

Redmine - Defect #18018

Project Sorting by Name broken (for large project lists) in v2.x

2014-10-02 20:30 - @ go2null

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Projects	Estimated time:	0.00 hour
Target version:		Affected version:	2.6.1
Resolution:	Wont fix		

Description

Apparently, this is due to the removal, in v2.5, of the patched awesome_nested_set plugin from the Redmine lib/plugins directory.
<http://www.redmine.org/projects/redmine/repository/show/tags/2.4.6/lib/plugins>
<http://www.redmine.org/projects/redmine/repository/show/tags/2.5.0/lib/plugins>
<http://www.redmine.org/projects/redmine/repository/show/branches/2.4-stable/lib/plugins>
<http://www.redmine.org/projects/redmine/repository/show/branches/2.5-stable/lib/plugins>

The patched awesome_nested_set is still available in trunk.
<http://www.redmine.org/projects/redmine/repository/show/trunk/lib/plugins>

Note that this either only affects large project lists (my Redmine has several hundreds) or old Redmine instances that were upgraded from pre-V2.5 and the Project.rebuild_tree"! method fix was not applied.

Workaround ([#6836#note-12](#)) per [Ken Zalewski](#) still works.

```
set_left_and_rights = lambda do |node|
  # set left
  node[left_column_name] = indices[scope.call(node)] += 1
  # find
-  where(["#{quoted_parent_column_full_name} = ? #{scope.call(node)}", node]).order("#{quoted_left_column_full_name}, #{quoted_right_column_full_name}, id").each{|n| set_left_and_rights.call(n) }
+  where(["#{quoted_parent_column_full_name} = ? #{scope.call(node)}", node]).order("#{quoted_left_column_full_name}, #{quoted_right_column_full_name}, name").each{|n| set_left_and_rights.call(n) }
}
  # set right
  node[right_column_name] = indices[scope.call(node)] += 1
  node.save!(:validate => validate_nodes)
end

# Find root node(s)
- root_nodes = where("#{quoted_parent_column_full_name} IS NULL").order("#{quoted_left_column_full_name}, #{quoted_right_column_full_name}, id").each do |root_node|
+ root_nodes = where("#{quoted_parent_column_full_name} IS NULL").order("#{quoted_left_column_full_name}, #{quoted_right_column_full_name}, name").each do |root_node|
  # setup index for this scope
  indices[scope.call(root_node)] ||= 0
  set_left_and_rights.call(root_node)
end
```

Related issues are:

- Feature [#6836](#) Sort project list by Name instead of identifier
- Defect [#12431](#) Project.rebuild! sorts root projects by id instead of name
- Defect [#11508](#) Projects not ordered alphabetically after renaming project

History

#1 - 2014-10-02 20:41 - @ go2null

Forgot to add, after the patch to awesome_nested_set, you need to run the following in the Rails Root directory.

```
bundle exec ruby script/rails runner -e production 'Project.update_all(:lft=>nil,:rgt=>nil); Project.rebuild!'
```

Note that you do not need to run `Project.rebuild_tree!`, just `awesome_nested_set`'s `Project.rebuild!`.

#3 - 2014-10-03 00:20 - Toshi MARUYAMA

@ go2null wrote:

Forgot to add, after the patch to `awesome_nested_set`, you need to run the following in the Rails Root directory.
[...]

Note that you do not need to run `Project.rebuild_tree!`, just `awesome_nested_set`'s `Project.rebuild!`.

You need to run `Project.rebuild_tree!`.

See source and comment.

[source:tags/2.5.2/app/models/project.rb#L417](https://source.rubygems.org/versions/2.5.2/app/models/project.rb#L417)

#4 - 2014-10-03 00:23 - Toshi MARUYAMA

- Status changed from New to Needs feedback

#5 - 2014-10-03 00:34 - Toshi MARUYAMA

@ go2null wrote:

Workaround ([#6836#note-12](#)) per [Ken Zalewski](#) still works.
[...]

It is not work because Redmine build tree after `_save` .

[source:tags/2.5.2/app/models/project.rb#L82](https://source.rubygems.org/versions/2.5.2/app/models/project.rb#L82)

#6 - 2014-10-03 00:42 - Toshi MARUYAMA

@ go2null wrote:

Apparently, this is due to the removal, in v2.5, of the patched `awesome_nested_set` plugin from the Redmine lib/plugins directory.

<http://www.redmine.org/projects/redmine/repository/show/tags/2.4.6/lib/plugins>

<http://www.redmine.org/projects/redmine/repository/show/tags/2.5.0/lib/plugins>

<http://www.redmine.org/projects/redmine/repository/show/branches/2.4-stable/lib/plugins>

<http://www.redmine.org/projects/redmine/repository/show/branches/2.5-stable/lib/plugins>

The patched `awesome_nested_set` is still available in trunk.

