Redmine - Defect #9849

Custom field values of several issue custom field formats don't get set/updated properly during email issue creation/reply

2011-12-21 03:05 - Mischa The Evil

Status:	New	**Start date:**	
Priority:	High	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Custom fields	**Estimated time:**	0.00 hour
Target version:			
Resolution:		**Affected version:**	

Description

As of r4394 we can set/update custom field values using email issue creation/reply. Though currently custom field values of version/user/boolean issue custom field formats don't get set/updated properly during email issue creation/reply.

Steps to reproduce:

1. Start with a clean and vanilla Redmine source:trunk@7970
2. Load default data (roles, statuses etc.)
3. Add a project and configure it
4. Configure RedmineReceivingEmails
5. Create nine issue custom fields (ICF); each without default value, enabled as an issuelist filter and with another format (so covering all nine ICF formats)
6. Send a properly laid out email containing the needed and to-be-tested fields and their values using keyword: value pairs
7. Return to Redmine that created the issue from email submission (though it happens also on email-reply)

Behavior description:

- **Wrong behavior** with **version-format** ICF:
  - value set by mail **is displayed correctly** on issues/show and in columns on issues/index **but**:
  1. during next issue edit, the field **isn't pre-filled** with the current value.
  2. on the issuelist the custom field filter **doesn't catch** the issues with the custom value "set" via email

- **Wrong behavior** with **user-format** ICF:
  - value set by mail **is not displayed** on issues/show nor in columns on issues/index, **besides**:
  1. during next issue edit, the field **isn't pre-filled** with the current value.
  2. on the issuelist the custom field filter **doesn't catch** the issues with the custom value "set" via email

- **Wrong behavior** with **boolean-format** ICF:
  - value set by mail **is not displayed** on issues/show nor in columns on issues/index, **besides**:
  1. during next issue edit, the field **isn't pre-filled** with the current value.
  2. on the issuelist the custom field filter **doesn't catch** the issues with the custom value "set" via email

- Correct behavior with list-format ICF:
  - value set by mail *is displayed correctly* on issues/show and in columns on issues/index, *and*:
  1. during next issue edit, the field *is pre-filled* with the current value.
  2. on the issuelist the custom field filter *does catch* the issues with the custom value "set" via email

- Correct behavior with date-format ICF:
  - value set by mail *is displayed correctly* on issues/show and in columns on issues/index, *and*:
  1. during next issue edit, the field *is pre-filled* with the current value.
  2. on the issuelist the custom field filter *does catch* the issues with the custom value "set" via email

- Correct behavior with text-format ICF:
  - value set by mail *is displayed correctly* on issues/show and in columns on issues/index, *and*:
  1. during next issue edit, the field *is pre-filled* with the current value.
  2. on the issuelist the custom field filter *does catch* the issues with the custom value "set" via email

- Correct behavior with long text-format ICF:
  - value set by mail *is displayed correctly* on issues/show and in columns on issues/index, *and*:
  1. during next issue edit, the field *is pre-filled* with the current value.

2. on the issuelist the custom field filter *does catch* the issues with the custom value "set" via email

- Correct behavior with integer-format ICF:
  ○ value set by mail *is displayed correctly* on issues/show and in columns on issues/index, *and*:
  1. during next issue edit, the field *is pre-filled* with the current value.
  2. on the issuelist the custom field filter *does catch* the issues with the custom value "set" via email

- Correct behavior with float-format ICF:
  ○ value set by mail *is displayed correctly* on issues/show and in columns on issues/index, *and*:
  1. during next issue edit, the field *is pre-filled* with the current value.
  2. on the issuelist the custom field filter *does catch* the issues with the custom value "set" via email

At first I though it was affecting version/user format ICF's only, but then I included all the other formats in my tests and noticed incorrect behavior with (only) the boolean format too.
In the tests of all of the above situations the keywords and their values were correctly stripped (thus recognized by Redmine), and - more importantly - the **correct values** were/are written into the custom_values table.

Note: to be sure I tested this behavior for the other (native) issue attributes too but these don't seem affected.

Environment:

Redmine: source:/trunk@7970
Ruby: 1.8.7 [x86_64-linux] (REE 1.8.7-2011.03)
RubyGems: 1.3.7
Rack: 1.1.2
Rails: 2.3.14
MySQL: 5.1.41
Browser: Mozilla Firefox 7.0.1

I wasn't able to retrieve any more info from e.g. the logs.

Please let me know if more information is required.

Related issues:

Related to Redmine - Defect #25726: Issue details page shows default values f...	**Closed**

History

#1 - 2013-10-20 16:06 - Olivier Houdas

In /app/models/custom_fields.rb, function

```
def possible_values_options(obj=nil)
```

it is building list of arrays, which then do not work very well when calling

```
possible_values_options.detect {|text, id| text.downcase == keyword}
```

If you change for example

```
when 'bool'
      [[l(:general_text_Yes), '1'], [l(:general_text_No), '0']]
```
to

```
when 'bool'
      [l(:general_text_Yes), '1', l(:general_text_No), '0']
```

it works better.
However, I wouldn't bet on whether it should be fixed there rather than in the call to the .detect() method.

#2 - 2017-05-25 13:14 - Toshi MARUYAMA

- Related to Defect #25726: Issue details page shows default values for custom fields that aren't actually set added