

Inheritable issue categories

2022/05/09 16:22 - Admin Redmine

ステータス:	New	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Issues_2	予定工数:	0.00時間
対象バージョン:		作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/11898	status_id:	1
category_id:	2	tracker_id:	3
version_id:	0	plus1:	9
issue_org_id:	11898	affected_version:	
author_id:	61572	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	45		
<div>説明</div> <p>This patch is related to issue #5358 and is written against rev 10433</p> <p>h2. Functional:</p> <p>A project may recursively inherit categories from its parent. There is a chain of inheritance that stops at first ancestor that does not inherit.</p> <p>Every issue can be assigned any issue category from its project or from inherited projects.</p> <p>When an inherited category (containing issues) would be lost either by explicitly rejecting inheritance or by a change in hierarchy, a new issue category is created in the project that is being modified and all issues in sub-projects are automatically re-affected.</p> <p>When explicitly destroying an issue category that has issues assigned to in the project or any sub-project, related issues can be re-assigned to any category form the project or any inherited category.</p> <p>h3. Limitations :</p> <p>Global issue categories are not implemented.</p> <p>h3. Regressions :</p> <p>None : if inheritance is not used, nothing is changed.</p> <p>h2. Implementation :</p> <p>A new attribute "inherit" is created in project along with a new method inherited-categories</p> <p>The list of inherited project is found from Project.self_and_ancestors.</p> <p>The overhead of searching inherited categories in current use cases (assigning a category to an issue) should not be visible.</p> <p>The overhead for less common use cases - changing parent, destroying a category might be more important</p> <p>h3. Tests :</p> <p>All tests pass without failure or error - every new method is tested by a new test</p> <p>Currently, new tests have not modified fixtures and use init methods.</p> <p>h3. Conversion to plugin :</p> <p>It would be possible to generate a plugin from this patch, but as it modifies core Redmine methods and views, I cannot write one that would be portable across different Redmine versions.</p> <div>workaround</div> <div>use redmine-org-5358-patch</div> <div>journals</div> <div>+1</div>			

Good functionality.

Also: jaw dropped at how nicely this ticket is filled. :)

+1

+1

+1 - It would be wonderful to get this integrated into trunk.

+1

I have applied this patch to version 2.2.1 stable

It shows ok, but when I click in the project inheritance and open the project settings I see the inherited category field, in the given subproject, but I am unable to modify the project, I get the following error:

Can you help me?

Also I would like to ask how do I run the tests generated?

Started GET "/redmine/projects/-eladat-elrendel-s/settings" for 10.0.2.172 at Mon Jan 28 22:13:17 +0100 2013

Processing by ProjectsController#settings as HTML

Parameters: {"id"=>"-eladat-elrendel-s"}

Current user: admin (id=1)

Rendered projects/\_form.html.erb (22.5ms)

Rendered projects/\_edit.html.erb (23.6ms)

Rendered projects/settings/\_modules.html.erb (2.5ms)

Rendered projects/settings/\_members.html.erb (77.5ms)

Rendered projects/settings/\_versions.html.erb (15.7ms)

Rendered projects/settings/\_issue\_categories.html.erb (121.3ms)

Rendered common/\_tabs.html.erb (243.7ms)

Rendered projects/settings.html.erb within layouts/base (244.2ms)

Completed 500 Internal Server Error in 259ms

ActionView::Template::Error (undefined method format\_project' for #<#<Class:0x7f4d21745930>:0x7f4d21172760>):

