

Create an 'Involve' mechanism to private issues

2022/05/09 15:38 - 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:	Candidate for next major release_32	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/8488	status_id:	1
category_id:	2	tracker_id:	2
version_id:	32	plus1:	24
issue_org_id:	8488	affected_version:	
author_id:	6054	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	151		

説明

As requested by some people (#7414#note-16, #7412#note-3, and so on), sometimes we need to involve someone that cannot access an issue by "default rules". It can happen, for example, when a customer send a problem via email and we need to create a Redmine issue ourself, and give this customer access to the ticket.

I asked Jean-Phillipe on #7412#note-2 to extend the visibility of an issue to observers, and some people liked it, but it was not done because observe mechanism was designed for notification purposes (I now I agree it is not good, that's why I'm creating this feature request).

So my request is:
Create a involve mechanism where some roles can add people in an issue, in a very similar way they add an observer, and the added users have access to the issue the were involved.

workaround
Use patch
<http://www.redmine.org/issues/14318#note-36>

journals

Also, it would be cool to set visibility level for each comment, like in JIRA, where you can write somethings to a specific users group.

Anton Nepomnyaschih wrote:

Also, it would be cool to set visibility level for each comment, like in JIRA, where you can write somethings to a specific users group.

Anton, There is already a feature request for that, #1554. Let's try don't mix requests.

Oh, you are right! Sorry!

+1

+1

ty!

+1

+1

There is a CCC PPlugin, which can add email addresses to be notified beyond redmine user base.

Terence Mill wrote:

There is a CCC PPlugin, which can add email addresses to be notified beyond redmine user base.

This comment has no relation to the issue. This issue not about notification, but about issue access control

While I've read the commentary about how watchers are the wrong mechanism for this, I needed this functionality to upgrade my company to v1.2.1 and meet our privacy needs. We'd previously been using the 'view own issues' patch in an old release which works roughly the same way.

Consequently, we cooked up this patch. It over-rides the @addable_watcher_users@ method from @acts_as_watchable@ to allow any user who can view issues to be added as a watcher, and then adds watchers of the issue to the @visible@ named scope and the @visible?@ method so that they can see operate on the issue.

I don't expect this patch to be accepted but if, like me, you can't wait another 4 years for this to get straightened out then maybe this will help you.

I've only tested this in the context of my environment, so I can't promise it won't do terrible things to yours. I'm not Rails wizard, but I think it's pretty low impact. Your Mileage May Vary. If it breaks, you get to keep both parts.

Hello

After patching some problems happened :

Simple users cant access Activity page and give 500 Internal Error .

so i redo patch changes and all things work true .

I have updated the patch for allowing watchers to see private issues to fix an issue with the Activity page -- there was a reference to Issue.visible_conditions in the journal model (which is related to the Activity page) which requires an include of the :watchers relation.

This updated patch should fix the issue with Activity giving a 500 Error.

Well done .

Thanks .

+1

In the meantime, thanks for sharing your patch, Justin!

Thank you Bruno for sharing this, saved me a lot of time and effort.

Would be nice to see it implemented upstream as well.

Alessio Cassibba wrote:

Thank you Bruno for sharing this, saved me a lot of time and effort.

I'm not the patch author, but you're welcome anyway :) The patch has been submitted by

Justin Hahn.

+1

Justin Hahn wrote:

I have updated the patch for allowing watchers to see private issues to fix an issue with the Activity page -- there was a reference to Issue.visible_conditions in the journal model (which is related to the Activity page) which requires an include of the :watchers relation.

This updated patch should fix the issue with Activity giving a 500 Error.

Thanks Justin!! exactly what i was searching for!!! :)

Justin Hahn wrote:

I have updated the patch for allowing watchers to see private issues to fix an issue with the Activity page -- there was a reference to Issue.visible_conditions in the journal model (which is related to the Activity page) which requires an include of the :watchers relation.

This updated patch should fix the issue with Activity giving a 500 Error.

But now if the assignee is changed, the old assignee gets an 403 error if he is only allowed to see tickets as an assignee or ticket issuer.

Is there a way to redirect the user to the issues page after altering the ticket assignee?

Is there a way to add the one updating the issue automatically as a watcher?

I guess I could port this patch to the 1.3-stable branch, but couldn't test yet. If someone could give some feedback on it, I would appreciate. :)

Bruno Medeiros wrote:

I guess I could port this patch to the 1.3-stable branch, but couldn't test yet. If someone could give some feedback on it, I would appreciate. :)

Hey Bruno,

i tested it on 1.3 and it seems to work like a charme :) Thank you very much. Only question i have, is it possible to send an email also out to the watcher, when he was added to the issue or the issue was updated?

Thank you very much for your work!! :)

Francesco

Francesco Trigger wrote:

i tested it on 1.3 and it seems to work like a charme :) Thank you very much. Only question i have, is it possible to send an email also out to the watcher, when he was added to the issue or the issue was updated?

I don't believe we can send email when adding the observer/watcher, but redmine is supposed to send an email when the issue is updated if the proper config is set on the administration options.

Thank you very much for your work!! :)

You're welcome, but I'm not the author of the patch, just ported it. We all need to thank

Justin Hahn! :)

mm MMY wrote:

After patching some problems happened :
Simple users cant access Activity page and give 500 Internal Error .

If I understand correctly, I have the same problem.

I use last patch: "private_issue_watchers_FIXED_1.3.diff":

http://www.redmine.org/attachments/6892/private_issue_watchers_FIXED_1.3.diff, but simple users can't access to links like this
@/projects/project_name/issues/report@.