<http://www.redmine.org/projects/redmine/repository/show/trunk/lib/plugins>

It is because `awesome_nested_set` after 2.1.6 refactoring is broken.

#7 - 2014-10-03 01:44 - @ go2null

I tried both `rebuild!` and `rebuild_tree!`, but without the patched `rebuild!` in `awesome_nested_set` it did not work.

The issue is not with `rebuild_tree!`. In fact, based on some debug puts that I added, the issue is actually that the `move_to_left_of(to_be_inserted_before)` in `set_or_update_position_under` does not properly update the `lft` and `rgt`. (The incorrect sort is consistent and reproducible every time.)

(I'm mobile so can't attach debug log nor be as descriptive as I'd like.)

#8 - 2014-10-03 02:15 - @ go2null

A naive question : is nested set functionality needed for Projects?

#9 - 2014-10-03 02:24 - Toshi MARUYAMA

@ go2null wrote:

I tried both `rebuild!` and `rebuild_tree!`, but without the patched `rebuild!` in `awesome_nested_set` it did not work.

What is patched?

`awesome_nested_set` 2.1-stable on Redmine trunk is refactored heavily.

Unit tests cover these cases.

[source:tags/2.5.2/test/unit/project_nested_set_test.rb](https://source.rubygems.org/versions/2.5.2/test/unit/project_nested_set_test.rb)

#10 - 2014-10-03 16:57 - @ go2null

Toshi MARUYAMA wrote:

@ go2null wrote:

I tried both `rebuild!` and `rebuild_tree!`, but without the patched `rebuild!` in `awesome_nested_set` it did not work.

What is patched?

`awesome_nested_set` 2.1-stable on Redmine trunk is refactored heavily.

I was referring to my manual patching of `.id` to `.name` in a pristine `awesome_nested_set-2.1.6`.

#11 - 2014-10-03 17:16 - Toshi MARUYAMA

@ go2null wrote:

I was referring to my manual patching of `.id` to `.name` in a pristine `awesome_nested_set-2.1.6`.

Did you **copy** `awesome_nested_set` 2.1.6 to `plugin/` directory?

#12 - 2014-10-03 19:06 - @ go2null

OK, found why the problem happens.

Say I have a project with 4 subprojects ([id] name):

=> [790] Project 1, [792] Project 2, [789] Project 3, [791] Project 4

Here is the execution flow.

- `rebuild_tree!`
- `rebuild!(false)`
 - sorts by id.
 - => [789] Project 3, [790] Project 1, [791] Project 4, [792] Project 2
- `all.each { |p| p.set_or_update_position_under(p.parent) }`
 - processes each project by id order.
 - => [789] Project 3
- `set_or_update_position_under`
- `to_be_inserted_before = sibs.sort_by {|c| c.name.to_s.downcase}.detect {|c| c.name.to_s.downcase > project.name.to_s.downcase }`
 - gets project that should be on the right, sorted alphabetically by name.
 - => [791] Project 4
- if `to_be_inserted_before @move_to_left_of(to_be_inserted_before)`
 - moves to left of project found above
 - => [790] Project 1, [789] Project 3, [791] Project 4, [792] Project 2

Repeating this for each project yields:

=> [789] Project 3, [790] Project 1, [791] Project 4, [792] Project 2 *# sorted by id*
=> [790] Project 1, [789] Project 3, [791] Project 4, [792] Project 2 *# [789] P3 < P4*
=> [789] Project 3, [791] Project 4, [790] Project 1, [792] Project 2 *# [790] P1 < P2*
=> [789] Project 3, [790] Project 1, [792] Project 2, [791] Project 4 *# [791] P4 < end*
=> [792] Project 2, [789] Project 3, [790] Project 1, [791] Project 4 *# [792] P2 < P3*

#13 - 2014-10-03 19:07 - @ go2null

Toshi MARUYAMA wrote:

@ go2null wrote:

I was referring to my manual patching of `.id` to `.name` in a pristine `awesome_nested_set-2.1.6`.

Did you **copy** `awesome_nested_set` 2.1.6 to `plugin/` directory?

No, I just edited the one in `/usr/lib/...` as I was trying different things.

#14 - 2014-10-03 19:16 - @ go2null

deleted

#15 - 2014-10-03 20:03 - @ go2null

So, based on [#18018#note-12](#) above, it seems that two possible solutions are to

1. include back the vanilla awesome_nested_set 2.1.6 and
 1. patch the id to name, and
 2. no need for the all.each { |p| p.set_or_update_position_under(p.parent) } in rebuild_tree!. (At least for this use case), or
2. patch set_or_update_position_under to recursively process each to_be_inserted_before. (to_be_inserted_before becomes an array?)

Disclaimer: my ruby skills are virtually non-existent :-)

#16 - 2014-10-04 03:29 - Toshi MARUYAMA

@ go2null wrote:

Toshi MARUYAMA wrote:

@ go2null wrote:

I was referring to my manual patching of .id to .name in a pristine awesome_nested_set-2.1.6.

