Redmine - Feature #6717

Custom list field with dynamic list content

2010-10-21 00:36 - Stefano Lenzi

Status:	New	**Start date:**	2010-10-21
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Custom fields	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

Use Case

The administrator of the **Galaxy** project adds a new custom list field *"affected version"*, which contains all the **version** defined for Galaxy project, so that when a new version is created the administrator does not have to add it to list manually

Query solution

When creating a custom field of type list the text area dedicated to the list content could contain a SQL query that will be used for populating the list when the field is shown. For legacy purpose, a checkbox "dynamic list" is shown which if select would use the SQL query

Related issues:

Related to Redmine - Feature #6716: Linking version to category	**New**	**2010-10-21**
Related to Redmine - Feature #2096: Custom fields referencing system tables (...	**Closed**	**2008-10-27**
Related to Redmine - Feature #13016: Dynamic adding values to custom fields	**New**	
Related to Redmine - Feature #9734: Custom field , value list from database q...	**New**	
Related to Redmine - Feature #13143: Populate dropdown custom field list base...	**New**	
Related to Redmine - Feature #15177: Dynamic query fields	**New**	
Related to Redmine - Feature #19224: Custom fields / List - make user editable	**New**	
Related to Redmine - Feature #22026: Conditional custom fields	**New**	

History

#1 - 2010-10-21 00:40 - Stefano Lenzi

The **General Solution** of issue #6716 would enhance even more the flexibility of the issue module to adapt easily to company needs, it also increases the usability and reduce the compilation time of the *Issues*.

#2 - 2010-12-29 12:05 - Fernando Hartmann

+1

This is really near as what we need, we need a custom field filed by a query that retrieve our customers list from the ERP table, but this ERP is on other database, so, there must be some connection information too. Then this query can access other databases !

#3 - 2010-12-29 20:54 - Terence Mill

Stefano Lenzi wrote:

> The **General Solution** of issue #6716 would enhance even more the flexibility of the issue module to adapt easily to company needs, it also increases the usability and reduce the compilation time of the *Issues*.

Huh! That would be a nice feature indeed. Instead of SQL Queries to froreign Databases, support for a common standard(?) http WS protocol for data queries would be better. So every dataprovider can offer its data "lists" via a web service where data can ve queried ad search results get returned. Someone has an idea what standards there are?

#4 - 2010-12-30 00:27 - Terence Mill

This is related to Issue #2096

#5 - 2013-03-18 20:34 - Dipan Mehta

While most usage will be served well with having Versions and Users type fields it would be really nice if the scope of this ticket is made extensible from there:

there are two specific cases where this can be thought of :

1. The custom fields which has many set of 'possible values' list:

The dynamic lists exists for many things - e.g. resources being used, configuration properties, OS and platform specific etc. In particular case, we use a config specific field with an *almost* same named field - in each project. So the purpose and role of that field is same in all projects- but it is actually a different custom field just because the list of "Possible values" is different in each project! And let me tell you, it's easier said than done when we 'create' these fields. All aspects of works flows - where in each state/tracker/role where the particular field become 'must', 'read-only', permissions and a whole set of things now needs to be repeated for each project. And then, when we create another project? do this whole thing all over again.

How this can be implemented: Ideally, it is very important that a given Dynamic List can have multiple set of possible-list-of-values, and either the source list can be selected in project. Alternatively, we can put a complete list of possible values in the main definition - and then specific projects can enable disable the fields that suits them. But even as the possible values differ per project actual workflow and behavior can remain as is for all aspects.

#6 - 2013-03-18 20:42 - Dipan Mehta

2. Custom fields which derive values dependent from external system:

The Request Tracker ticketing system has something called an External Fileds . A potential use case could be to hook to external systems - e.g. I can lookup on the external Inventory systems to know the list of machines that can apply for the given context, or a list of versions of external platforms -e.g. Ruby, or Chrome or even list of stable Linux versions, or for that matter Bug numbers, Test cases of external systems etc. In our case can also be a specific lists related to config parameters without which the bugs are hard to reproduce. The essence is that the system which populates such a list is essentially external to Redmine and hence it would be wrong to keep manually updating the admin to keep adding items such as this inside the table list. Also the relevance could change depending on the project - hence the said list could be unique for each project.

How should this be implemented? : Definitely calling upon external DB directly could be very wrong. And currently i see no webserivce standard's schema exists for such a purpose, and ideally, even if we try to create one, it's not practical to apply everywhere. The way it should be done should be very similar to how email based issue creation works. There should be a REST API that exposes a given custom_field ID and it's possible current values can be updated. Now one can write any script on ones' own that fetches the right set of list from whatever appropriate place and run a cron-type job to keep updating this! Even the current custom_fields can be very easily extended in this manner. One very critical thing to support is that in the history some fields could have been valid - so they could exist in DB, but now they might not be among the possible values - this needs to be supported.

#7 - 2013-12-16 18:29 - Dipan Mehta

Any feedback or update on this?

#8 - 2013-12-18 06:14 - Toshi MARUYAMA

- Related to Feature #13016: Dynamic adding values to custom fields added

#9 - 2013-12-18 06:18 - Toshi MARUYAMA

- Related to Feature #9734: Custom field , value list from database query added

#10 - 2013-12-18 06:25 - Toshi MARUYAMA

- Related to Feature #13143: Populate dropdown custom field list based on previous another selection added

#11 - 2014-06-12 08:48 - Maxime Vez

+1

#12 - 2014-07-04 13:35 - Maicon Zucco

+1

#13 - 2014-07-08 09:07 - Yar n

one more vote.

#14 - 2014-07-16 09:50 - Alice Etchegaray

+1

#15 - 2014-08-13 10:23 - Toshi MARUYAMA

- Category changed from Issues to Custom fields

#16 - 2015-01-26 09:55 - sebastien lemaitre

+1

#17 - 2015-04-03 10:38 - Andrey Tatarnikov

+1

#18 - 2015-05-18 13:35 - Toshi MARUYAMA

- Related to Feature #15177: Dynamic query fields added

#19 - 2015-07-03 05:44 - Toshi MARUYAMA

- Related to Feature #19224: Custom fields / List - make user editable added

#20 - 2016-04-09 10:16 - Dipan Mehta

Is this coming anytime soon?

#21 - 2016-05-05 05:54 - Toshi MARUYAMA

- Related to Feature #22026: Conditional custom fields added

#22 - 2016-06-01 14:59 - Jefferson Campos

Is this coming anytime soon? I din't find it in roadmap. This feature would be very very usefull.

#23 - 2016-11-25 05:11 - Mei Chua

+1

#24 - 2017-04-07 01:35 - Vincent Bruggeman

+1

#25 - 2017-10-10 16:29 - Roberto Vieweg

+1

#26 - 2020-07-21 12:06 - Amir Sepahram

+1

#27 - 2020-12-29 15:48 - Alexey Smirnov

https://www.redmine.org/plugins/custom_field_sql

#28 - 2021-06-02 00:13 - Alexandr Chernyaev

+1