The error, that I've got:

```
Mysql::Error: Unknown column 'watchers.user_id' in 'where clause': select          s.id as status_id,
                                          s.is_closed as closed,
                                          j.id as tracker_id,
                                          count(issues.id) as total
from
      issues, projects, issue_sta
where
      issues.status_id=s.id
      and issues.tracker_id=j.id
      and issues.project_id=projects.i
      and (((projects.id = 2) AND
IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND (projects.id IN (1,43,34,
OR ((projects.is_public = 1) OR (projects.id IN (SELECT projects.id
FROM projects, project_non_member_users
WHERE projects.id = project_non_member_users.project_id
AND (project_non_member_users.user_id = 886
OR project_non_member_users.group_id IN (240,241,293,
854))))
AND ((issues.is_private = 0 OR issues.author_id = 886 OR issues.assigned_to_id
IN (886,240,241,293,467,573,638,655,659,695,711,718,804,854) OR watchers.user_id = 886))) OR projects.id IN (13)))
group by s.id, s.is_closed, j.id
```

Backtrace:

```
C:/Ruby/lib/ruby/gems/1.8/gems/activerecord-2.3.14/lib/active_record/connection_adapters/abstract_adapter.rb:227:in `log'
C:/Ruby/lib/ruby/gems/1.8/gems/activerecord-2.3.14/lib/active_record/connection_adapters/mysql_adapter.rb:324:in `execute'
C:/Ruby/lib/ruby/gems/1.8/gems/activerecord-2.3.14/lib/active_record/connection_adapters/mysql_adapter.rb:639:in `select'
C:/Ruby/lib/ruby/gems/1.8/gems/activerecord-2.3.14/lib/active_record/connection_adapters/abstract/database_statements.rb:7:in
n
`select_all_without_query_cache'
C:/Ruby/lib/ruby/gems/1.8/gems/activerecord-2.3.14/lib/active_record/connection_adapters/abstract/query_cache.rb:60:in `select_all'
C:/Ruby/lib/ruby/gems/1.8/gems/activerecord-2.3.14/lib/active_record/connection_adapters/abstract/query_cache.rb:81:in `cache_sql'
C:/Ruby/lib/ruby/gems/1.8/gems/activerecord-2.3.14/lib/active_record/connection_adapters/abstract/query_cache.rb:60:in `select_all'
D:/Draft/redmine_svn/app/models/issue.rb:1073:in `count_and_group_by'
D:/Draft/redmine_svn/app/models/issue.rb:751:in `by_tracker'
D:/Draft/redmine_svn/app/controllers/reports_controller.rb:31:in `issue_report'
```

And when I run tests, some of them give the same errors.

The patch 1.3 dont work for our updated Redmine 1.4.1 (from 1.2.1).
Are there any solutions?

Thanks for any hints.

+1, this is a major requirement when dealing with a part of the company that should only have access to issues that you've assigned to as watchers when they may only have "view

own issues" permissions.

We need also a solution for 1.4.1

Is there somebody we can hire to solve this problem?

Max Schwer wrote:

We need also a solution for 1.4.1
Is there somebody we can hire to solve this problem?

I would offer myself if I were something more than a rails noob.. :P

Max Schwer, I successfully applied code from "private_issue_watchers_FIXED_1.3.diff" in Redmine 2.0.1 (just manually copy-paste in app/model/issue.rb and app/model/journal.rb). And now watchers can view private issues. Thanks to Justin Hahn and Bruno Medeiros.

Vadim Goncharov wrote:

Max Schwer, I successfully applied code from "private_issue_watchers_FIXED_1.3.diff" in Redmine 2.0.1 (just manually copy-paste in app/model/issue.rb and app/model/journal.rb). And now watchers can view private issues. Thanks to Justin Hahn and Bruno Medeiros.

How stable is this? What is the process to get this steamed into the main Redmine branch?

I guess we could pull this out into a plugin, but it would be nicer to have Redmine support this idea natively.

I have just uploadaded the diff with 1.4.1 version.

I would really love this small improvement to be pushed into mainstream.

Applied ok on current 1.4-stable, didn't fully tested yet. Thanks!

Does the inviolved user have any rights in redmine or project? I assime that the user at least mus be registered. Or is this possible with adding an email the owber gest one time links?

William Roush wrote:

How stable is this? What is the process to get this steamed into the main Redmine branch?

As explained on the second paragraph of the description, this approach will not be accepted on Redmine. I personally see one big downside now (I defended this approach in the past): If an user that gained access to the ticket by the observer mechanism clicks 'Unwatch', he irreversibly looses access to the ticket. That's not acceptable and that's why I created this ticket to ask for a 'Involve mechanism' very similar to the Observe mechanism.

I guess we could pull this out into a plugin, but it would be nicer to have Redmine support this idea natively.

This is more reasonable for now, but I don't know how to create plugins.

Bruno Medeiros wrote:

If an user that gained access to the ticket by the observer mechanism clicks 'Unwatch', he irreversibly looses access to the ticket. That's not acceptable and that's why I created this ticket to ask for a 'Involve mechanism' very similar to the Observe mechanism.

Ok, let suppose feature 'Involve' will be done.
What should redmine do when some customer ask to uninvolve from issue?
I suppose he should loose access to the ticket. Should he?
to stop confuse only one additional feature required to be added to attached patch:

When customer click "unwatch" redmine should check - this customer loose access after this action and if so - show new additional dialogue like this:
Attention! unwatch will loses access to the ticket. Continue?

So i think redmine should use "Watcher mechanism" and should not do new "Involve mechanism".