40: <% for p in projs %>

41: <% for c in p.issue\_categories %>

42: <tr>

43: <td><%= format\_project(p, projs) %></td>

44: <td><%= c.name %>

45: <% if ! inherited.include?(c) %>\*<% masked = true %><% end %>

46: </td>

app/views/projects/settings/\_issue\_categories.html.erb:43:in app\_views\_projects\_settingsissue\_categories\_html\_erb\_1443391838\_69984626288240'

app/views/projects/settings/\_issue\_categories.html.erb:41:in `\_app\_views\_projects\_settingsissue\_categories\_html\_erb\_1443391838\_69984626288240'

app/views/projects/settings/\_issue\_categories.html.erb:40:in each'

app/views/projects/settings/\_issue\_categories.html.erb:40:in app\_views\_projects\_settingsissue\_categories\_html\_erb\_1443391838\_69984626288240'

app/views/common/\_tabs.html.erb:24:in `\_app\_views\_common\_tabs\_html\_erb\_1345646726\_69984623227140'

app/views/common/\_tabs.html.erb:23:in each'

app/views/common/\_tabs.html.erb:23:in app\_views\_common\_tabs\_html\_erb\_1345646726\_69984623227140'

app/helpers/application\_helper.rb:263:in render\_tabs'

app/views/projects/settings.html.erb:3:in app\_views\_projects\_settings\_html\_erb1349938208\_69984623258840'

Well, I am sorry for you, but it is not really a surprise : I wrote that patch 4 months ago against a pre-2.1 version from trunk. I said I didn't want to make a plugin from it because I had to modify some core redmine files and compatibility issues were expected in new versions.

I don't really know how such a patch could be inserted in trunk to ensure it will be available in any future version.

You said you tried it against 2.2.1, but I noticed that current stable is 2.2.2. Are you sure you really prefer 2.2.1 ? I still remember how I worked, and I think I could port the patch against one of those versions without too much work. I spend some times on it and I'll be glad if it could be useful to somebody.

I won't have much time next week (I'll go to ski and won't have any access to my dev box ...) so I can't promise it before half February, but if you confirm the version you want to use, I'll port the patch against it.

Folks keep doing patches and Redmine keeps on changing versions and breaking them - is there some way to get the devs attention and just integrate this feature into core? or has it been rejected from core Redmine for some reason?

Well, I do agree with you : it would be better if the patch could be included into core. But I don't know the right way to do so. All I found in the wiki page on "submission guidelines" was rules to generate the patch.

Thanks a lot. We are on 2.1.1. because 2 weeks ago this was the latest stable release.  
I'd upgrade to 2.2. but I see that there are any new or fixed issues.

We have installed Redmine 2 weeks ago, migrating from mantis - I have also a lot of SQL scripts for a better migration.  
I have seen several problems around listing tree hierarchies in the system, with sorted orders and so on which can lead to problems.  
These orders can be done using recursive SQL-s.

I did not develop ruby on rails applications yet, but I am open to join the development team, to improve this product because it is good. I have strong database knowledge, and 10+year development experience (C++, J2EE)

I finally found some time for Redmine (sorry but holidays and Valentine's day were most important to me :-).  
In my first patch a view used a helper function declared in an other view's helper. It worked fine in previous versions (of rails) but doesn't work any longer in current versions of Redmine.

Here is a new version of the patch wrote against 2.2.1-stable, but I think it should work with any sub-version of 2.2-stable branch and maybe others.

:) Thanks a lot!

I just applied the patch on a 2.2.3 installation which looks fine except one issue. Changing the project of multiple issues via bulk edit works fine but the category information is lost. Would you be able to fix this problem?

Updated patch for 2.2.3 resulting of my port is attached.

I found where the problem is, and I think I should release a new version of the patch in a couple of days or not much later.

I read your updated patch but didn't clearly identified if you added something you needed to keep. Would you mind if I start back from my own port to 2.2.stable ?

Hi Serge, no, I didn't add anything. When I tried to apply the patch you provided on stable2.2 (which was already 2.2.3) the patch utility didn't accept it since there were already other changes in the same file coming from 2.2.3. Thanks for taking this up & regards, Ruedi.

@ Ruedi,

Here's a new version of my patch fixing the problem of bulk editing - thank for testing it in your environment.

