

<div>ステータス: New</div> <div>優先度: 通常</div> <div>担当者:</div> <div>カテゴリ: Issues workflow_41</div> <div>対象バージョン:</div> <div>Redmineorg_URL: https://www.redmine.org/issues/12005</div> <div>category_id: 41</div> <div>version_id: 0</div> <div>issue_org_id: 12005</div> <div>author_id: 57516</div> <div>assigned_to_id: 0</div> <div>comments: 160</div>	<div>開始日: 2022/05/09</div> <div>期日:</div> <div>進捗率: 90%</div> <div>予定工数: 0.00時間</div> <div>作業時間: 0.00時間</div> <div>status_id: 1</div> <div>tracker_id: 2</div> <div>plus1: 39</div> <div>affected_version:</div> <div>closed_on:</div> <div>affected_version_id:</div>
<div>説明</div> <div>Ability to hide field</div> <div>Related to #8050</div> <div>workaround</div> <div>Use Plugin</div> <div>https://github.com/nanego/redmine_customize_core_fields</div> <div>(hide or show core fields based on the user roles per project.)</div>	
<div>journals</div> <div>Tested it on trunk today, works fine!</div> <div>Yes, all copyrights goes to "Gurvan Le Dromaguet":http://www.redmine.org/users/39193.</div> <div>Thanks, man.</div> <div>I would appreciate it so much guys if you could implement this in the next release. Thank you in advance :)</div> <div>+1</div> <div>+1</div> <div>Which one of the attached files is the newest?</div> <div>David Robinson, latest.</div> <div>It would be quite useful to have this feature in the next release!</div> <div>+1</div> <div>The way I understand this, hidden should also be read-only (hidden should not be MORE permissive than read-only, but LESS permissive) in the attached (v0.03) a user that has "hidden access" to a field can still edit it...</div> <div>I would suggest functionality along the lines of:</div> <div><pre>def read_only_attribute_names(user=nil)</pre></div>	

```
workflow_rule_by_attribute(user).reject {|attr, rule| rule != 'readonly' and rule != 'hidden'}.keys
end
```

P.S.: I don't know Ruby, nor anything about the Redmine codebase, so if this suggestion is foolish this is why.

And wrote:

David Robinson, latest.

Thank you # And for your prompt reply.
But I am afraid I see the following:

```
workflow_hidden_field.diff Magnifier (4.79 KB) # And, 2012-10-02 12:03
workflow_hidden_field_v0.03.diff Magnifier (8.73 KB) # And, 2012-10-02 12:03
```

That is: two files with the same date and time.
Can I assume that these files are sorted in some way?
Maybe by version ascending (That is oldest file on top, newest at bottom)? Or maybe by version descending?

OK, thanks David Robinson. Unfortunately, I'm not familiar with Ruby language too. I'll check you advice later.

David Robinson wrote:

And wrote:

David Robinson, latest.

Thank you # And for your prompt reply.
But I am afraid I see the following:

```
workflow_hidden_field.diff Magnifier (4.79 KB) # And, 2012-10-02 12:03
workflow_hidden_field_v0.03.diff Magnifier (8.73 KB) # And, 2012-10-02 12:03
```

That is: two files with the same date and time.
Can I assume that these files are sorted in some way?
Maybe by version ascending (That is oldest file on top, newest at bottom)? Or maybe by version descending?

Answer is very simple, I have not permissions to change order, or remove old patch attachment. So we get what we have.

David Robinson wrote:

That is: two files with the same date and time.
Can I assume that these files are sorted in some way?
Maybe by version ascending (That is oldest file on top, newest at bottom)? Or maybe by version descending?

And yes - latest attachments at bottom.

Most recent version of my patch is 0.03, sorry I did not put any versioning in my first pulication.

Are you guys going to include this patch in 2.1.3 release? Would be awesome :)

I did some tests - applied 0.03 patch and 2.1.0 (ok, not latest but it shouldn't have impact).

- When clicking on update, hidden fields appear with content, it's editable and once done, it disappear from 'header' of the issue,
- Hidden fields appear in mails notification,
- Hidden fields modifications appears in history,
- Hidden fields appear in issues tables, filter, columns..

So those fields are not completely hidden, is it wanted ? Can someone confirm those behaviour ?

Thanks for your testing :)

No it's not wanted, dev is not complete and it looks like I have a bad condition somewhere, fields should not appear in edit :(

claude g wrote:

I did some tests - applied 0.03 patch and 2.1.0 (ok, not latest but it shouldn't have impact).

- When clicking on update, hidden fields appear with content, it's editable and once done, it disappear from 'header' of the issue,
- Hidden fields appear in mails notification,
- Hidden fields modifications appears in history,
- Hidden fields appear in issues tables, filter, columns..

So those fields are not completely hidden, is it wanted ? Can someone confirm those behaviour ?

I have attached a patch file for what seems to work for me with History, Atom, CSV and PDF, I can't check mail notifications because I haven't got a working mail setup.

I have included a function to list attributes that are hidden on all statuses of all trackers on projects. This functions result can possibly be cached for efficiency...

The function is used to completely hide columns in the list of available columns and filters.

When revising the code please remember that I don't actually know Ruby, so have at it ;-)

Does the above patch (workflow_hidden_field_DR_v0.04.patch) need either (or both) of the previous patches? I can apply workflow_hidden_field_DR_v0.04.patch on it's own, but the hidden field option doesn't show up (patch doesn't appear to modify app/helpers/workflows_helper.rb to add a "hidden" option like the other two patches do, so I assume one or both is required).

If I try applying one or both of the previous patches I get a load of "Reversed (or previously applied) patch detected! Assume -R?". Some of the hunks apply, but enough if them don't to make me think I've got something very wrong.

I've tried this in both the stable-2.1 and master branches.

Sorry Jamie MacIsaac I'm affraid some of the original patch was missing in my patch :(I will try to upload a patch with all needed code ASAP, including a possible implementation for Mails which I've finally got running :-)

I've attached a new patch file with the missing code from the previous patch, and hidden fields in status mails.

Again, please be ruthless in the code review...

Also please note that a mail will be sent for each combination of hidden fields for the recipients, so in theory this could mean several mails need to be sent.

This could noticeably degrade performance for the end-user if mails are sent synchronously and the mail server is slow to respond.

There was a copy/pasta error in v0.05 I've attached a fixed patch.