Maxim Nikolaevich wrote:

Ok, let suppose feature 'Involve' will be done.
What should redmine do when some customer ask to uninvolve from issue?
I suppose he should loose access to the ticket. Should he?

I think that involved should not uninvolve himself.

to stop confuse only one additional feature required to be added to attached patch:

When customer click "unwatch" redmine should check - this customer loose access after this action and if so - show new additional dialogue like this:
Attention! unwatch will loses access to the ticket. Continue?

So i think redmine should use "Watcher mechanism" and should not do new "Involve mechanism".

And what if you want to involve someone, but don't want to have this person being notified by email? There is no way to do that using only one mechanism.

I see your point, and that was my point before, but I can understand why Jean-Phillipe don't like it. Using something that was designed for notification purposes to access control is kind of a hack and will have its limitations.

I like this patch, and I'm using this on my redmine instance, and I say thanks to the author, but I understand if it's no integrated at any time.

Is there a fix for 2.0 ?

Francesco Trigger wrote:

Is there a fix for 2.0 ?

I tried to modify the files like in Redmine 1.x but i get this trying to access redmine in the production.log.

```
Connecting to database specified by database.yml
Creating scope :active. Overwriting existing method User.active.
Creating scope :open. Overwriting existing method Version.open.
DEPRECATION WARNING: The InstanceMethods module inside ActiveSupport::Concern will be no longer included automatically.
Please define instance methods directly in Collectiveldea::Acts::NestedSet::Model instead. (called from include at
/opt/redmine/apps/redmine/htdocs/lib/plugins/awesome_nested_set/lib/awesome_nested_set/awesome_nested_set.rb:58)
OpenIdAuthentication.store is nil. Using in-memory store.
```

Francesco Trigger, could you provide a patch? It's not a good idea override whole files since they change over time.

I'm sorry for long time answer

Bruno Medeiros wrote:

And what if you want to involve someone, but don't want to have this person being notified by email? There is no way to do that using only one mechanism.

Can you describe real case how it can be: 'Guy was involved but should not notified' ?
current settings about email notification "here":<http://www.redmine.org/my/account> says:

1. For any event on all my projects
2. Only for things I watch or I'm involved in
3. Only for things I am assigned to ...

So if you are involved - you should receive emails - it's like you watch.
If you watcher - you are involved.
It's easy to understand and easy to code.

In other way let me suppose involved and watched are different things.
I see many problems in this way like:

1. Guy is watcher but has no permission - somebody think they receive message but actually not
2. It's possible to set watchers in time of creating issue but probably watcher has no permissions...

So i see only one issue - 'unstar' with lose permission - but my solution i mention in comment above

In other words:

What is goal for involving somebody?

Goal is to notify him when hidden issue is changed. Goal is to allow him see hidden issue changes.

So watcher = involved i think

Let me know if i miss something

Maxim Nikolaevich wrote:

In other words:

What is goal for involving somebody?

Goal is to notify him when hidden issue is changed. Goal is to allow him see hidden issue changes.

So watcher = involved i think

Let me know if i miss something

Yes, you missed. Involve someone doesn't necessarily mean that you want this person being notified by email. You may want to let this user only access an issue when he wants to, not notifying him every time the issue is changed. You're simplifying things to much considering that involve = observe.

Bruno Medeiros wrote:

Involve someone doesn't necessarily mean that you want this person being notified by email. You may want to let this user only access an issue when he wants to, not notifying him every time the issue is changed.

If somebody assign issue to you it means you will be notified every time about changing in this issue. Is it problem? I think no.
So when somebody set you as watcher - you also will be notified about every changing in this issue.

I can't to find real usacase when i need to involve somebody to issue (from closed project!) without notifying them about changing issue.

If i assign issue from closed project to some guy then him will be notified about every changes, right?

I completely agree with -Bruno Medeiros- Maxim Nikolaevich. Watcher = involved, involve = observe.

For my project this feature is key feature. Only inability to give personal access to private issues blocks upgrade of Redmine from 1.3 to 2.x (with lot of fixes and new features).

I can not organize customer's workflow without this feature.

Dear developers, can you summarize status of this issue?

I completely agree with Bruno Medeiros. Watcher = involved, involve = observe.

Bruno Medeiros has different point of view. Bruno Medeiros tell ut these should be different settings. So it's strange that you are agree but tell my point of view about this things.

What you want to tell really:

1. You agree with me
2. You think watcher should be not equal involved, and involve should be not equal observe.

Dear developers, can you summarize status of this issue?

I tottaly agree. Dear developers please give us your plans about this issue.

Maxim, you are right. I just copied a wrong name to my post.

So, my main idea is "Watcher = involved, involve = access". I see no reason to "watch" something and have no access to it but gets only notification mail about it with, actually, all same information, but packed in e-mails.

Or "be involved" but get no notification mails.

"Be envolved" mostly means "have access and get notifications" and Redmine should provide it in this way.

Bruno Medeiros wrote:

Francesco Trigger, could you provide a patch? It's not a good idea override whole files since they change over time.

+1.

i have "deploy overwriting" mode.

thanks,

Daniele

Attached is the patch for version 2.1.

The original patch resolve not only watchers for private issue.

It is also resolve, watchers not able to view non-public projects issues when they are path of the watchers list.

I am wondering why Redmine developers are not integrating this patch.

Does it cause any other bug or slow down the query?

+1.

This would significantly aid the use of Redmine in a Help Desk situation. Honestly, the "view Issues Created by or Assigned to" should include "or Watching."

Maxim Nikolaevich wrote:

I completely agree with Bruno Medeiros. Watcher = involved, involve = observe.