Did you **copy** awesome_nested_set 2.1.6 to plugin/ directory?

No, I just edited the one in /usr/lib/... as I was trying different things.

Did you modify gem *.rb files?

It causes unexpected result.

If you want to modify awesome_nested_set 2.1.6,
you should copy awesome_nested_set 2.1.6 as [r12687](#) and [r12688](#),
then remove awesome_nested_set from Gemfile.
[source:tags/2.5.2/Gemfile#L10](#)

#17 - 2014-10-05 00:34 - @ go2null

Toshi MARUYAMA wrote:

@ go2null wrote:

Toshi MARUYAMA wrote:

@ go2null wrote:

I was referring to my manual patching of .id to .name in a pristine awesome_nested_set-2.1.6.

Did you **copy** awesome_nested_set 2.1.6 to plugin/ directory?

No, I just edited the one in /usr/lib/... as I was trying different things.

Did you modify gem *.rb files?

It causes unexpected result.

If you want to modify awesome_nested_set 2.1.6,
you should copy awesome_nested_set 2.1.6 as [r12687](#) and [r12688](#),
then remove awesome_nested_set from Gemfile.
[source:tags/2.5.2/Gemfile#L10](#)

Ok, will do so next week.

#18 - 2015-02-02 01:55 - Daniel Freudenberger

Do you have any update on this? Ordering projects by name is definitely broken in redmine 2.5.3 and I guess it's still broken in redmine 2.6.* since no fix is mentioned in the changelogs.

Furthermore I can confirm that applying the patch / workaround mentioned in the ticket and running Project.rebuild_tree! fixes the issue

#19 - 2015-02-09 00:36 - Evgeniy Kostenko

Daniel Freudenberger wrote:

Do you have any update on this? Ordering projects by name is definitely broken in redmine 2.5.3 and I guess it's still broken in redmine 2.6.* since no fix is mentioned in the changelogs.

Yes, it's broken in 2.6.x. Currently i'm using 2.6.1.stable (without any modifications in sources) and after running `Project.rebuild_tree!` i got a total mess with sorting (even worst than before).

#20 - 2015-02-09 04:58 - Toshi MARUYAMA

- Related to Defect #19060: Project Sorting by Name is broken in v2.6.x added

#21 - 2015-02-09 04:59 - Toshi MARUYAMA

- Subject changed from Project Sorting by Name broken (for large project lists) in v2.5 stream to Project Sorting by Name broken (for large project lists) in v2.x stream

- Affected version deleted (2.5.0)

#22 - 2015-02-09 04:59 - Toshi MARUYAMA

- Related to deleted (Defect #19060: Project Sorting by Name is broken in v2.6.x)

#23 - 2015-02-09 05:03 - Toshi MARUYAMA

Redmine 2.x sorts projects at [source:tags/2.6.1/app/models/project.rb#L1045](https://source.rubygems.org/lookup/2.6.1/app/models/project.rb#L1045), due to awesome_nested_set bugs.

This logic is removed in Redmine 3.

#24 - 2015-02-09 05:04 - Toshi MARUYAMA

- Subject changed from Project Sorting by Name broken (for large project lists) in v2.x stream to Project Sorting by Name broken (for large project lists) in v2.x

#25 - 2015-02-09 05:04 - Toshi MARUYAMA

- Affected version set to 2.6.1

#26 - 2015-03-25 19:49 - Kevin Wojniak

I too am seeing this issue.

#27 - 2015-04-29 18:36 - Hugues C.

We are also seeing this issue, following a migration from 2.6.0 to 3.0.1.

#28 - 2015-04-29 19:42 - Toshi MARUYAMA

- Status changed from Needs feedback to Closed

- Resolution set to Wont fix

Hugues Clouâtre wrote:

We are also seeing this issue, following a migration from 2.6.0 to 3.0.1.

Migration from Redmine 2.6 to 3.0 does not touch :lft and :rgt of project table.

If your project tree of Redmine 3.0 is broken, it was broken in Redmine 2.6.

You can rebuild tree with this command in Redmine 3.0.

```
$ bundle exec ruby bin/rails runner -e production 'Project.rebuild_tree!'
```

Redmine 3.0 does not use awesome_nested_set, so I close this issue.

If you have problem in Redmine 3.0, please create new issue.

#29 - 2015-04-29 21:56 - Hugues C.

Toshi MARUYAMA wrote:

If your project tree of Redmine 3.0 is broken, it was broken in Redmine 2.6.

Ok, thank you for the quick answer. We'll try this fix. We did not have any issues in 2.6.0. Since migrating to 3.0.1 last week, all new projects created appear at the bottom, but the ones created before the migration are correctly sorted by name. Anyway, if the `Project.rebuild_tree!` command doesn't do it, we'll open a new issue.