+1

The patch (v0.06) applies cleanly to the current master branch (I'm using the GitHub official mirror, official SVN repo apparently not working just now?) but fails for me against current stable-2.1 (aka revision 10764, which is what the patches appear to be based on - see truncated output at bottom of post).

When I run against the master I can create issues (I even receive the email notification), but see an error (undefined local variable or method `position_name') when I try to display them (happens immediately after the issue is created).

Any idea what I'm doing wrong?

Patching against stable-2.1:
patching file app/models/mailer.rb
Hunk#3 FAILED at 66.
patching file app/models/query.rb
Hunk#2 FAILED at 365.

Hunk#5 FAILED at 927.

Jamie MacIsaac wrote:

The patch (v0.06) applies cleanly to the current master branch (I'm using the GitHub official mirror, official SVN repo apparently not working just now?) but fails for me against current stable-2.1 (aka revision 10764, which is what the patches appear to be based on - see truncated output at bottom of post).

When I run against the master I can create issues (I even receive the email notification), but see an error (undefined local variable or method `position_name') when I try to display them (happens immediately after the issue is created).

Any idea what I'm doing wrong?

Patching against stable-2.1:
patching file app/models/mailer.rb
Hunk#3 FAILED at 66.
patching file app/models/query.rb
Hunk#2 FAILED at 365.
Hunk#5 FAILED at 927.

Did you remember to upgrade the database? AKA: "rake db:migrate RAILS_ENV=production". Can you post the whole error?

David Robinson wrote:

Did you remember to upgrade the database? AKA: "rake db:migrate RAILS_ENV=production".

Nope. Running the migration cleared up the error - thanks!

One of my colleagues has been testing the patch. All seems OK now, except none of the emails sent by the test Redmine server include a 'To:' field in the message envelope (no emails at all, not just emails for projects with hidden fields). Any ideas?

Jamie MacIsaac wrote:

One of my colleagues has been testing the patch. All seems OK now, except none of the emails sent by the test Redmine server include a 'To:' field in the message envelope (no emails at all, not just emails for projects with hidden fields). Any ideas?

The only thing I can think of is the "Blind carbon copy recipients (bcc)" configuration (under settings, e-mail notifications).

Duplicated by Feature #5011

David - yep, the "Blind carbon copy recipients (bcc)" was indeed checked. Thanks for your help and the also for the patch - seems to be working fine.

Please release with 2.2.0 it will make "Mightful workflow field enhancement" complete! ;)

Hi,

We have been testing the latest patch for 2 weeks now on our Development Redmine environment and have not found any new issues.

Is there any chance this could be put into 2.2.0

We are very keen to use this in our live Redmine however we have reservations about upgrading to a development branch.

Thanks

John

I've applied the patch to my 2.1.4 installation, but it's now causing an error whenever I try to update an issue:

```
NoMethodError (undefined method notified_users' for #<Issue:0x7f32dfc8faa0>):  
app/models/journal_observer.rb:27:in after_create'  
app/models/journal.rb:47:in save'  
app/models/issue.rb:1269:in create_journal'  
app/models/issue.rb:903:in save_issue_with_child_records'  
app/models/issue.rb:890:in save_issue_with_child_records'  
app/controllers/issues_controller.rb:171:in update'
```

The "notified_users" method is totally missing from the Issue model. Any thoughts?
Thanks!

John Moody wrote:

The "notified_users" method is totally missing from the Issue model. Any thoughts? Thanks!

Hi there,

well this seems to be a defect installation.

Look at this: r10547

This Revision introduced the new method "notified_users". If this is totally missing in your model, your previous upgrades introduced this error. Please try to reupgrade your installation and try again.

Best regards,

Daniel

Hi, I have been trying to apply this patch to a clean 2.1.4 installation.

I am getting errors similar to John Moody above.

Sorry if I am stating the obvious, but this v0.06 patch seems to depend on the private comment branch (r10547 mentioned above from #1554) which seems to be targeted at the 2.2.0 release.

Do I need to upgrade to a later build than 2.1.4 to be able to apply this patch?

Hi, updated my installation to 2.2.0.stable.10995 and the patch works fine after manually fixing one issue with mailer.rb.

Ian

ian stoddart wrote:

Hi, updated my installation to 2.2.0.stable.10995 and the patch works fine after manually fixing one issue with mailer.rb.

Ian

Can you please patch the patch, for issue with mailer.rb? =)

I would appreciate it so much guys if you could implement this in the next release!

Yeah, I saw tons of ppl want this feature to be integrated!

And wrote:

Can you please patch the patch, for issue with mailer.rb? =)

Hi,

I'm not Ian, but I've "patched the patch" - workflow_hidden_field_DR_v0.07 applies cleanly to 2.2.0. The sole difference between v0.06 and 0.07 is on line 288 where "cc = issue.watcher_recipients - recipients" is replaced by "cc = journal.watcher_recipients - recipients" in app/models/mailer.rb. I'm pretty sure this is the issue ian stoddart refers to in 35 above.

oh yeah, love it ;)

h1. +1

Jamie MacIsaac wrote:

Hi,

I'm not Ian, but I've "patched the patch" - workflow_hidden_field_DR_v0.07 applies cleanly to 2.2.0. The sole difference between v0.06 and 0.07 is on line 288 where "cc = issue.watcher_recipients - recipients" is replaced by "cc = journal.watcher_recipients - recipients" in app/models/mailer.rb. I'm pretty sure this is the issue ian stoddart refers to in 35 above.

Hi Jamie,

thanks for providing this patch. I will test it tomorrow morning and give some quick feedback. Thanks for contribution!

Hi there,

this is working fine on r11165 for me.

In addition, I added a little patch for the german translation, if this patch is integrated.

Best regards,

Daniel

Hi And,

Sorry, I've not been watching this. I am not very familiar with generating a patch and didn't want to confuse matters by putting up a bad patch.

I believe the patch failed for me on 2.2.0 because the line numbers referenced in hunk # 3 in mailer.rb did not match and so the patcher could not work out where it should apply. All I did was manually find the correct lines of code and insert the hunk by hand.

Interestingly I am, now testing in more detail and have spotted that if I update a custom field which is hidden from some users, they still get an email informing that a change has happened, but the information they receive in the email shows no change. I would be happier if they did not get the email at all if the change affects a field they cannot see...

I am not sure if the issue Jamie has pointed out may or may not affect this?

hi,

i use redmine stack (redmine-2.1.5 and redmine-2.1.6) . try patch with workflow_hidden_field_DR_v0.07.patch. there is no other plugin.

