

Redmine - Feature #16549

Set multiple values in emails for list custom fields

2014-04-07 14:31 - Felix Schäfer

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Custom fields	Estimated time:	0.00 hour
Target version:	3.4.0		
Resolution:	Fixed		
Description			
It is currently not possible to set multiple values for list custom fields in incoming emails.			
Related issues:			
Related to Redmine - Feature #11537: Allow multiple default values for custom...			New
Related to Redmine - Defect #26148: Cannot import custom fields with multiple...			Closed

Associated revisions

Revision 16380 - 2017-03-08 20:55 - Jean-Philippe Lang

Allow to set multiple values in emails for list custom fields (#16549).

History

#1 - 2014-04-07 14:34 - Felix Schäfer

- File 16549.diff added

- Status changed from New to Resolved

Attached is a patch to add this possibility for list custom fields, it doesn't support user or version custom fields for now. Furthermore, the values are matched case-sensitive.

#2 - 2014-04-07 22:27 - Felix Schäfer

- File 16549.diff added

Here's an updated version of the patch with support for any sort of field that can have multiple values and matches are case-insensitive. The tests are examples and can be spun out to their own tests if needed.

#3 - 2014-04-16 12:17 - Jan from Planio www.plan.io

- Related to Feature #11537: Allow multiple default values for custom fields added

#4 - 2014-04-29 17:49 - Felix Schäfer

- File 16549.diff added

There still were 2 small issues with the last proposed patch. This one should be good. (unfortunately I can't delete the old/defective ones, sorry for the noise)

#5 - 2016-08-10 14:02 - Yiannis Tsiouris

Is there any update on that? The patch definitely needs update as it doesn't apply cleanly; although, it works! I might be able to update the patch if more people are interested. I find it very useful and I'd really like that to be merged in core...

By the way, how come you used ';' for value separator and not ',' (that is the default when adding multiple values via Redmine UI)?

#6 - 2016-08-10 23:19 - Yiannis Tsiouris

- File `set_multiple_values_via_email.diff` added

First try in the attached!

#7 - 2016-08-11 23:38 - Yiannis Tsiouris

- File `set_multiple_values_via_email.diff` added

Patch updated 'cause some tests were failing! Thanks!

#8 - 2016-09-04 18:19 - Toshi MARUYAMA

- Target version set to 3.4.0

#9 - 2016-09-04 18:19 - Toshi MARUYAMA

- Status changed from Resolved to New

#10 - 2016-10-01 14:00 - Jean-Philippe Lang

Due to the changes done in [r14503](#), users can be specified in different ways (name, email, username...) and the proposed path is not compatible with that. I understand that this patch more or less handles values that contain the separator but it would be much simpler to split the keyword first with the separator then search for the corresponding values.
Is it an acceptable solution?

#11 - 2016-10-03 09:41 - Felix Schäfer

Yiannis Tsiouris wrote:

By the way, how come you used ';' for value separator and not ','?

Because you're less likely to use ; in a possible value for a field than ,. For example, is "Simon, Garfunkel, Pitt, Brad" the values "Simon, Garfunkel" and "Pitt, Brad" or "Simon", "Garfunkel" and "Pitt, Brad"? It could also be "Simon" and "Garfunkel, Brad, Pitt" and so on.

; is still a compromise in that case. We could also use for example | (pipe symbol) or an even more obscure unicode character that has even less likelihood of being in a custom value, but then it gets tedious entering that in emails.

(that is the default when adding multiple values via Redmine UI)

Do you have an example? Do you mean when they are shown or when you enter them?

#12 - 2016-10-03 10:10 - Felix Schäfer

Jean-Philippe Lang wrote:

Due to the changes done in [r14503](#), users can be specified in different ways (name, email, username...) and the proposed path is not compatible with that. I understand that this patch more or less handles values that contain the separator but it would be much simpler to split the keyword first with the separator then search for the corresponding values.
Is it an acceptable solution?

Splitting by separator first essentially makes all custom values containing that separator unfit to use in the email updates. The code would be simpler in that case though, yes.

If it is an explicit decision to say that such values with the separator will not work in the email keywords, it is fine for me. In this case using a less likely character as the separator, i.e. ; rather than , , would ensure that this edge case happens less often.

On the other hand [source/trunk/app/models/principal.rb@15884#L152](#) uses nearly the same technique as our proposed patch, even if it's written a little differently. It tries to match the name against the list of logins or mails or "firstname lastname" or names. That method could be refactored a bit to 2 methods, one emitting the list of logins, mails, "firstname lastname", names with the corresponding user objects, the other trying to match the passed string to that list. The first method could then be used in the fashion of the possible values for the custom fields, thus allowing to first create a list of possible values and then match the values against that list. We could prepare a patch proposal if this sounds like a working solution for you.

Please note that our proposed solution (first create a list of possible values, try to match those possible values against the value in the email) allows for the separator to be in the custom field possible values, it still has a caveat. If a possible value is a combination of 2 other values with the separator in between, this method can not decide which it matches. For example if the separator is ; and the possible values are "one", "two" and "one; two", our proposed method would not be able to decide whether "one; two" as a keyword value in an email is the values "one" and "two" or the value "one; two" for the custom field.

I hope I was clear enough in my explanation. I can provide more code and/or examples if necessary.

#13 - 2016-10-08 10:33 - Jean-Philippe Lang

Felix Schäfer wrote:

Please note that our proposed solution (first create a list of possible values, try to match those possible values against the value in the email) allows for the separator to be in the custom field possible values, it still has a caveat. If a possible value is a combination of 2 other values with the separator in between, this method can not decide which it matches. For example if the separator is ; and the possible values are "one", "two" and "one; two", our proposed method would not be able to decide whether "one; two" as a keyword value in an email is the values "one" and "two"

or the value "one; two" for the custom field.

Understood, that's what I meant when saying "I understand that this patch more or less handles values that contain the separator".

#14 - 2017-03-08 21:01 - Jean-Philippe Lang

- Status changed from New to Closed
- Assignee set to Jean-Philippe Lang
- Resolution set to Fixed

Feature is added in [r16380](#). It works with values that contain the separator (,) in a similar way as the proposed patch (see [some of the tests](#)) and is compatible with the curent implementation for enumeration custom fields and user custom fields.
Please let me know if there's anything wrong.

#15 - 2017-03-08 21:14 - Felix Schäfer

Thank you for your work on this. From what I read and judging from the tests I think this should work great!

#16 - 2017-07-02 07:31 - Mischa The Evil

- Subject changed from Set multiple values in emails for list custom fields to Set multiple values in emails for list custom fields

#17 - 2017-07-18 16:14 - Toshi MARUYAMA

- Related to Defect #26148: Cannot import custom fields with multiple values in 3.3 added

Files			
16549.diff	3.7 KB	2014-04-07	Felix Schäfer
16549.diff	4.22 KB	2014-04-07	Felix Schäfer
16549.diff	4.22 KB	2014-04-29	Felix Schäfer
set_multiple_values_via_email.diff	5 KB	2016-08-10	Yiannis Tsiouris
set_multiple_values_via_email.diff	5.02 KB	2016-08-11	Yiannis Tsiouris