@Samdar

This patch should work fine against 2.2.1 and against 2.2.0 to 2.2.2 - you'd better use this new version as soon as Ruedi confirms he can use it.

I found myself in a problem I'm not sure how to solve : the patch contains a migration. As I wrote it 5 months ago, the migration is older than 3 other. Should I rename it at each patch or is it enough to notice the fact ?

---EDIT---

DON'T USE THIS PATCH : IT'S BROKEN but I don't know how to remove it

Please see below.

Hi Serge, just tested the patch but unfortunately the category has changed after moving project via bulk edit is set to the first value in the enumeration. Somehow the original category is lost.

Can you please try to reproduce?

Hi Ruedi, I could reproduce and fix ...

@#! was stupid ... wrote 1 '=' when 2 were expected ... wonder what I'd drunk ...

Well I also fixed the tests that I wrote with only one category ... and of course they passed ...

I hope this new version should work fine.

Concerning the migration, there's no problem : I could verify that rails finds the migration even if it's not the last.

Leo Shklovskii wrote:

Folks keep doing patches and Redmine keeps on changing versions and breaking them - is there some way to get the devs attention and just integrate this feature into core? or has it been rejected from core Redmine for some reason?

Trying with a bigger one now ;-)

h1. +1

Thanks so much for the patch, but our IT support won't accept patches - so it would be great to just merge this into core (assuming no objection from core devs)

---

All I can do for merging the patch into core is porting it against current trunk - and I'll do it as soon as I'll be asked to. Maybe it would be good to have some feedback on the feature. Will your IT support accept plugins ?

I think I could convert the patch into a plugin but of course without any hope it will be portable across different redmine versions.

Very useful.

Hi Serge, eventually I managed to test the patch. Now it works fine and I also validated against 2.2.4-stable. Thanks for the efficient support, Ruedi

+1

Hi,

I use stable-2.3.2 and patched all files manually. It seems to work fine.

I don't know how to program ruby and I never used patch and svn before, please excuse my questions. I used "svn co <http://svn.redmine.org/redmine/branches/2.3-stable> redmine-2.3" to get the latest files and copied the patched files (in /tmp to make a diff-file). How can I make a patchfile (like the other patchfiles in this ticket)? If I use "svn diff > file.patch" the generated file doesn't look like the other patches, and it misses the new generated files in "test".

As this is a very interesting feature is it possible to get it in the trunk (so at the next version I don't have to do the work again)?

I think you should rename the new field in the database (20120830171625\_add\_inherit\_to\_project.rb) from "inherit" to "inherit\_cat", because in 2.3 it is possible to inherit members, and "inherit" is not specific enough (I think).

Best regards,

Wolfram

Hi Wolfram,

I do agree with your last remark and I will upgrade my patch to rename inherit to inherit\_cat. I think I could do it next week.

I'm pleased to know that the patch works against 2.3-stable (maybe it is not so bad ...).

For the format of patches, as I'm not a core developer of Redmine, I can't commit to the official svn depot and I just cloned the github copy - that's the reason why my patches are git's format and not svn's one.

You normally apply the patches using "patch -p 1".

Regards,

Dear Serge,

thank you for your answer. This helped me very much.

I attach the svn-patch with the new files for 2.3.2-stable (rev 12041, but should also work with 12032 as the difference between 12032 and 12041 are only es.yml and ru.yml locales). I think this patch should be used with "patch -p 0".

What must we do to integrate this patch in the trunk by a core developer? If needed I can make a patch for the trunk-files (and rename "inherit" to "inherit\_cat"). But I can't do the tests (rake test) as I don't know enough about rake, ruby, ... (and I don't have a test database).

I think many people would like this feature (see issue #5358)

Best regards,

Wolfram

Hi Serge,

I found a little bug: if I make a new project as a subproject, and select "inherit categories" (I didn't save the subproject before), and save the new subproject, the selection "inherit categories" is not marked; the selection is not saved. But the selection of "inherit users" is saved. If I set "inherit categories" after I saved a new project and save the change, the selection will be saved. Only when I make a new subproject with inherited categories the selection to inherit categories will not be saved (at creation of the subproject).