.

..

patching file app/models/mailer.rb

Hunk #3 FAILED at 66.

..

patching file app/models/query.rb

Hunk #2 FAILED at 365.

Hunk #3 succeeded at 405 (offset -58 lines).

Hunk #4 succeeded at 434 (offset -58 lines).

Hunk #5 FAILED at 927.

2 out of 5 hunks FAILED -- saving rejects to file app/models/query.rb.rej

..

what is wrong ?

Best Regards,

ilgar

ilqar Caferov wrote:

i use redmine stack (redmine-2.1.5 and redmine-2.1.6) . try patch with workflow_hidden_field_DR_v0.07.patch. there is no other plugin.

This will be your problem. The patch is for the current trunk, not for Redmine 2.1.6.

The linenumbers are different. You must apply this patch manually, but I won't guarantee that this patch will work on 2.1.6 instantly.

Sorry if I am missing something obvious here, I just saw this thread and the "Related Issues" list is promising to have immediately :-)
. I was looking for some information on what all features included into this patch being disussed here. Are there any information on what is included in this patch, other than going through the code? (I am not familiar with Ruby)

Just to notify you all: I ´ve successfully patched the 2.2.1.stable version.

I ´m testing all the functionalities.

Good job!

Luiz, great news, keep us up-to-date please! Thanks for your work on that!

Hi,

I am testing using patch 0.07 on 2.2.1.stable version.

I have some custom fields marked in the workflow / fields permissions as completely hidden for some roles.

When I log in as a user with one of these roles, the custom fields are hidden in the issue as expected.

If I open the issues page and expand the options to view available columns the completely hidden custom fields are available for selection and when they are selected the relevant data is displayed in the issues table.

I assume this is not expected behavior?

It makes the patch much less usable for my application as it is.

Everything else about the patch seems to be working great and if this one point is resolved then the patch is just what I need!

Ok, now I understand. I had marked the custom field as hidden in all the states in use by the current issue type. The custom field was only used in one of the issue types.

For the custom field not to be displayed in the available columns list, the custom field needs to be hidden in all the states for all issue types.

I had missed that tickbox in the Workflow / Fields Permissions tab that hides statuses not in use for the current issue type.

ian stoddart wrote:

Ok, now I understand. I had marked the custom field as hidden in all the states in use by the current issue type. The custom field was only used in one of the issue types.

For the custom field not to be displayed in the available columns list, the custom field needs to be hidden in all the states for all issue types.

I had missed that tickbox in the Workflow / Fields Permissions tab that hides statuses not in use for the current issue type.

Thanks for the update Ian!

In the meantime while I was trying to reproduce the issue you mentioned, I have seen that there is an issue regarding this, columns, not values, showing, if you go to list issues without selecting a project first (so you're listing issues for all projects). I'm currently in the process of fixing this problem on our system, I will post a diff when done, if no one beats me to it.

I set this to a release candidate for the next major release. This could be a huge improvement for the workflow handling.

David Robinson wrote:

ian stoddart wrote:

Ok, now I understand. I had marked the custom field as hidden in all the states in use by the current issue type. The custom field was only used in one of the issue types.

For the custom field not to be displayed in the available columns list, the custom field needs to be hidden in all the states for all issue types.

I had missed that tickbox in the Workflow / Fields Permissions tab that hides statuses not in use for the current issue type.

Thanks for the update Ian!

In the meantime while I was trying to reproduce the issue you mentioned, I have seen that there is an issue regarding this, columns, not values, showing, if you go to list issues without selecting a project first (so you're listing issues for all projects). I'm currently in the process of fixing this problem on our system, I will post a diff when done, if no one beats me to it.

I have attached the above mentioned diff file.

I just want to mention again, that code review of any code I have submitted, is a very good idea, as I do not know Ruby nor the codebase, which means my code is very "hackish".

It seems I botched the diff file, it appears to be missing the changes from acts_as_watchable.rb

I have attached another diff file, hopefully with all the needed patches O_o

Hi,

I am hitting an error:

ActionView::Template::Error (undefined local variable or method `detail' for #<#:0x7f93a337a0d0>):

```
1: <% reply_links = authorize_for('issues', 'edit') -%>
2: <% for journal in journals %>
3:   <% if details_to_strings(journal.details).any? || journal.notes.blank? == false %>
4:
5:
6:
```

```
<%= link_to "#{journal.indice}", {:anchor => "note-#{journal.indice}"}, :class => "journal-link" %>
app/helpers/issues_helper.rb:244:in `details_to_strings'
app/helpers/issues_helper.rb:243:in `each'
app/helpers/issues_helper.rb:243:in `details_to_strings'
app/views/issues/_history.html.erb:3:in `_app_views_issues__history_html_erb__862122133_70136036374580'
app/views/issues/_history.html.erb:2:in `each'
app/views/issues/_history.html.erb:2:in `_app_views_issues__history_html_erb__862122133_70136036374580'
app/views/issues/show.html.erb:128:in `_app_views_issues_show_html_erb__381019951_70136035944920'
app/controllers/issues_controller.rb:119:in `show'
app/controllers/issues_controller.rb:116:in `show'
```

I think it is down to details_to_strings in issues_helper.rb:

```
# Returns the textual representation of a journal details
# as an array of strings
def details_to_strings(details, no_html=false, options={})
  ...
  ...
  values_by_field.each do |field_id, changes|
    unless detail.journal.issue.hidden_attribute?(detail.prop_key, options[:user])
      detail = JournalDetail.new(:property => 'cf', :prop_key => field_id)
      if changes[:added].any?
        detail.value = changes[:added]
        strings << show_detail(detail, no_html, options)
      elsif changes[:deleted].any?
        detail.old_value = changes[:deleted]
        strings << show_detail(detail, no_html, options)
      end
    end
  end
end
```

on the second line detail is used before it is defined on the third line??

I am trying to get my head round what it should actually say, so thought I should post it here in the mean time as it might take me a while to work it out!

Hi Ian

Looks like it should be

```
detail = JournalDetail.new(:property => 'cf', :prop_key => field_id)
unless detail.journal.issue.hidden_attribute?(detail.prop_key, options[:user])
  if changes[:added].any?
    detail.value = changes[:added]
    strings << show_detail(detail, no_html, options)
  elsif changes[:deleted].any?
    detail.old_value = changes[:deleted]
    strings << show_detail(detail, no_html, options)
  end
end
```

Where are you getting this? AKA: what did you do when you got the error?

