

## Redmine - Feature #6637

### "unwanted" features as plugins

2010-10-12 13:17 - Albert Rosenfield

<b>Status:</b> Closed	<b>Start date:</b> 2010-10-12
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	
<b>Resolution:</b>	
<b>Description</b> We started using Redmine for issue tracking (but not for a software development project).  (It's a great piece of work! I've already started writing a few plugins.)  A couple of features are irrelevant in our scenario: * News * Repository * Forum * Calendar * Gantt  I can remove them easily by disabling access to them for normal administrators and the users they create.  The problem is that there are a number of artifacts from these features still present when you create a new issue, for example "start date", "due date", "estimated time", "% done" etc. These fields do not add value (as I mentioned this is not a software development scenario), which means that they only make the system more complex to use. They devalue the installation by making it more difficult to use. (Again, in our scenario only, not generally.)  It would be nice if the above mentioned features could be turned into plugins, so that they could be removed completely for installations where they are unwanted, including all the fields in various forms that currently remain.  I am willing to put hours into this, but I need a bit of guidance as to what is the easiest way / best solution / whatever.	
<b>Related issues:</b>	
Related to Redmine - Feature #1091: Disabling default ticket fields per tracker	<b>Closed</b> <b>2008-04-23</b>

### History

#### #1 - 2010-10-12 22:08 - Felix Schäfer

The attributes you name are inherent attributes of tickets in the redmine model, not things that are induced or "only there for" any of the modules you mentioned. There are plans to have more flexibility wrt the attributes shown with issues, but that's not something that will happen soon.

Your best bet to remove those would be to patch the core views and remove the elements you don't need, all should have default values with which the issues should validate every time.

#### #2 - 2010-10-13 11:27 - Albert Rosenfield

The attributes you name are inherent attributes of tickets in the redmine model, not things that are induced or "only there for" any of the modules you mentioned.

Apparently we have no use for any of the modules that make use of these attributes, what with the attributes being completely useless to us and all.

There are plans to have more flexibility wrt the attributes shown with issues, but that's not something that will happen soon.

Hmm, what is the stumbling block?

From a totally superficial point of view, it seems that the part of the issue view where fields are displayed is a pretty straight-forward two-column layout.

I could imagine working on making an administrative settings page where each field can be shown/hidden, placed in column left/right, and moved up/down at will.

Your best bet to remove those would be to patch the core views and remove the elements you don't need,

I did this already, but the attributes still pop up in various places (filter dropdowns for example). Then people come around asking me why they are there and if I can get rid of them (and if I can add other, custom, attributes that are missing which they mistook them for - but that's another story).

My solution so far is to dig into the database and remove the relevant columns by hand. This causes the web interface to fail. I then test every page in the web interface and fix it up to work without the unwanted attributes.

The problem with this approach is maintainability, every time I upgrade Redmine, I have to start all over. I would be much happier with a generic solution.

all should have default values with which the issues should validate every time.

Agreed, this part works perfectly.

### #3 - 2010-10-13 12:47 - Felix Schäfer

Albert Rosenfield wrote:

There are plans to have more flexibility wrt the attributes shown with issues, but that's not something that will happen soon.

Hmm, what is the stumbling block?

Time, and other tasks to be done before that, particularly the refactoring going on (should provide more flexibility in the code) and a complete overhaul of the permissions system (which currently is tied too much to projects).

From a totally superficial point of view, it seems that the part of the issue view where fields are displayed is a pretty straight-forward two-column layout.

I could imagine working on making an administrative settings page where each field can be shown/hidden, placed in column left/right, and moved up/down at will.

The idea was to have it even configurable per role what fields to show or not, should it be configurable per project or not (and seeing everything tends to be configurable per project more and more, it probably will be), provide facilities for plugins/modules, deactivate other parts of the UI that would need knowledge about those fields, as well as search, and so on, and so on. All that should also be taken with a grain of performance salt, if you end up having 30 "UI"-queries to the DB for 3 queries to fetch one issue and related info, that won't fly either. Anyway, there's not been much discussion going on around this, but it's probably not something that will happen "overnight".

### #4 - 2010-10-13 13:36 - Albert Rosenfield

The idea was to have it even configurable per role what fields to show or not,

Sounds nice.

should it be configurable per project or not  
(and seeing everything tends to be configurable  
per project more and more, it probably will be)

Fair enough.

Personally, I would configure the layout as a whole, not per role and definitely not per project. I think it results in too much confusion if one user sees one thing and another user sees something else. I think that if "you" need to hide some fields from some users, what you have is probably a problem with the organisations workflow more than it is a problem in the Redmine UI. But that's just me :-).

[...] deactivate other parts of the UI that would need knowledge about those fields

Agree, needs to be possible somehow..

All that should also be taken with a grain of performance salt,  
if you end up having 30 "UI"-queries to the DB for 3 queries to  
fetch one issue and related info, that won't fly either.

Yes, I see your point.

Google Code has an interesting solution to this problem, where all the fields that are not related to other issues (think foreign keys or similar) are stored as in the same 'labels' structure, even when they have nothing in common.

Such as  
Severity-Critical  
Target-1.7  
Affects-Everyone

The first part is the pseudo-field-name, and the second part is the value. That way, you can query for a whole bunch of field values in one big swoop of a query.

Also possible to query for which fields are actually used in a project/installation, if you make it easy to query for just the first part (in example above, Severity, Target and Affects). One query gives all the possible search criteria, for example.

[Technically, database's have substring indexes that could be useful, or triggers that update a cache table, or .. there might be more techniques]

(Also fixes a default value problem quite nicely, because you don't have to insert odd NULLs in any field.)

I don't like Google Code's UI very much, where you can manually enter anything into any label. But I think the storage structure where non-referential fields are all stored as labels is very nice.

Anyway, there's not been much discussion going on around this, but it's probably not something that will happen "overnight".

Okay.

I am upgrading a specific redmine install right now, and I would like to remove unwanted stuff again, what do you think is the best approach?

(My best idea right now is I could restart with removing database columns and patching the UI, but it seems silly.)

#### **#5 - 2010-10-13 20:25 - Felix Schäfer**

Albert Rosenfield wrote:

I am upgrading a specific redmine install right now, and I would like to remove unwanted stuff again, what do you think is the best approach?

(My best idea right now is I could restart with removing database columns and patching the UI, but it seems silly.)

Can't really think of anything better really, other than delving into the code and making sure you miss nothing... Regarding the maintainability, I have found git and having a branch per "feature" (or in your case lack thereof) very handy, as the changes you do to it are focused on one thing only. That might make your job and upgrading a little easier.

#### **#6 - 2010-11-02 01:02 - Gergely Daróczi**

This feature would be really helpful for us also.  
If anyone could manage to implement this option, please let me know.

#### **#7 - 2015-03-06 04:42 - Go MAEDA**

- *Related to Feature #1091: Disabling default ticket fields per tracker added*

#### **#8 - 2015-03-06 04:44 - Go MAEDA**

- *Status changed from New to Closed*

We can disable standard fields from Redmine 2.1 ([#1091](#)).  
I close this issue.