
I confirmed that TimeEntryCustomFields are not arranged in order of position. It was solved by adding the following patch to 33281-v2.patch.

```
diff --git a/app/models/query.rb b/app/models/query.rb
index 31c5c911b..45e9882a7 100644
--- a/app/models/query.rb
+++ b/app/models/query.rb
@@ -618,6 +618,11 @@ class Query < ActiveRecord::Base
     ProjectCustomField.sorted
   end

+  # Returns a scope of time entry custom fields that are available as columns or filters
+  def time_entry_custom_fields
+    TimeEntryCustomField.sorted
+  end
+
   # Returns a scope of project statuses that are available as columns or filters
   def project_statuses_values
     [
diff --git a/app/models/time_entry_query.rb b/app/models/time_entry_query.rb
index 39c931d4a..c8f8eeaeb 100644
--- a/app/models/time_entry_query.rb
+++ b/app/models/time_entry_query.rb
@@ -100,7 +100,7 @@ class TimeEntryQuery < Query
     add_available_filter "comments", :type => :text
     add_available_filter "hours", :type => :float

-    add_custom_fields_filters(TimeEntryCustomField)
+    add_custom_fields_filters(time_entry_custom_fields)
     add_associations_custom_fields_filters :project
     add_custom_fields_filters(issue_custom_fields, :issue)
     add_associations_custom_fields_filters :user
@@ -109,7 +109,7 @@ class TimeEntryQuery < Query
```

```
   def available_columns
     return @available_columns if @available_columns
     @available_columns = self.class.available_columns.dup
-    @available_columns += TimeEntryCustomField.visible.
+    @available_columns += time_entry_custom_fields.visible.
                             map {|cf| QueryCustomFieldColumn.new(cf) }
     @available_columns += issue_custom_fields.visible.
                             map {|cf| QueryAssociationCustomFieldColumn.new(:issue, cf, :totalable => false)
}
```

#6 - 2020-05-08 06:36 - Go MAEDA

- Subject changed from The order of the totals of project custom fields ignores their position value to Totals of custom fields may not be sorted as configured

#7 - 2020-05-08 06:46 - Go MAEDA

- Status changed from New to Closed

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the patch. Thank you.

Files

custom-fields-position.png	33.7 KB	2020-04-08	Go MAEDA
order-of-totals.png	167 KB	2020-04-08	Go MAEDA
33281.patch	1.71 KB	2020-04-22	Yuichi HARADA
33281-v2.patch	3.94 KB	2020-04-24	Yuichi HARADA
33281-v3.patch	5.29 KB	2020-05-07	Yuichi HARADA