Also more stuff, this time from pdf.rb, appears to be missing in my latest diff.

I have attached a diff, that includes both a fix for this error and the missing things from pdf.rb.

Hi David,

thanks for the quick update.

I was updating the data in some custom fields in an issue. A number of the custom fields are hidden from many of the users. Some of these users are set to get email updates on any project change.

Does that help?

Hi David,

sorry to report, if I make that change, I get an error when I try to display some particular tickets.

Started GET "/support/issues/300" for 10.xx.xx.xx at Tue Feb 12 14:20:20 +0000 2013

Processing by IssuesController#show as HTML

Parameters: {"id"=>"300"}

Current user: ian.stoddart (id=14)

Rendered issues/_action_menu.html.erb (6.4ms)

Rendered attachments/_links.html.erb (147.8ms)

Rendered issue_relations/_form.html.erb (2.8ms)

Rendered issues/_relations.html.erb (5.3ms)

Rendered issues/_history.html.erb (1412.1ms)

Rendered issues/show.html.erb within layouts/base (1667.6ms)

Completed 500 Internal Server Error in 1733ms

ActionView::Template::Error (undefined method `issue' for nil:NilClass):

1: <% reply_links = authorize_for('issues', 'edit') -%>

2: <% for journal in journals %>

3: <% if details_to_strings(journal.details).any? || journal.notes.blank? == false %>

4:

5:

6:

<%= link_to "#{journal.indice}", {:anchor => "note-#{journal.indice}"}, :class => "journal-link" %>

app/helpers/issues_helper.rb:245:in `details_to_strings'

app/helpers/issues_helper.rb:243:in `each'

app/helpers/issues_helper.rb:243:in `details_to_strings'

app/views/issues/_history.html.erb:3:in `_app_views_issues_history_html_erb__1110742084_70276145134480'

app/views/issues/_history.html.erb:2:in `each'

app/views/issues/_history.html.erb:2:in `_app_views_issues_history_html_erb__1110742084_70276145134480'

app/views/issues/show.html.erb:128:in `_app_views_issues_show_html_erb__910177246_70276146378560'

app/controllers/issues_controller.rb:119:in `show'

app/controllers/issues_controller.rb:116:in `show'

If I comment out the 'unless' line and a matching end the it seems to behave fine.

If I read the first half of the function correctly, the 'values_by_field' has already been using an 'unless' clause, so is it required to do it again in the second

Hi Ian

I don't know, it appears that somehow detail.journal is nil after detail = JournalDetail.new(:property => 'cf', :property => 'cf') on line 245.

This seems very "wrong" to my untrained eyes when you take the following into consideration from journal_detail.

```
class JournalDetail < ActiveRecord::Base
  belongs_to :journal
```

I can't reproduce this on my system.

Can you tell me if there is something "special" about the issues that exhibit this behaviour?

Maybe `detail.journal.issue` isn't the right way to get to the right issue object in this function? Maybe someone with more knowledge about the codebase can chime in if they see this :-)

Or maybe it's just that I'm complete crap at getting these diff files right :-(

It seems that 0.010 still has the error with using detail before assignment

I have, yet again, attached a, hopefully, more correct diff file...

Hi David,

I have been having a look and a think...

this only happens on a couple of issues.

The issues do have a large number of fields (almost 50), some fields are visible to all users (around 35), some fields are completely hidden from some of the users (around 15) (set as hidden in all the states for all issue types)

The changes that seem to have triggered the problem seem to be where I have made changes to issues where the only change was to fields which are completely hidden to some of the other users.

The thing I cannot understand is that I get the error if I log in as one of the users that cannot see the fields or as myself. When I log in as me I should be able to see these changes so I am not sure why the list would come back empty.

If I comment out the lines I mentioned then if I log in as me, I can see the complete history, if I log in as one of the limited users, then I just don't see the changes in the history at all. (this is what I would expect)

I will have more of a play and see if I can determine anything else about the particular issues.

This one should definitely find its way in 2.4.

I can't let people think that it will be added to 2.4.0. The patch has no test and I doubt that this feature will be added in a near future.

Why that? Only because of a missing test?

This feature would be a huge improvement for workflows!

Think about an environment where customers are allowed to write tickets but shouldn't be bothered with categories, assignee or start/due date.

Do you really want to tell them just to ignore these fields? The "hide" option would be a killer feature for the most of us involved in this thread...

Please, Jean-Philippe, give this a second chance!

Hi there,

I can understand both sides of the story.

On Jean-Philippe's side...

I can understand that missing tests would make much trouble. The integration server couldn't handle a proper check if any feature is broken after a bug fix or a new feature implementation. This is a real problem. This will slow down the development and could cause buggy releases. Which is something, no one of us wants to have in the future, as Redmine is a real stable software in comparison to other project management tools.

The other thing is, that this patch has many points, some with a bigger impact. The patch has to be tested par excellence. There shouldn't be any side-effects by this patch.

On User side...

There are many products, even paid products, which give the user many many abilities to perform ticket tasks. Many abilities to visualize the workflow, many abilities to relate, plan, time issues and so on. But most of them has just one lack. They are horrible to

maintain on the user side.

A project manager would love to have all this abilities. Maybe he could blind them out in his view. But a normal customer, who just adds an issue for the first time to the tracker, would be overwhelmed.

Jean-Philippe, just try to imagine (even if it's hard, as your quite long in the business, as the most us experienced PM's). You have bought a new coffee machine, everything is fine and ok. You used it every day. Maybe for two or three months. Then it's just wracked and you conduct the support to get a new one. The support aims you to the helpdesk (installed via Redmine, just for example). You see a mask, which holds up to 30 or 40 input fields. Some of them will be marked as required, but your not sure what they want in the input field labelled "QRS-Nr.". There are plenty of optional fields.

I'm sure, you just will fill all the required fields, as your not pretty sure, which of the optional fields is needed. After filling everything out, even the field QRS-Nr. - just in case you filled it with 1234, as it just accepts integer values and you don't know what they want and you won't call the rough support person again.

You send the request to the support. After two or three days (without coffee), you get the response.

"Your QRS-Nr. couldn't be found. Did you enter this Number? This Number is filled by the Helpdesk! More than that, you haven't filled the two optional fields [...].

We can't solve your request, until you've updated your request with the needed information!"