Bruno Medeiros has different point of view. Bruno Medeiros tell ut these should be different settings. So it's strange that you are agree but tell my point of view about this things.

What you want to tell really:

1. You agree with me
2. You think watcher should be not equal involved, and involve should be not equal observe.

Men, I believe you're missing the main point here. I created this ticket because Redmine developers rejected the idea of using observe mechanism to do access control.

I said it before and I'll say it now again: As user, I agree with you and I use the patch that is being maintained here. Using the already existing observe mechanism to involve people is reasonable.

But, as a developer, I can understand why Redmine developers don't want to integrate this patch.

Please, let's stop asking something that has already been rejected by Redmine developers. it will only bother them.

Dear developers, can you summarize status of this issue?

I totally agree. Dear developers please give us your plans about this issue.

The status is implicit: Implement a involve mechanism, apart from observer mechanism, and it can be considered to be integrated on Redmine.

Or we can maintain this patch in parallel forever (many thanks for who is doing that now, I'm a very happy user of this patch :)).

Hi,

After applying this patch, I got the following error when trying to access page '<http://myserver/projects/issues/report>'

```
ActiveRecord::StatementInvalid (Mysql::Error: Unknown column 'watchers.user_id' in 'where clause': select  s.id as status_id,
s.is_closed as closed,
j.id as tracker_id,
count(issues.id) as total
from
issues, projects, issue_statuses s, trackers j
where
issues.status_id=s.id
and issues.tracker_id=j.id
and issues.project_id=projects.id
and (((projects.id = 84) AND (projects.status=1 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE
em.name='issue_tracking')) AND (projects.id IN (86,89) OR (projects.id IN (84,80) AND ((issues.is_private = 0 OR issues.author_id =
3 OR watchers.user_id = 3 OR issues.assigned_to_id IN (3,5))))))
group by s.id, s.is_closed, j.id):
app/models/issue.rb:1099:in count_and_group_by'
app/models/issue.rb:789:in by_tracker'
app/controllers/reports_controller.rb:31:in `issue_report'
```

But when I access this page with admin account, I do not get this error. I am really appreciate your help!

Regards,

Bruno Medeiros wrote:

I created this ticket because Redmine developers rejected the idea of using observe mechanism to do access control.

I saw #7412-13 discussion

Jean-Philippe Lang wrote:

A user who unwatches a private issue would lost access to it. It may be quite confusing for many users.

So developers rejected this patch because, it does not solve one very important usecase (which i describe above):

When customer click "unwatch" redmine should check - this customer loose access after this action and if so - show new additional dialogue like this:
Attention! unwatch will loses access to the ticket. Continue?

So i agree without additional dialogue this feature should not be used in production. (like this patch allow)
But i think this dialogue not so hard to code

hm... this page has 'Google Page Rank'=3

So google see how this feature is important :)

Maxim Nikolaevich wrote:

So developers rejected this patch because, it does not solve one very important usecase (which i describe above):

When customer click "unwatch" redmine should check - this customer loose access after this action and if so - show new additional dialogue like this:
Attention! unwatch will loses access to the ticket. Continue?

So i agree without additional dialogue this feature should not be used in production. (like this patch allow)
But i think this dialogue not so hard to code

This isn't the only issue with the current patch/approach. I will list all the problems I see now (again):

Unwatch make user irreversible lose access to ticket - This is the first issue people first realized, and maybe the biggest issue in this patch. And I don't think create a dialog fixes the problem. Make a confirmation dialog would create a big inconsistency: If I click unwatch in a ticket that I have access granted by "normal" rules, I turn off email notification for this ticket. If I do that with a ticket that I'm involved, I lose access? That doesn't make sense at all.

Be involved not equals watch - This is a fact. If they seem the same for you, you need less from Redmine than other people would. A watcher needs to be involved, but involved people don't necessarily need to watch. Like I said before: Involve someone doesn't necessarily mean that you want this person being notified by email. You may want to let some user only access an issue when he wants to, not notify this user every time the issue is changed. You're simplifying things to much considering that involve = observe.

Permission control needs to be separated - We have today 3 permissions: Add watcher, Remove watcher, See watchers list. Using observe mechanism to involve people implies that every role that is able to add/remove watchers will be able to involve/uninvolve people. This would be a design failure. We need separated permissions for involve.

Watch mechanism was designed for notification purposes - And when I say mechanism I don't mean only the option to watch/unwatch a ticket, I mean

this combo on the ticket screen, plus all the permissions to allow different roles to add and remove watchers, plus all tables in the database, mail senders, etc. All this was designed thinking of notifying people with emails about changes, not control access. If Redmine developers decide to use this mechanism to control access to tickets, this could bring problems to the project in the future.

And before people start to say again that I'm against this patch: I'm not against this patch, I'm a very happy and thankful user of it. I just understand why Redmine developers refuse to integrate it.

Abdul Halim Mat Ali wrote:

Attached is the patch for version 2.1.

The original patch resolve not only watchers for private issue.
It is also resolve, watchers not able to view non-public projects issues when they are path of the watchers list.

I am wondering why Redmine developers are not integrating this patch.
Does it cause any other bug or slow down the query?

Hey,

thanks for the new patch, i tried it, but it does not work :(
Any ideas?

Francesco

I have tried it on Redmine version 2.1.0.
I don't know whether it will work on other version or on the latest svn checkout.

Francesco Trigger wrote:

Abdul Halim Mat Ali wrote:

Attached is the patch for version 2.1.

