

Redmine - Defect #6523

Sort order doesn't treat german "umlaute" correct in user listing

2010-09-28 15:49 - Frank Helk

Status:	Needs feedback	Start date:	2010-09-28
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	I18n	Estimated time:	0.00 hour
Target version:	Candidate for next major release	Affected version:	1.0.1
Resolution:			
Description			
Sorting of object seems to treat german special characters not correct.			
Ä should range equal to ae, Ö should range equal to oe, Ü should range equal to ue and ß should range equal to ss.			
Seems to be correct when user groups are listed, but broken when Users are listed (at least in groups), where ä,ö,ü and ß range behind z			

History

#1 - 2013-01-15 11:41 - Daniel Felix

- Due date set to 2013-01-22

- Status changed from New to Needs feedback

Well, any news on this? I tried this, but it seems to work on my system.

If there is no feedback on this issue, I'm going to close this next week.

#2 - 2013-01-25 16:15 - Frank Helk

Sorry for being somewhat late

I'm somewhat overwhelmed with project work, so our redmine incarnation is not very up-to-date - means I'm at 1.3.1stable here.

I've checked on that anyhow, because it's way more current than the version the problem was reported for (1.0.1). The problem ist definitely there on V1.3.1.

I've opened a group's user list, and found the following sort order (names shortened ...)

Fi...
Ge...
Gi...
Gl...
Go...
Gr...
Gä... (wrong position)
Gü... (wrong position)
Ha...
He...
Hi...

Ho...

Hä... (wrong position)

Ig...

Could anybody from DE locale please check that with the current version ?

#3 - 2013-01-27 17:31 - Toshi MARUYAMA

- *Category set to I18n*

#4 - 2013-02-19 00:43 - Filou Centrinov

In v2.2 everything is fine.

#5 - 2013-02-19 00:46 - Toshi MARUYAMA

Filou Centrinov wrote:

| *In v2.2 everything is fine.*

Ruby 1.8 or 1.9?

#6 - 2013-02-19 00:56 - Filou Centrinov

| *Ruby 1.8 or 1.9?*

ruby 1.9.1

Update: Only in the admin settings the group list is sorted correctly. When I want to add a group as a member to a project (in the project settings) the list is not correct!

Hint: In the german language the sorting of ä is equal to a. (ö=o, ü=u)

#7 - 2013-03-17 17:16 - Daniel Felix

- *Target version set to 2.4.0*

Any news on this issue?

Any idea how to fix this problem? Which coalition is used in the database table?

Maybe this could be a 2.4 goal.

#8 - 2013-05-07 09:34 - Daniel Felix

- *Due date deleted (2013-01-22)*

#9 - 2013-05-23 06:15 - Gabriel Mazetto

This problem is somehow "unfixable", as there is no silver bullet.

By choosing one collation, you will "fix" it for one language, but break it for another one, because that's what collations are made for, to solve the problem of ordering characters.

Maybe the best solution could be to high awareness of collations and help new users figure out what is the correct one they should use to support their languages.

Here are some documentations that could give more info on the problem:

- <http://dev.mysql.com/doc/refman/5.5/en/charset-charsets.html>
- <http://dev.mysql.com/doc/refman/5.5/en/charset-unicode-sets.html>
- <http://www.unicode.org/charts/uca/>

#10 - 2013-05-23 06:16 - Gabriel Mazetto

And a plus about the problem specific:

- <http://stackoverflow.com/questions/5526169/what-is-the-best-mysql-collation-for-german-language>
- <http://stackoverflow.com/questions/2344118/utf-8-general-bin-unicode>
- <http://dev.mysql.com/doc/refman/5.5/en/charset-collation-effect.html>

(sorry, had to split it in two answers as redmine "flaged" as a possible spam)

#11 - 2013-05-23 09:02 - Etienne Massip

Gabriel Mazetto wrote:

This problem is somehow "unfixable", as there is no silver bullet.

By choosing one collation, you will "fix" it for one language, but break it for another one, because that's what collations are made for, to solve the problem of ordering characters.

Redmine/Rails could handle ORDER BY .. COLLATE part depending on user profile?

#12 - 2013-09-15 15:50 - Jean-Philippe Lang

The problem seems to be more with ruby #sort:

Using the database sort (MySQL):

```
irb(main):002:0> Group.sorted.all.map(&:name)
←[1m←[35mGroup Load (0.0ms)←[0m SELECT `users`.* FROM `users` WHERE `users`.`
type` IN ('Group') ORDER BY users.lastname ASC
=> ["Gä", "Gr", "Gü"]
```

Using ruby #sort (1.9.3):

```
irb(main):003:0> Group.all.map(&:name).sort
←[1m←[36mGroup Load (0.0ms)←[0m ←[1mSELECT `users`.* FROM `users` WHERE `user
```

```
s.`type` IN ('Group')←[0m  
=> ["Gr", "Gä", "Gü"]
```

Filou Centrinov wrote:

| *Update: Only in the admin settings the group list is sorted correctly.*

Because it uses a SQL ORDER

| *When I want to add a group as a member to a project (in the project settings) the list is not correct!*

In 2.2, groups are sorted using ruby #sort for when adding a project member. Since 2.3, it uses a SQL ORDER statement so it should be OK now.

#13 - 2013-10-13 10:02 - Jean-Philippe Lang

- Target version changed from 2.4.0 to Candidate for next major release

2.4 feature freeze

#14 - 2015-04-21 14:40 - Rafał Lisowski

I had the same problem on edit/add issue page and <https://github.com/twitter/twitter-cldr-rb#sorting-collation> help me.

I added this twitter_cldr gem to Gemfile.local and modify

app/helpers/application_helper.rb

```
--- a/app/helpers/application_helper.rb  
+++ b/app/helpers/application_helper.rb  
@@ -392,7 +392,10 @@ module ApplicationHelper  
  s << content_tag('option', "<< #{l(:label_author)} >>", :value => author.id)  
  end  
  groups = "  
- collection.sort.each do |element|  
+ lang = TwitterCldr.supported_locale?(current_language) ? current_language : :en  
+ collator = TwitterCldr::Collation::Collator.new(lang)  
+ collection.sort do |a, b|  
+ collator.compare(a.name, b.name)  
+ end.each do |element|  
  selected_attribute = ' selected="selected"' if option_value_selected?(element, selected) || element.id.to_s == selected  
  (element.is_a?(Group) ? groups : s) << %(<option value="#{element.id}"#{selected_attribute}>#{h element.name}</option>)  
end
```