So... everything is "pretty". You go to the tracker, open the issue fields, type in everything needed and sent it to the support. The support fills all those fields after some days. You won't get any notification that your coffee machine is on the way to you. You just get some little hacky mails, which provides some strange information about different fields, such fields like the QRS-Nr.

These are all information, you just don't need. But you will be bothered with it.

Be honest, I'm sure you have such a situation before with some support. :-)

Well I have such a situation in my daily business with many issue tracker or with some project management software. Just as an example, Microsoft Project, It provides the user with many menus and each of them differ slightly, depending on which object the user wants to add. But each of them has many fields, I don't use. Just for example, if I just plan a small project for myself or just a project which doesn't need a huge resource plan. Every Dialog asks me, which resource should be used. Which blows up the menus. If those menu would be more intelligent, and just display the resource options, If I have activated the resource plan, each dialog would be smaller and easier to manage.

Another example, currently we're using a self-made issue tracker, which should be replaced by Redmine. We have just 10 fields which are relevant for our daily business. Three are needed for each request, some are needed for special cases and some others are needed for our internal planning. All of them are revealed to our internal customers. They see, that many of them are useless for their request (our internal fields) they tend to just ignore all of those fields and just write plain text, which slows down our support. We need to manually check every issue and fill out all the needed fields. Just because, that there are to many fields they don't understand/use.

I'm sure, we could get a better solution than that! :-)

My suggestion...

#We evaluate the side-effects of this patch better. This task need some help of the community.

#We provide tests, which provide a good integration. This task need some help of the community, as the help of Toshi or Jean-Philippe, as you both are one of the experienced developers in the team and always provide good test cases.

#Maybe there could be a integration branch for this feature, which runs parallel to the normal 2.4 branch and will be merged with 2.4 or 2.5 if this integration is as stable as it should be.

Maybe this could be a way to please both sides?

I would love to see a feature like this in Redmine. Although I don't think it is a "killer feature", but certainly highly useful for certain use cases.

That said, from an engineering point of view, I have to agree with Jean-Philippe (for what little this may be worth). But I don't read him as rejecting this patch, nor does he say it's not important. His argument seems rather quite neutral and rational to me.

After all, there is an unresolved comment by Ian here, reporting issues with the patch. The author himself has stressed that he is new to Ruby, and that the code should hence be carefully reviewed (actually, I would hope that careful code review is undertaken for any submission).

As desirable as this feature is, I don't think it's good to import an untested, unreviewed implementation. What if it introduces data corruption or if its design is broken (not that I have any indication for either)? Backing out a change is in general much harder than putting it in. A thorough set of tests would greatly help to alleviate these fears.

It was very disheartening to know this patch is out from 2.4. I think this was very important functionality.

However, instead of getting into yes vs. no debate - I would like to ask what are the preconditions which will allow this patch to finally go in? What work is missing and what more that community can do to get this patch to the trunk in the fastest possible way?

+80 people in our company =)

+1

Looking forward to this functionality

It seems, it doesn't work with 2.3 =(

Hey, Daniel told me it will be available with 2.3, but it seems 2.3 came out without it?

BTW, I've applied the last patch to 2.3 and everything works, except hidden column in issues list. Only issues_helper and query don't want to be patched. I've applied dirty JS hack to prevent the column shown. Temporary solution. Eager to see the feature in trunk.

This is not part of 2.3, if somebody told you differently, then they were mistaken resp. overeager. For details, please look at comment 62 (

<http://www.redmine.org/issues/12005#note-62>) and the discussion following it.

Hi,

We are already intersted by this feature in my company.

Another duplicate #13729

Just imagine how many people need this feature!! Please can we schedule this?

+1

I need to hide (disable?) watchers and file attachments

I completely agree with Jean-Philippe about everything (apart maybe from prioritizing this feature ;-), what I've made is a hack and should be treated as such.

I would love to do everything required to merge this feature, unfortunately I don't have the needed skills, it was a stretch for my non-existent Ruby knowledge to make the attached hack. I won't be able to acquire those skills in any reasonable time-frame either. It is clear that if the Redmine community want this feature "soon" they will have to lift this burden themselves, someone from the community with the skills needed, will have to start doing this feature properly, so the Redmine team has an honest chance of merging a community created feature.

Either that or wait until the Redmine team finishes other, more important tasks, and get around to making this feature.

As Daniel quite clearly put it we as a community have the following tasks if we want this integrated soon:

#We evaluate the side-effects of this patch better. This task need some help of the community.

#We provide tests, which provide a good integration. This task need some help of the community, as the help of Toshi or Jean-Philippe, as you both are one of the experienced developers in the team and always provide good test cases.

If anyone with the needed skills are out there (Redmine System knowledge and/or Ruby knowledge and/or Test generation skills), and willing to work on this, please let yourself be known :-)

will the patch allow to hide the default "status" field?

Jai prakash wrote:

will the patch allow to hide the default "status" field?

If you are referring to the attached "workflow_hidden_field_DR_v0.011.patch", the answer is no.

@David

Yes I was referring to the patch.

Daniel Felix wrote:

Jean-Philippe, just try to imagine (even if it's hard, as your quite long in the business, as the most us experienced PM's). You have bought a new coffee machine, everything is fine and ok. You used it every day. Maybe for two or three months. Then it's just wracked and you conduct the support to get a new one. The support aims you to the helpdesk (installed via Redmine, just for example). You see a mask, which holds up to 30 or 40 input fields. Some of them will be marked as required, but your not sure what they want in the input field labelled "QRS-Nr.". There are plenty of optional fields.

You can actually make these fields read-only to the customer and they won't show up in the issue form.

I understand that this feature might be desirable but having to handle visibility per tracker and status is overcomplex IMHO and the patch does not seem to do it perfectly. Being able to set custom field visibility per role would be a nice improvement and a much more reasonable change to add in a next release. I'm reopening #5037 for 2.4.0, any feedback is welcome.

I think, if there is possible to set read-only permission, then it should be also possible to set visibility also - it's logical: if you can't change a field, then maybe you also shouldn't see it sometimes.

But role-based visibility is enough for development process in our company now. So, I'm totally for the improvement in the next release =)

Jean-Philippe Lang wrote:

I understand that this feature might be desirable but having to handle visibility per tracker and status is overcomplex IMHO and the patch does not seem to do it perfectly. Being able to set custom field visibility per role would be a nice improvement and a much more reasonable change to add in a next release. I'm reopening #5037 for 2.4.0, any feedback is welcome.

Sounds very reasonable, and cool. I for one am all for closing this in favor of #5037.