The original patch resolve not only watchers for private issue.
It is also resolve, watchers not able to view non-public projects issues when they are path of the watchers list.

I am wondering why Redmine developers are not integrating this patch.
Does it cause any other bug or slow down the query?

Hey,

thanks for the new patch, i tried it, but it does not work :(
Any ideas?

Francesco

+1

I would love this feature, although I understand this patch would break a lot of currently used workflow.

Thanks for the patch, it works well with 2.1.

Christophe Badoit wrote:

+1

I would love this feature, although I understand this patch would break a lot of currently used workflow.

Thanks for the patch, it works well with 2.1.

What kind of workflow it will break? I am curious.

+1

And also, there is a need for redmine 2.2

I was tried to manual update but there was not successfully. :(

Is there a plan for integrating this patch to stable releases?

+1

I met the same question, add a watcher means u want notify the issue's change to the watcher.

Because many people seem to need this. I propose to set this as a candidate for the next major release. Maybe Jean-Philippe would disagree, but this is just a proposal. :-)

As I suggested in #12666, I think a clean way to do this would be, when creating a custom field and setting the type to 'user', having a check box for enabling involvement in the issue. Coupled with multi-select, this lets you have as much or as little control over how users are involved in an issue as you need.

I have updated the patch for v2.2.3 of redmine. It appears to be working fine for me. Please let me know if you experience problems with it on 2.2

It turns out that patch has a subtle defect. Attempting to sort on Assignee or any other user-based column would result in no data being returned. After much debugging, I think I've found a resolution. For the technically inclined, I've moved from eager loading the :watchers association to using an explicit LEFT OUTER JOIN via a :join parameter/method.

It's not obvious to me at all why this was breaking with watchers as an include, and I suspect the way I've done this is a touch ugly (It's certainly not very DRY), but it works at least. I really wish the proper "Involve" functionality were in place so we could end of life this tweak.

Anyhow, corrected patch attached.

Justin Hahn wrote:

It turns out that patch has a subtle defect. Attempting to sort on Assignee or any other user-based column would result in no data being returned.

I tried to apply your patch and got an error when trying to see a project or "My page":

```
(0.5ms) SELECT COUNT(DISTINCT `issues`.`id`) FROM `issues` LEFT OUTER JOIN `projects` ON `projects`.`id` = `issues`.`project_id` LEFT OUTER JOIN `issue_statuses` ON `issue_statuses`.`id` = `issues`.`status_id` LEFT OUTER JOIN "watchers" ON "watchers"."watchable_id" = "issues"."id" AND "watchers"."watchable_type" = 'Issue' WHERE `issues`.`assigned_to_id` IN (11, 17, 24, 85) AND (((projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='cking')) AND ((projects.id IN (73,49,12,77) AND ((issues.author_id = 11 OR watchers.user_id = 11 OR issues.assigned_to_id IN (11,17,24,85)))) OR (projects.id IN (66,72) AND ((issues.is_private = 0 OR issues.author_id = 11 OR watchers.user_id = 11 OR issues.assigned_to_id IN (11,17,24,85)))) OR (projects.is_public = 1 AND ((issues.author_id = 11 OR watchers.user_id = 11 OR issues.assigned_to_id IN (11,17,24,85)))) OR projects.id IN (72) OR (projects.id IN (69,68,64,57,63,71,12,70,66,72,3) AND (issues.is_private = 0 OR issues.author_id = 11 OR watchers.user_id = 11 OR issues.assigned_to_id IN (11,17,24,85))))))
```

```
atuses.is_closed = 0)
```

```
mysql::Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ""watchers" ON "watchers"."watchable_id" = "issues"."id" AND "watchers"."watchabl' at line 1: 5
UNT(DISTINCT `issues`.`id`) FROM `issues` LEFT OUTER JOIN `projects` ON `projects`.`id` = `issues`.`project_id` LEFT
JOIN `issue_statuses` ON `issue_statuses`.`id` = `issues`.`status_id` LEFT OUTER JOIN "watchers" ON "watchers"."watchabl
= "issues"."id" AND "watchers"."watchable_type" = 'Issue' WHERE `issues`.`assigned_to_id` IN (11, 17, 24, 85) AND (((p
tus <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND
ts.id IN (73,49,12,77) AND ((issues.author_id = 11 OR watchers.user_id = 11 OR issues.assigned_to_id IN (11,17,24,85)
projects.id IN (66,72) AND ((issues.is_private = 0 OR issues.author_id = 11 OR watchers.user_id = 11 OR issues.assigned
(11,17,24,85)))) OR (projects.is_public = 1 AND ((issues.author_id = 11 OR watchers.user_id = 11 OR issues.assigned
11,17,24,85)))) OR projects.id IN (72) OR (projects.id IN (69,68,64,57,63,71,12,70,66,72,3) AND ((issues.is_private = 0
s.author_id = 11 OR watchers.user_id = 11 OR issues.assigned_to_id IN (11,17,24,85)))))) AND (issue_statuses.is_closed
```

I suppose there is a problem because of double-quotes (") instead of back quotes (`) around some table/column names:

```
LEFT OUTER JOIN "watchers" ON "watchers"."watchable_id" = "issues"."id" AND "watchers"."watchable_type" = 'Issue'
```

I tried to modify your patch according to this (attaching). I will keep testing it.

Notes:

- I am less than beginner with Ruby/Rails, I work with it only from sysadmin point of view. Therefore, I might have done some incredibly stupid things because of misunderstanding. Feel free to correct me.

