

Redmine - Feature #15340

Custom field referencing other tickets

2013-11-13 12:12 - lligid

| | | | |
|--|---------------|------------------------|-------------------|
| Status: | New | Start date: | |
| Priority: | Normal | Due date: | |
| Assignee: | | % Done: | 0% |
| Category: | Custom fields | Estimated time: | 0.00 hour |
| Target version: | | | |
| Resolution: | | | |
| Description | | | |
| <p>Redmine could become much more powerful if it would be possible to add custom fields which reference other tickets. It should be possible to filter the set of tickets referenceable per field. Via this a user could setup very powerful structures. The field should be</p> <ul style="list-style-type: none">• configurable to show #id, name or both• displayed as a hyperlink which when clicked opens the referenced ticket <p>In general handling tickets as objects and applying filters to create sets of filters and support referencing between tickets would be a very powerful architectural choice. Next logical steps would be powerful reporting features (you want to be able to analyze according to the interdependencies), powerful dependency handlers (e.g. showing content of a referenced ticket directly in a referencing ticket).</p> <p>Example 1: customer database</p> <ul style="list-style-type: none">• a user could create a customer project with tickets per customer• in a reference custom field the admin could define a filter which only allows references to tickets in this customer project• the user can select from the available customer tickets and the ticket would allow to click on the customer reference and get detailed information <p>Example 2: Multi scale categorization</p> <ul style="list-style-type: none">• You have a hierarchical project structure setup with architectural work packages• You additionally want to group V&V tasks, engineering tasks, ... across the hierarchy and across trackers• You additionally want to group according to features (across the hierarchy), which may be available as tickets already• You additionally want to group according to requirements (not necessarily the same as a feature) | | | |
| Related issues: | | | |
| Related to Redmine - Feature #13690: Allow plugins to be able to add relation... | | New | |
| Related to Redmine - Feature #7849: custom issue relation types | | New | 2011-03-13 |

History

#1 - 2013-12-05 08:09 - Toshi MARUYAMA

- Category set to Custom fields

#2 - 2013-12-16 18:18 - Dipan Mehta

+1. Very useful. One of the important usecase is to indicate a chain of cause-effect issues where the fix of ticket #xyz has caused the current bug. Most issue relations follow-precend / blocks etc. are used mainly to deal with planning; however there is no way to create custom relationships between issue. Hence, CF of the type "issue-reference" will make it very easy to create such relations easily.

#3 - 2014-01-08 19:22 - Pierre Luce

- File ISSUE_TYPE_CUSTOM_FIELDS.diff added

Here is a basic implementation I made (from trunk development version)

Just use a classical Text custom field. If the content begins with '#' (like in "#42"), it will convert the number that follows in link to the corresponding issue. No idea of what happens if the number is not a number or if no issue corresponds.

If you list #values separated by a comma ("#42,#24,#422"), it will display as a list of links.

#4 - 2014-01-09 06:44 - Dipan Mehta

Nice to see some progress on this.

However, even today this is possible by creating a text field (and using one of the plugins that converts text in link). Unfortunately, it will except all other type of text as well.

I think a good way would be equivalent to what works in related issues or parent issue field.

1. The field should accept only integers, (avoiding any invalid text)
2. The field can use the same tool tip to show issue title as one enters the number (used by parent/related issue)
3. The field can render issue along with subject line under format_value

If you want to gain more insight about how custom fields of arbitrary types can be developed look at the project:

<http://projects.andriylesyuk.com/projects/extended-fields>

#5 - 2014-01-09 09:39 - Il Igid

thanks for the nice little implementation for prototyping.

For a final solution I would rather make a new field type available for custom fields and make the use very similar / same to the current relation fields. However, the current relation field implementation does contain some specifics (e.g. relation types) which do not make them easily reusable for a more generic theme without a refactoring.

#6 - 2014-01-09 20:27 - Pierre Luce

[#13690](#) proposes a solution to allow new relationships to be added. However, I'm not fluent enough in Ruby to implement it (in fact, my patch is my very first lines of Ruby)

#7 - 2014-01-30 08:48 - Toshi MARUYAMA

- Related to Feature [#13690](#): Allow plugins to be able to add relationship types added

#8 - 2014-01-30 08:49 - Toshi MARUYAMA

- Related to Feature [#7849](#): custom issue relation types added

#9 - 2014-05-09 08:18 - Quanlong He

+1. It's a very useful feature.

#10 - 2020-10-02 10:38 - Stefan Lindner

+1

#11 - 2020-10-23 08:23 - Fabien Durand

+1

#12 - 2021-07-23 16:40 - shawn freeman

Related to [#4799](#)?

Files

| | | | |
|-------------------------------|---------|------------|-------------|
| ISSUE_TYPE_CUSTOM_FIELDS.diff | 1.08 KB | 2014-01-08 | Pierre Luce |
|-------------------------------|---------|------------|-------------|