Meanwhile #5037 has been resolved. But this only handles "custom fields" and not the "standard fields". So this is not at all what at least I need, because we don't use that much custom fields anyway. We still need a way to hide or make "read-only" standard fields for certain roles (i.e most commonly estimated time, target version, priority, assignee ...).

+++++++111

We use the estimated time field a lot and we don't want our customers to see the estimated time field because they will start making date calculations based on those estimates they see and that would be a huge problem to our planification.

Also, if we estimate that some task will take 8 hours but we charge the customer 10hs, say we use those 2hs for testing, we may let the customer know we are charging extra hours for testing but that's not always possible cause on low-budget projects the customer may respond that they will do the testing themselves so we don't charge them 2 extra hours, and that's not acceptable at all, I mean, not having time to test our work cause it would reduce our output's quality dramatically.

So, if the customer sees that we estimated it in 8hs and we charged them for 10 that would cause a instant complaint from to customer to our sales team.

You may say we can use a custom field for estimates but that won't work with any plugin that uses the estimated hours field to output reports.

Please reconsider adding this feature, it not only happens with that field I mentioned, due date is another controversial field we need to hide from the customer, the deliver plan is an entire different document (buffered) than the real estimation and internal delivery of the finished tasks.

Showing everything leaves us with too little room to play with the dates.

Thanks!

Santiago Burgues wrote:

+111

We use the estimated time field a lot and we don't want our customers to see the estimated time field because they will start making date calculations based on those estimates they see and that would be a huge problem to our planification.

I agree 100%!

Also, if we estimate that some task will take 8 hours but we charge the customer 10hs, say we use those 2hs for testing, we may let the customer know we are charging extra hours for testing but that's not always possible cause on low-budget projects the customer may respond that they will do the testing themselves so we don't charge them 2 extra hours, and that's not acceptable at all, I mean, not having time to test our work cause it would reduce our output's quality dramatically.

I agree to that sometimes customers cab't be convinced to accept taht testing is part of the job to make good products. However why do you don't just enter 10 hours because the estimation is not only for coding but anlyssis, sepcification, planning, coding testing <>bug fixing. Its an estimation for the whole buncg to bring th idea into production not only coding. Don't publish every steps estimation to ur customer. He is only interested in the final date and the final cost to get his feature on air.

So, if the customer sees that we estimated it in 8hs and we charged them for 10 that would cause a instant complaint from to customer to our sales team.

Don't confuse profit with costs, don't use this estimation field for sales.Either you have an contact with ur customer ou ged paid per effort, so you just have to mulitply your spent time with ur hourl rate. Or you have a contract you get paid a fixed price for the outcome , which is mostly a release/version and is bargained after an estimation and pallning phase for the release. This will alos include possible risks and profits you wanna make with the job.

+1 for hiding time related fields!

This is a show-stopping issue for client access.

+1

This is a show-stopping issue for client access.

Totally agreed. Or for corporate clients with mandates to only let people see what they need to see to get the job done.

- +1
- +1
- +1
- +1
- +1

in Redmine 2.3.3, i had an error (undefine method reject! in query.rb so, i changed @available_columns.reject! to available_columns.reject! and it works fine only in show information. but in edit or mail, we still find the hidden fields.

Please have a look at version 2.4 which resolved issue #5037 ["Role-based issue custom field visibility"](#)

In 2.4 you are now able to hide certain fields based on a role...

+1

+1

+1 waiting to be released

+1

+1 I used Redmine with client access in mind and realizing this feature is not here yet was a big let down :(.

+1

+1

I would like to know this feature will comes with which version? As I can't see it in the roadmap.

It can not at least because patch is not enough complete/clean/tested (See JP's arguments above) and also because JP considers it will give too much complexity to Redmine's administration.

I have merged it into my Redmine 2.4, will try to publish the patch but as of now it is combined with other customizations of mine not related.

That said, combined with Feature #14162, it is very powerful !

+1

Gurvan Le Dromaguet wrote:

It can not at least because patch is not enough complete/clean/tested (See JP's arguments above) and also because JP considers it will give too much complexity to Redmine's administration.

I have merged it into my Redmine 2.4, will try to publish the patch but as of now it is combined with other customizations of mine not related.

That said, combined with Feature #14162, it is very powerful !

Is it possible to implement this feature as a plugin? This would solve the problem that Jean-Philippe Lang doesn't want this feature in Redmine itself.

André Bachmann wrote:

Is it possible to implement this feature as a plugin? This would solve the problem that Jean-Philippe Lang doesn't want this feature in Redmine itself.

It would be the best surely. I have never done a plugin yet and not expert at ruby/rails. I will 1st publish the 2.4 patch, and maybe with the help of the community - who has already turned my initial patch into something more consistent - we will get this done :)

http://www.redmine.org/plugins/redmine_hide_estimated_hours there is also this plugin as a workaround for some use cases. Seems to work well.

Łukasz Zaroda wrote:

http://www.redmine.org/plugins/redmine_hide_estimated_hours there is also this plugin as a workaround for some use cases. Seems to work well.

Looks like working "role based" not "workflow based". Then this is implemented in 2.4 for Custom fields, so this plugin is a "complement" of #11963, but not what we try to implement here.

For anyone using the attached patch, derivatives thereof, or intending to, I just became aware that "priority" hiding doesn't work. All occurrences of "issue.hidden_attribute?('priority')" should be replaced with "issue.hidden_attribute?('priority_id')" for that to work.

Gurvan Le Dromaguet wrote:

I have merged it into my Redmine 2.4, will try to publish the patch but as of now it is combined with other customizations of mine not related.

That would be awesome. We're currently limited to running old versions of Redmine, because we haven't been able to afford the resources needed to update the patch. So this would be a big help, at least for us!

Did it fast and not tested, but this is a start, attached patch for redmine 2.4.2

Hiding of standard fields based on role and tracker is a must for anyone who wants to provide client access to Redmine.

I created a customer specific role & user and edited their permissions (role and workflow) to hide time related fields. The correct behavior is exhibited for this user under this role creating/updating issues - no time related fields are shown. However when viewing the issue as this user the start date, due date, % done and estimated time is visible, as is the history related to this. Additionally, the role constraints do not flow through under issues -> options -> columns list: the due date, estimated time and % done are visible, can be selected & added as columns onto the issue list although the user's role disables all time tracking

There have been so many similar requests (mine is Issue #16095). Is the demand for this feature not enough to be considered as core to the product ?