*** I am using @mysql@ database backend. I have no idea if the changes might break if you use other database.**

Stanislav, thanks for the report. However, it turns out there are other problems with the patch I put together. I use/test on postgres, so the issue you experienced must be due to that difference. Regardless, the change I put together is bad. It will break pagination on large issues lists in that pages will have fewer results than expected and you'll end up with inaccessible issues. (This is because the join that I added causes the query Rails/Redmine issues to the database to return duplicates)

So, instead, I've modified the patch to work differently. Instead of attempting to :join or :include the watchers relation, I've gone with fetching a list of watched issue ids by the given user, and then adding those issue ids to the WHERE clause on the SQL conditions. This seems to work much better, and is less invasive (in particular we no longer have to modify the app/models/journal.rb file)

I think this updated patch may solve both our issues. To be honest, I'm surprised past versions of this patch didn't have similar issues. Make sure to apply this against the pristine sources, and not against a previously patched version of the code. (revert app/models/issue.rb and app/models/journal.rb)

Patch attached.

I hate to update my own patch in less than an hour, but I found a slightly cleaner way to do what I did. So, alas, here is FIXED_4. It's not, functionally, any different than FIXED_3 but the code is a bit cleaner.

I use redmine 2.2.3

Environment:

Redmine version	2.2.3.stable
Ruby version	1.9.2 (x86_64-linux)
Rails version	3.2.12
Environment	production
Database adapter	Mysql2

After install the patch (private_issue_watchers_2.2_FIXED_4.diff) when the page opens with a list of problems some users displays 500 error.

log:

```
Started GET "/projects/5135p?jump=overview" for 172.19.32.97 at 2013-03-07 09:57:57 +0400
Processing by ProjectsController#show as HTML
  Parameters: {"jump"=>"overview", "id"=>"5135p"}
  Current user: sapatoit (id=89)
```

Redirected to http://172.19.32.47:3000/projects/5135p
Completed 302 Found in 10ms (ActiveRecord: 1.1ms)
Started GET "/projects/5135p" for 172.19.32.97 at 2013-03-07 09:57:57 +0400
Processing by ProjectsController#show as HTML
Parameters: {"id"=>"5135p"}
Current user: usertest(id=89)
Completed 500 Internal Server Error in 67ms

```
ActiveRecord::StatementInvalid (Mysql2::Error: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near ')))) OR (projects.id IN (14,3) AND ((issues.author_id = assigned_to' at line 1: SELECT COUNT(DISTINCT `issues`.`id`) AS count_id, tracker_id AS tracker_id FROM `issues` LEFT JOIN `projects` ON `projects`.`id` = `issues`.`project_id` LEFT OUTER JOIN `issue_statuses` ON `issue_statuses`.`id` = status_id` WHERE (((projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em=issue_tracking')) AND ((projects.is_public = 1 AND ((issues.author_id = 89 OR issues.assigned_to_id IN (89,97) OR issues.id IN ()))) OR (projects.id IN (14,3) AND ((issues.author_id = 89 OR issues.assigned_to_id IN (89,97) OR issues.id IN ()))))) AND (issue_statuses.id IN (14) AND ((issues.author_id = 89 OR issues.assigned_to_id IN (89,97) OR issues.id IN ()))))) AND (issue_statuses.id = 0) AND ((projects.id = 14 OR (projects.lft > 7 AND projects.rgt < 8))) GROUP BY tracker_id);
app/controllers/projects_controller.rb:156:in `show'
```

After removal of the patch, everything works fine.

How can I fix this?

Oh, interesting. I believe the problem is there are no issues being watched by those users. I don't have a good way to test this right now. Can you try this to verify for me?

Take one of the users with this issue and have them watch a single issue. If the problem goes away, I think I know how to fix it. If not, let me know.

OK, I've been able to reproduce in my environment. This patch should fix that issue. We now only insert the watched issues.id clause when there are actually watched issues for the user. Let me know if you run into anything else.

OK, I've been able to reproduce in my environment. This patch should fix that issue. We now only insert the watched issues.id clause when there are actually watched issues for the user. Let me know if you run into anything else.

Justin, I tried your last patch and it seems to work (still using mysql).

Btw, I've also tried the original patch (private_issue_watchers_2.2.diff) and haven't been able to reproduce the problem. Can you tell me the exact way how to do that?

Does the patch work with Redmine 2.3.0?

Given that 2.3.0 was just released two days ago, I'd say there's at best a 50/50 chance of it. Feel free to post an updated version if it doesn't.

Justin Hahn, thank you for the patch, I tried it (private_issue_watchers_2.2_FIXED_5.diff) and it works when I create the issue -but not when the issue is already created and watchers are added. Using redmine 2.2.3.-

Works ok

The patch seems working with 2.3.0.

I've made a plugin for the 1.4.x version: http://www.redmine.org/plugins/redmine_extended_watchers

It also disables adding watchers foreign to the project. It seems weird to me the possibility to be watchers and not being able to inspect and be notified about some issue, in private projects.

Probably this aspect needs fix with public projects; any contribution is appreciated

I've just upgraded my Redmine to 2.3 and installed the plugin. So far, so good!

One downside I see on the plugin approach is that some fragments of the core are copied in the plugin, and there will be no conflict if the core changes. Any thoughts on this?

Hello,

private_issue_watchers_2.2_FIXED_5.diff and redmine_extended_watchers plugin have same effect and they works well. But watcher search function does not work during the adding new watcher when you use one of them ??

Environment:

Redmine version 2.2.3.stable
Ruby version 1.9.3 (x86_64-linux)
Rails version 3.2.12
Environment production
Database adapter MySQL2

Unfortunately after patching issue.rb on Redmine 2.5.0 stable I have an Internal Error 500 when I want to update issue. More strange is that after reverting changes, error still appears. Apart of this error, patch doesn't make issues visible for watchers.