Best regards,

Wolfram

Hi,

I found the bug and could fix it.

I also followed your advice and renamed inherit to inherit\_cat.

I found simpler to start back from my own patch, by if you changed something I will integrate it in next version I you ask for it.

So here is a new version of my patch against release 2.3.2, but it should work against any 2.3-stable.

For what it's worth, I applied the patch added by Serge Ballesta to SVN commit 13138 on the 2.5-stable branch. I will attach the SVN diff for that version. I had to manually change the issue\_category.rb file to apply the patch, for the rest it mostly worked right away.

I also added Dutch translations for the additional strings.

Thanks for the great effort. However, I think this the inverse approach, an approach like for versions would be preferable: for each version you can specify if it should be shared with the project, subprojects, the project hierarchy or all projects. Something similar would be great for categories, since now I have to go by all existing projects to enable the Inherit Categories function.

---

Egbert van der Wal wrote:

For what it's worth, I applied the patch added by Serge Ballesta to SVN commit 13138 on the 2.5-stable branch. I will attach the SVN diff for that version. I had to manually change the issue\_category.rb file to apply the patch, for the rest it mostly worked right away.

I also added Dutch translations for the additional strings.

Thanks for the great effort. However, I think this the inverse approach, an approach like for versions would be preferable: for each version you can specify if it should be shared with the project, subprojects, the project hierarchy or all projects. Something similar would be great for categories, since now I have to go by all existing projects to enable the Inherit Categories function.

Is there any possibility to have plugin for this functionality. Please kindly do reply, if possible.

---

Hi,

I've installed the patch `inherit_cats_2.5.1.stable.13138.patch.gz` on my Redmine 2.5.2.stable without error. But when I set the checkbox as true, it doesn't save, resetting back to false when clicking on save.

## Where should I look for the persistence code?

Well, the problem was the file `projects_controller.rb`. It wasn't patched. Now everything is working as expected.

Valdir Stiebe Junior wrote:

Hi,

I've installed the patch `inherit_cats_2.5.1.stable.13138.patch.gz` on my Redmine 2.5.2.stable without error. But when I set the checkbox as true, it doesn't save, resetting back to false when clicking on save.

Where should I look for the persistence code?

---

This would be really useful, so I'll add my +1 and hopefully we can see this in the roadmap soon!

I've applied and successfully use the patch in our production environment (Redmine svn 2.5-stable@13419).

The only issue I have is that the tests of some plugins fail with messages "unknown method 'inherit\_cats'". And I haven't found the reason for that. The database is up-to-date (for test-env also).

Edit: Sorry - my fault - the migration files were not added to the svn patch, which I've used to set up the production. That's why my production env had not all migrations. The database take over were done via mysql. My testing environment was taken from production... and so on...

Just applied the same patch cleanly to 2.6 and 3.0.

Any chance this might be merged in at some point?

+1

It would be great to have something like this in the official codebase.

+1

+1

---

+1

Hello, the patch isn't compatible with 3.3.0. There is some changes on `project.rb` that should be reviewed.

I have just applied `inherit_cats_2.5.1.stable.13138.patch` for Redmine 3.2.1.stable, and I encountered some problems:

```
patching file app/controllers/issue_categories_controller.rb
patching file app/controllers/issues_controller.rb
Hunk #1 succeeded at 233 (offset -8 lines).
patching file app/controllers/projects_controller.rb
Hunk #1 succeeded at 74 with fuzz 2 (offset -5 lines).
Hunk #2 FAILED at 181.
1 out of 2 hunks FAILED -- saving rejects to file app/controllers/projects_controller.rb.rej
patching file app/helpers/application_helper.rb
Hunk #1 succeeded at 1334 (offset -12 lines).
patching file app/helpers/issue_categories_helper.rb
patching file app/models/issue.rb
```