It is very frustrating that this is only issue preventing us from providing role based customer access to Redmine.

+1 This would be very good addition

I tried the patch for 2.4.2 but i get this error when applying it?

Hunk #1 FAILED at 25.

1 out of 1 hunk FAILED -- saving rejects to file app/helpers/workflows_helper.rb.rej

patching file app/models/project.rb

patching file app/models/workflow_permission.rb

Unreversed patch detected! Ignore -R? [n] y

patching file app/models/issue.rb

Unreversed patch detected! Ignore -R? [n] y

Hunk #1 succeeded at 503 (offset 20 lines).

Hunk #2 succeeded at 544 (offset 20 lines).

patching file app/views/mailer/_issue.text.erb

patching file app/views/mailer/_issue.html.erb

patching file app/views/issues/show.html.erb

Hunk #1 FAILED at 33.

1 out of 1 hunk FAILED -- saving rejects to file app/views/issues/show.html.erb.rej
patching file config/locales/en.yml
Hunk #1 succeeded at 889 (offset 1 line).

patching file lib/redmine/export/pdf.rb

Oscar Corsvall wrote:

I tried the patch for 2.4.2 but i get this error when applying it?

Thanks for trying ... and sorry for the issue, I will have a try and come back

Hi!

Is there any way to remove that patch?

It seems to create issues when i try to create or update a issue.

I get "Internal error"

And log says:

Completed 500 Internal Server Error in 182.2ms

ActionView::Template::Error (undefined local variable or method `user' for #<#[Class:0x0000000712d408:0x00000007139a50](#)>):

2: <%= issue_url %>

3:

4: * <%=I(:field_author)%>: <%= issue.author %>

5: <%= unless issue.hidden_attribute?('status', user) %>

6: * <%=I(:field_status)%>: <%= issue.status %>

7: <%= end %>

8: <%= unless issue.hidden_attribute?('priority', user) %>

Oscar Corsvall wrote:

Hi!

Is there any way to remove that patch?

svn revert if you do not have other modifications.

Oscar Corsvall wrote:

Hi!

Is there any way to remove that patch?

If you aren't using any SCM:

If you don't have any other changes in those files, you could just download them again, and overwrite.

I've attached a patch for v2.4.4 (r12953) that appears to work on my system. It has only been very lightly tested, so beware.

To remove changes introduced by a patch you can revert a patch!

To reverse the patch, use the patch command with the -R option:

```
patch -p1 -R < path/file.patch
```

(If your patch was applied with the -p0 option, use that instead.)

Anyone tested it on redmine 2.5.0?

I have gotten around to a bit more testing, and Line 843 must be:

```
recipient_groups[variations.index(hidden_attribute_names(user) + (custom_field_values - visible_custom_field_values(user)))] << us  
er
```

otherwise you cannot create or edit issues.

Is there anyone around with experience with plugins that might be able to spare some time to try to convert this to a plugin. I can lend any desired assistance.

+1 for this feature

how i use path file for edit ???

Doan Phuoc wrote:

how i use path file for edit ???

I assume you meant "patch" not "path"..? It's in the [\[\[Patch|Redmine wiki pages\]\]](#).

The above patch workflow_hidden_fields_redmine2.4.4_v1.01.patch did not completely for me with the following specs:

```
@Redmine version      2.5.1.stable  
Ruby version          2.0.0-p481 (2014-05-08) [x86_64-linux]  
Rails version         3.2.17  
Environment           production  
Database adapter      Mysql2@
```

Here are the steps I took to apply:

```
[root@redmine redmine-2.5.1]# patch -p0 < workflow_hidden_fields_redmine2.4.4_v1.01.patch  
(Stripping trailing CRs from patch.)  
patching file app/helpers/issues_helper.rb  
Hunk #1 succeeded at 241 (offset 7 lines).  
Hunk #2 succeeded at 248 with fuzz 2 (offset 7 lines).  
Hunk #3 succeeded at 268 (offset 7 lines).  
(Stripping trailing CRs from patch.)  
patching file app/helpers/workflows_helper.rb  
(Stripping trailing CRs from patch.)  
patching file app/models/issue.rb  
(Stripping trailing CRs from patch.)  
patching file app/models/issue_query.rb  
(Stripping trailing CRs from patch.)  
patching file app/models/journal.rb  
(Stripping trailing CRs from patch.)  
patching file app/models/project.rb  
(Stripping trailing CRs from patch.)  
patching file app/models/query.rb  
Hunk #2 succeeded at 342 (offset 2 lines).  
Hunk #3 FAILED at 776.  
1 out of 3 hunks FAILED -- saving rejects to file app/models/query.rb.rej  
(Stripping trailing CRs from patch.)  
patching file app/models/workflow_permission.rb  
(Stripping trailing CRs from patch.)  
patching file app/views/issues/_history.html.erb  
(Stripping trailing CRs from patch.)  
patching file app/views/issues/show.html.erb  
(Stripping trailing CRs from patch.)  
patching file app/views/mailer/issue_edit.html.erb  
(Stripping trailing CRs from patch.)  
patching file app/views/mailer/issue_edit.text.erb  
patch unexpectedly ends in middle of line
```

```
[root@redmine redmine-2.5.1]# /etc/init.d/httpd restart
Stopping httpd: [ OK ]
Starting httpd: [Thu Jul 24 11:38:19 2014] [warn] NameVirtualHost *:80 has no VirtualHosts

After application, the Issues page would give an internal server error. Here is the information from the log:

Started GET "/issues?page=7&set_filter=1&sort=tracker%253Adesc%252Cstatus%252Csubject%253Adesc" for 66.249.67.250 at
2014-07-24 13:58:40 -0400
Processing by IssuesController#index as */*
  Parameters: {"page"=>"7", "set_filter"=>"1", "sort"=>"tracker%3Adesc%2Cstatus%2Csubject%3Adesc"}
  Current user: anonymous
Completed 500 Internal Server Error in 115.4ms

NameError (undefined local variable or method `hidden_fields' for #):
  app/models/query.rb:346:in `available_filters'
  app/models/query.rb:229:in `build_from_params'
  app/models/issue_query.rb:119:in `build_from_params'
  app/helpers/queries_helper.rb:175:in `retrieve_query'
  app/controllers/issues_controller.rb:56:in `index'
```

Still having trouble with this. Any help is appreciated.

Michael Sanders wrote:

Still having trouble with this. Any help is appreciated.

One thing that may make a difference to you....