Hi,

I had the same error as Konrad on my Redmine 2.4.2 (working on Bitnami VM). Updating the patch helped. I attach the patch working in my 2.4.2 version (maybe it will also work in 2.5.0?).

Thank you Krzysztof. It's working on 2.5.0.

this patch is not working on our version , i am trying to patch redmine 2.5.1.stable

I getting the following outup

```
File to patch: app/models/issue.rb
patching file app/models/issue.rb
Hunk #1 FAILED at 94.
Hunk #2 succeeded at 134 (offset 13 lines).
Hunk #3 succeeded at 151 with fuzz 1 (offset 18 lines).
1 out of 3 hunks FAILED -- saving rejects to file app/models/issue.rb.rej
```

even we are not seeing any error message. but even after applying this patch, the purpose of the patch is not met. the purpose of this patch was : "Private Issues should be visible to Watchers". its not happening even after patch

Following are my environment details

Environment:
Redmine version 2.5.1.stable
Ruby version 1.9.3-p545 (2014-02-24) [i686-linux]
Rails version 3.2.17
Environment production
Database adapter MySQL

https://github.com/NEXSTEP/redmine_show_private_for_watcher

tested on last trunk

Dmitry Salashnik wrote:

The following is Jean-Phillippe Lang's comment from #7412#note-13:

watchers were designed for notification purpose only. A user who unwatches a private issue would lost access to it. It may be quite confusing for many users.

See also #14318#note-23.

So we should think different approach.

Go MAEDA wrote:

So we should think different approach.

Okay, agreed. But Redmine really needs a solution for that!

Tobias Fischer wrote:

Hang Xie wrote:

patch for 3.1.x (working on 3.1.1 and 3.1.2)

I can confirm that this patch also works for @3.2.3.stable.15464@ and should be included as a feature in the next release.

While testing the patch @private_issue_watchers_3.1.x.diff@ with Redmine 3.2.3 I found a bug which allows users to view tickets they are not assigned to or which they aren't watching.

So I wouldn't recommend using the patch in production!

For those of you who need the functionality right now, I suggest to use the patch @allow_watchers_and_contributers_access_to_issues_3.2.0.patch@ by Fabrizio Sebastiani from #14318#note-22. It works well for @3.2.3.stable.15464@.

However, I still agree with Go MAEDA that we need another/better solution for implementation in one of the next releases...

+1 for this feature

Go MAEDA wrote:

I want this feature, but I think any patches that uses watchers for access controlling purpose would not be accepted for now.

The following is Jean-Phillippe Lang's comment from #7412#note-13:

watchers were designed for notification purpose only. A user who unwatches a private issue would lost access to it. It may be quite confusing for many users.

So we should think different approach.

We've had similar requests to this at Planio numerous times and we've spent quite some time internally thinking about better or alternative ways. Yet we couldn't come up with anything simple that wouldn't make things even more confusing for users. Being able to set people as watchers AND being able to "involve" users via two different buttons seemed to be causing even more confusion unfortunately.

So, we've implemented this based on watchers for Planio and it seems that users "get it". We're not receiving many support requests by people who are confused by the new feature.

Felix has posted the patch in #23546 for everyone's consideration :-)

Jan from Planio www.plan.io wrote:

So, we've implemented this based on watchers for Planio and it seems that users "get it". We're not receiving many support requests by people who are confused by the new feature.

Jan, I'm a happy Planio user and at the same time my company develop a custom solution based on Redmine. I see your point and I use this patch in quite a few Redmine instances. As a Redmine core developer, though, I wouldn't accept it on core, exactly as Jean-Philippe stated. I wonder if it is really a good idea to have it on Planio core.

The main problem with this patch, reported by a few of our custom solution's customers, is: Once you are involved in an issue (in this case, using the observe mechanism) you cannot stop receiving emails from that issue. If you stop observing, you lose access. If you keep observing, emails keep coming. If you reach that point, there is no way back. No solution. Period. And this is kind of obvious, because this patch uses something designed to handle email notifications to do access control, which is wrong by design.

Besides that, if you allow a role to add observers, this role is also allowed to give access to users that may not be supposed to have access. "Add watchers" permission becomes much more powerful and dubious.

See "my previous comment":/issues/8488#note-105 for more details.

I think a lot of people demand that because they don't care about losing the observer mechanism, this is a valid way, but we need to accept one thing: Observe and be involved are different things, we cannot fully resolve the problem with only one flag. If you drop the observe feature and transform it in an involve feature (I think that is what most people that use this patch do), that's fine, but keep in mind that you may be creating a bigger problem for you if people want to use the observe as it was designed to work.

Bruno Medeiros wrote:

The main problem with this patch, reported by a few of our custom solution's customers, is: Once you are involved in an issue (in this case, using the observe mechanism) you cannot stop receiving emails from that issue. If you stop observing, you lose access. If you keep observing, emails keep coming. If you reach that point, there is no way back. No solution. Period. And this is kind of obvious, because this patch uses something designed to handle email notifications to do access control, which is wrong by design.

Maybe we should understand what it is to be a watcher differently. Nowadays, and I may be wrong, being a watcher means I could be notified via email: in "my account" I choose to be notified in combos regarding my involvement with the issues. I can not specify what changes I want to be notified about, neither select which level of involvement triggers notification individually. Neither I can choose to silence specific issues. Actually, users end up resorting to email filters to complement this lack of control in Redmine.