Hunk #1 succeeded at 345 (offset 31 lines).  
patching file app/models/issue\_category.rb  
Hunk #1 succeeded at 35 (offset 1 line).  
patching file app/models/mail\_handler.rb  
Hunk #1 FAILED at 367.  
1 out of 1 hunk FAILED -- saving rejects to file app/models/mail\_handler.rb.rej  
patching file app/models/project.rb  
Hunk #1 succeeded at 138 (offset 5 lines).  
Hunk #2 FAILED at 399.  
Hunk #3 FAILED at 415.  
Hunk #4 FAILED at 429.  
Hunk #5 succeeded at 1072 with fuzz 2 (offset -6 lines).  
3 out of 5 hunks FAILED -- saving rejects to file app/models/project.rb.rej  
patching file app/views/context\_menus/issues.html.erb  
Hunk #1 FAILED at 77.  
1 out of 1 hunk FAILED -- saving rejects to file app/views/context\_menus/issues.html.erb.rej  
patching file app/views/issue\_categories/destroy.html.erb  
patching file app/views/issues/\_attributes.html.erb  
patching file app/views/projects/\_form.html.erb  
patching file app/views/projects/settings/\_issue\_categories.html.erb  
patching file config/locales/en.yml  
Hunk #1 succeeded at 343 (offset 9 lines).  
Hunk #2 succeeded at 1123 (offset 64 lines).  
patching file config/locales/fr.yml  
Hunk #1 succeeded at 363 (offset 29 lines).  
Hunk #2 succeeded at 1141 (offset 125 lines).  
patching file config/locales/nl.yml  
Hunk #1 succeeded at 1012 (offset -8 lines).  
Hunk #2 succeeded at 1057 (offset -8 lines).  
patching file db/migrate/20120830171625\_add\_inherit\_to\_project.rb  
patching file db/migrate/20130801123526\_rename\_inherit\_catags.rb  
patching file test/functional/issue\_categories\_controller\_inheritance\_test.rb  
patching file test/functional/projects\_controller\_inheritance\_test.rb  
patching file test/unit/helpers/issue\_categories\_helper\_test.rb  
patching file test/unit/inherited\_categories\_test.rb  
patching file test/unit/inherited\_issue\_category\_test.rb  
patching file test/unit/inherited\_issue\_project\_test.rb

I have no ideas how to develop Ruby language. Can anyone help me to solve it?  
Thank all you so much and sorry for my English.

---

Hello,  
I tried to apply the patch with Redmine 3.3 and it goes wrong.  
I had a look in the patch : there are some minor issues I can fix on my own.

Unfortunately two files have had major changes :

- projects\_controller.rb
- project.rb As I don't know the model I cannot apply changes that easily.

Please can someone with some experience give a hand ?

Hello,  
I implemented the category inheritance function on the current 3.4.

It is working on my environment, but I have not done enough operation check yet.  
Test code is not prepared yet.

Is there anyone who can help?

Please check below URL

<http://www.redmine.org/issues/5358#note-97>

I hope it incorporated into the core of 3.4, 4.0.

Regards.

I added a category inheritance function to Redmine 3.4.6 and released it to GitHub repository.

See <http://www.redmine.org/issues/5358#note-98>

Hi, we have developed a patch that is a generalization of Categories sharing.

I think it replies totally to the need expressed here.

The only thing you have to do is to create a new ProjectEnumeration custom field.

You can check it here :

[http://www.redmine.org/plugins/redmine\\_smile\\_project\\_enumerations\\_custom\\_field\\_format](http://www.redmine.org/plugins/redmine_smile_project_enumerations_custom_field_format)

related\_issues

relates,New,5358,Share Issues Categories for sub-projects

#### 履歴

#1 - 2022/05/10 17:15 - Admin Redmine

- カテゴリ を Issues\_2 にセット