Michael Sanders wrote:

The above patch workflow_hidden_fields_redmine 2.4.4 _v1.01.patch did not completely for me with the following specs:
@Redmine version 2.5.1.stable
...

I don't have a 2.4.4 setup available to test with, but I have a feeling that's why the patch isn't working for you.

Hello everybody,

I made a plugin for this based on the previous patches. Seems to work well under 2.5.2.
Git repo: https://github.com/alexwais/redmine_workflow_hidden_fields

Feel free to contribute.

Alex Wais wrote:

I made a plugin for this based on the previous patches. Seems to work well under 2.5.2.
Git repo: https://github.com/alexwais/redmine_workflow_hidden_fields

Hi Alex,
Do you have any plans about porting this plugin to latest versions of Redmine?
I have just upgraded to 2.6.1, and it does not work,
I am getting error:

Web application could not be started

```
cannot load such file -- tcpdf (LoadError)
/usr/local/lib/ruby/gems/1.9.1/gems/activesupport-3.2.21/lib/active_support/dependencies.rb:251:in `require'
/usr/local/lib/ruby/gems/1.9.1/gems/activesupport-3.2.21/lib/active_support/dependencies.rb:251:in `block in require'
```

/usr/local/lib/ruby/gems/1.9.1/gems/activesupport-3.2.21/lib/active_support/dependencies.rb:236:in `load_dependency'

....

Maybe there is a quick fix that I could apply by myself?

Regards,
Tomek

@Everyone: please don't start discussions about referenced plugins in this issue. Keep those in the plugin author's issues. This issue is about a Redmine core feature request and should be used to discuss how a proposed change like requested could be specified, designed and finally implemented in a clean, clear, resource-efficient, tested and maintainable way.

Tomasz O wrote:

Alex Wais wrote:

I made a plugin for this based on the previous patches. Seems to work well under 2.5.2.

Git repo: https://github.com/alexwais/redmine_workflow_hidden_fields

[...]

I have just upgraded to 2.6.1, and it does not work,

[...]

Maybe there is a quick fix that I could apply by myself?

@Tomasz: now that I'm already responding, take a look at the "dedicated issue in the plugin's GitHub repo":

https://github.com/alexwais/redmine_workflow_hidden_fields/issues/5 and the therein referenced "commit on a fork of the plugin": https://github.com/letic/redmine_workflow_hidden_fields/commit/1e6d33edfde0f361d671209ce7970557e32dd3e9.

This seems to fix compatibility with 2.6.x.

Disclaimer: I haven't used nor tested this plugin and the referenced patches.

+1! need it from the box!

+1

+1

When will this feature be available for redmine 3 ?

+1 !

+1 here too... this would be incredibly useful!

+1

+1

+1

It may help: I have just created a very simple plugin which lets you hide core fields, based on roles.

More details here:

http://www.redmine.org/plugins/redmine_customize_core_fields

Hello Vincent, functionality is very interesting, but I can not access the settings , gives me " page not found" when I access http://miredminehost/core_fields. can you help mi

Vincent Robert wrote:

It may help: I have just created a very simple plugin which lets you hide core fields, based on roles.

More details here:

http://www.redmine.org/plugins/redmine_customize_core_fields

+1

It is core functionality that makes redmine more functional for serious redmine installations with mixed external and internal redmine users working on same projects.

Is this feature planned in next redmine versions?

+1

+1

+1

I'm looking forward to see this realized in a future Redmine version.

+1 That would be great!

+1

+1

+1

This task is actually very simple for custom fields.

It may be that I have never fully understood the request but in my eyes the possibility of hiding a custom field in the views is very simple since the function that does it already exists during custom field editability calculations.

So a few months ago I wrote a simple patch with a custom field setting flag "Hide if read only in workflow" that hide that field in the show if in the workflow is declared as read_only.

I did it for Issues but it shouldn't be difficult to extend it to other entities custom fields.

Thanks

Lorenzo

Yeah, the logic is quite simple for custom fields, unfortunately non-custom ("core") fields are treated very differently in many parts of the code.

I cannot say why this is so, but it is.

+1

Any chance somebody on the Core team (Go Maeda? Marius Balteanu?) might look at this?

The fact that "Core" fields and Custom-fields are treated so differently makes maintaining a plugin for this functionality pretty complicated.

It would be really nice if the base system simply treated core fields on par with custom-fields, allowing plugin authors to customize core fields just like custom fields.

Right now all views that use core fields have to be specifically patched for every new release if you make a plugin that touches core fields.

Maybe it would be better if I create a separate issue for "unifying" core/custom field logic?

related_issues
relates,Closed,8050,Mightful workflow field enhancement: visible, read only and mandatory
relates,Closed,5011,Limit Visible Issue Fields Based on Permissions
relates,New,4707,private fields
relates,Closed,9736,custom field visibility based on tracker
relates,Closed,5037,Role-based issue custom field visibility
relates,New,8313,Restrict Assignee List by Role
relates,Closed,9421,Prevent attribute changes once an issue is assigned to a closed version
relates,Closed,8162,Issue custom field visibility per workflow
relates,New,2618,Move some change properties fields into the "more" link
relates,Closed,10140,Allow certain fields in an issue to be designated as uncommon for display purposes
relates,New,10162,require notes on specific fields getting set to specific values
relates,Closed,12954,Required field can't be read-only
relates,Closed,2979,New issue : choose displayed fields by profile
relates,New,2539,New project setting: mandatory/optional configuration for target version issue-attribute
relates,Closed,4670,Fine tune "New Issue"
relates,New,11853,Field permissions
relates,New,4485,Create an ACL privilege for assigning tickets
relates,New,15782,Workflow Field Permissions - Status-Based Field Visibility
relates,Closed,15918,Managing "description" field
relates,New,3088,Estimated hours field able to hide role based
duplicates,Closed,12249,Configurable required fields per tracker/role with 'Not visible'
duplicates,Closed,11329,Private custom fields
duplicates,Closed,8302,Wrong columns displayed on Issue list
duplicates,Closed,13729,Field permission invisible
duplicates,Closed,12745,Disable the default "Description/Status/Priority" field
duplicates,Closed,11963,Hide the estimated times on issues for certain roles
duplicates,Closed,14162,Hidden Fields - Role Based
duplicates,Closed,11490,Permissions for roles to see fields per tracker/status

履歴

#1 - 2022/05/10 17:15 - Admin Redmine
- カテゴリ を Issues workflow_41 にセット