So, what really would I like to have? Let me try and design something simple, that people should "get":

- In "my account", a user should be able to choose which events to be notified about by email; ** It could be the selected subset from Administration > Settings > Email notifications; ** Not in a combo. Make each kind of notification specific.
- In "my account", a user should be able to choose if i should be notified if I am the owner, involved... ** Not in a combo. Make each kind of notification specific.
- Redmine should have a way to give unrelated people access to issues (IMO, this is what people understand as watching); ** Some roles could give more people access to the issue; ** Give view access to the issue should be the default feature.
- Receive email notifications should be a separate option: ** When seeing an issue, the user can see if they are receiving email, show in the side panel; ** Users set the default value in "my account"; ** Users could add themselves to a whitelist (for each issue) to receive emails; ** Users could add themselves to a blacklist (for each issue) to receive emails;

PS:

- If the user loses view permission to an issue, remove him from the issue whitelist or blacklist.
- Please, do not redirect him out of the project, to an error page, back link not working; ** Error 404: tell user also he may not have access to page.

* When migrating the database, whoever would be in an issue watcher list, should be added to its whitelist.

Thanks for this patch!

I was looking for a solution to this problem for a very long time.

But if use patch for 3.1.X on redmine 3.3.X show up bug: All users can see issue by direct link like this

[http://redmine_url/issues/\[issue_number\]](http://redmine_url/issues/[issue_number])

We can fix it next code in patch:

```
when 'own'
-       self.author == user || user.is_or_belongs_to?(assigned_to)
+       #self.author == user || user.is_or_belongs_to?(assigned_to)
+       !self.is_private? || (self.author == user || self.watched_by?(user) || user.is_or_belongs_to?(assigned_to))
to

when 'own'
-       self.author == user || user.is_or_belongs_to?(assigned_to)
+       #self.author == user || user.is_or_belongs_to?(assigned_to)
+       self.author == user || self.watched_by?(user) || user.is_or_belongs_to?(assigned_to)
```

@Ilya: You could also try and check the patches from related issue #14318#note-35 which are updates for Redmine 3.3 and 3.4

Tobias Fischer wrote:

@Ilya: You could also try and check the patches from related issue #14318#note-35 which are updates for Redmine 3.3 and 3.4

Tobias, Thank you!

This patch is much better!

+1

+1

+1

Definitely +1.

Redmine is an excellent piece of software, we've been wanting this Private functionality for a couple of years and it's the one causing us more trouble (the need to keep separate projects)

Diego Smalinsky wrote:

Definitely +1.

Redmine is an excellent piece of software, we've been wanting this Private functionality for a couple of years and it's the one causing us more trouble (the need to keep separate projects)

I am trying to install the 23546-watched_or_created_or_assigned_issue_visibility.patch but unable to do it.

I run the command `patch -p0 < filename.patch` but getting too many errors. Could you give me the steps you followed to install it?

Thanks.

+1

Patch on <http://www.redmine.org/issues/14318#note-35> works great. Apology for

updating closed ticket.

Is there any possibility to get this as a plugin ?

I only want some persons to see tickets they are assigned as "watcher" (or similar) to.

Philipp Neuhaus wrote:

Is there any possibility to get this as a plugin ?

I only want some persons to see tickets they are assigned as "watcher" (or similar) to.

+1

Any chance of adding this functionality?

+1

+1 Watchers need to be able to see private issues

+1 IMHO this is really a feature needed by a lot of people and need to be re-evaluated.

It's absolutely necessary to be able to have a permission like "Observer is allowed to read and comment tickets" for a role.

+1

Cyrion Beleriand wrote:

+1

This plugin did the job for me: https://github.com/maxrossello/redmine_extended_watchers

The affected watcher can see the issue he is watching, and not the other issues.

Tested on redmine 4.1.1.

+1 again and I don't agree that #14318 should be closed!

related_issues

relates,Closed,7414,Private issues

relates,Closed,7412,Add an issue visibility level to each role

relates,New,11888,No e-mail notification for non-members who are watchers

relates,Closed,12945,watcher can't see this issue from My Page-Watched issues

relates,New,12666,Involvement boolean for Custom Field 'User' type

relates,New,13533,Concept for controlling visibility of users

relates,Closed,285,Tracker role-based permissioning

relates,New,16845,Add permission rule to allow watchers can edit issues

relates,New,22977,A project member has no access and gets no notification, when being a watcher of the issue

relates,New,23546,Issue visibility "watched by, created by or assigned to" for roles

relates,New,27731,Allow users to view and comment on their own issues, even if moved into a project where they don't have access.

duplicates,Closed,9330,Watchers can not see the ticket status, if their role is set to only own or assigned tickets.

duplicates,Closed,10173,Private bugs are not visible to watchers

duplicates,Closed,12584,Watchers on Private Issues

duplicates,Closed,11718,If you add a watcher to an issue, they should have rights to view it.
duplicates,Closed,13828,when users can only see their own issues; give watchers the ability to view issues they watch
duplicates,Closed,14318,Watchers Alerted To Changes But Cannot See Issues (potentially)
duplicates,Closed,27028,Issue's visibility for the watchers
duplicates,Closed,27677,New notification settings value in role setting
duplicates,Closed,28368,regarding watcher added in private project but not getting notification
duplicates,Closed,29911,[Email Notification] involved in clarification
duplicates,Closed,13512,Add a way to make specific issues visible to a user

履歴

#1 - 2022/05/10 17:18 - Admin Redmine

- カテゴリ を Issues_2 にセット

- 対象バージョン を Candidate for next major release_32 にセット