

## Default value for the private issue flag

2022/05/09 15:52 - Admin Redmine

ステータス:	New	開始日:	2011/10/17
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/9432	status_id:	1
category_id:	2	tracker_id:	2
version_id:	32	plus1:	3
issue_org_id:	9432	affected_version:	
author_id:	8115	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	46		
<b>説明</b>			
It would be useful to have the possibility, to set the default value for the "private issues"-flag for each project.			
journals			
<p>I tried one more time to create a patch for this, but failed :(</p> <p>There is a mechanism to set <code>@safe_attributes@</code>, and I could understand it because I'm ruby/rails noob. If someone have a clue, please post it!</p> <p>This feature would be really helpful. There are almost always 2 streams of issues, internal and for the customer. It happens a lot that developers forget to choose private for internal issues. So we choose to keep a 2 project strategy for now.</p>			
<p>Geert Smit wrote:</p> <p style="padding-left: 40px;">It happens a lot that developers forget to choose private for internal issues.</p> <p>It's just a matter of upgrading the issues model.</p> <p>We have uploaded an Active Record file for this migrate. We tested it and are using it. Now, all issues are created private by default.</p> <p>Greetings,</p>			
<p>I wrongly marked this as Resolved (can't revert the change now). The issue is not resolved as it is not configurable at project level.</p>			
<p>+1 on this, really useful if you're running a hybrid environment (behind closed doors development + open tickets among users).</p>			

Lucas, I don't understand how your patch works.. How could a DB migration script make the private checkbox appear checked/unchecked when you're creating a new issue? Shouldn't you change the Issue model? Sorry if it's a stupid question, I'm a ruby noob, but I can't really understand it.

PS.: I see the redmine logs when inserting a new issue, and the value "false" is sent to the database. Am I crazy?

Bruno Medeiros wrote:

Lucas, I don't understand how your patch works.. How could a DB migration script make the private checkbox appear checked/unchecked when you're creating a new issue? [...]

The patch sets a default value for the @:is\_private@ attribute of the @:issues@ column.

Mischa The Evil wrote:

The patch sets a default value for the @:is\_private@ attribute of the @:issues@ column.

Ok, I got it, but it doesn't seem to change the default behavior. The 'false' comes from the application, it's not false because of the database default value, AFAIK.

Bruno Medeiros wrote:

Mischa The Evil wrote:

The patch sets a default value for the @:is\_private@ attribute of the @:issues@ column.

Ok, I got it, but it doesn't seem to change the default behavior. The 'false' comes from the application, it's not false because of the database default value, AFAIK.

Bruno, I'm currently not in the position to test this patch, but since it's a DB-migration script you should run @RAILS\_ENV=production rake db:migrate@ after creating the "patch"-file under @/db/migrate@. That should do the trick...

Mischa The Evil wrote:

Bruno, I'm currently not in the position to test this patch, but since it's a DB-migration script you should run @RAILS\_ENV=production rake db:migrate@ after creating the "patch"-file under @/db/migrate@. That should do the trick...

Sorry for the misunderstanding, I couldn't believe that a script, which is supposed to alter the database, could also change the default value on the view form, before the entry reach the database. Rails is really impressive on this!

**I can confirm that the script works great, thanks all!**

Bruno Medeiros wrote:

I can confirm that the script works great, thanks all!

Bruno (or anyone else), can you comment on whether this enables private=true for users who don't have permission to use it?

For example, if the customer role does not have permission to set an issue private, preventing the checkbox from appearing on forms they see, and private=true is the default for new issues, will any new issues filed by a customer also default to private=true?

My goal is for all internal roles (developers, managers, etc.) to see the checkbox, checked by default, while customers do not see the checkbox, and all issues they file are public.

So I ran this locally to see if I could answer my own question. I'm a little saddened by the results: All new issues have the Private option checked by default.

All, including new issues from users without the right to set issues private. Even though the check box isn't visible, it's still defaulting to a value of true. While this isn't devastating, it's a little less handy than I originally had hoped. Our developers can file issues without worrying about forgetting to keep them private, but externally filed issues will need to be reviewed and made public.

Hey folks, this isn't much, but I went ahead and took the patch approach to solve this one for my needs...

In `app/controllers/issues_controller.rb` I added the following line toward the end of the `@def build_new_issue_from_params@` block (around line 410 in the case of v2.1.2):

```
@issue.is_private = true if User.current.allowed_to?(:set_own_issues_private, @project)
```

This is meant to work without using the db migrate script. Here's my setup for this:

**Split your roles in two sets, some external (customer), some internal.**

**Allow internal roles to view all issues, public and private, and restrict external roles to non-private issues.**

**Allow internal roles to set their own issues private, but do not give this right to externals.**

In place, this makes all new issues filed by internal users private, while all new issues filed by customers must be public. It's not perfect, though: for internal roles, new issues will always be created private, even if you uncheck the box on the form, requiring the filer to then update the issue to make it public.

Since unchecking the box does work when using the db migrate script, it might be possible to do a hybrid of the two...using the script the set the default value to private, and then changing the patch to set the value to false if the permission does NOT exist. If there's a way to do that, it would allow internal users to uncheck private when desired, and external users don't need that flexibility since the option won't exist for them anyway...

**But I have no doubt there are simpler solutions. I'm just a total noob.**

Yeah, so, simpler solution:

```
app /views /issues_form.html.erb
```

In the very first block (for the `is_private` safe attribute), right around line 6, change:

```
<%= f.check_box :is_private, :no_label => true %>
```

to:

```
<%= f.check_box :is_private, :checked => @issue.new_record? || @issue.is_private, :no_label => true %>
```

Super simple, right? XD It's already wrapped in a check for permission to set the attribute.

No other modifications necessary. Sigh, lol. I've learned way too much about Redmine during this effort!

**This patch exactly meets my needs, no caveats so far. This guarantees that external users without set-private rights will always post public issues, and internal users will always default to private, with the freedom to uncheck the box.**

Joshua DeClercq, I'm happy you could manage your problem, but what I wanted for my needs was really the first patch. I want to have all issues private, even if the user don't have

rights to change. What is confusing here is that there is no permission to "make issue private", there is a permission to "change private/public flag". So, if the user doesn't have this permission, the issue is public by default. I wanted to change that: If the user doesn't have this permission, the issue is private.

Bruno, understood. For my case, and a few others I was pressured into figuring this out for, the idea was for customers to have shared visibility, while internal issues were internal unless management decides to make them visible to customers.

If issues are always private, even for those who don't have the permission, there's no shared visibility between this class of users, since the only private issues that would be visible to them would be the ones they're involved in.

If, however, there were an option to extend 'involvement' to members of your group, if you're a member of the project via that group, that would be pretty cool and achieve the same result.

Ultimately, though, I prefer a solution that doesn't involve a DB migration. A code tweak just requires a server restart, and makes me a lot less nervous.

additionally: It would be even nicer, if one could set a default value for the "private" flag depending of the tracker used.

e.g. I am using an "Administrative Tasks" Tracker that handles tasks beside the regular software development process in our Redmine. However, I would prefer having those tasks within this Tracker private by default.

---

What you're describing might be better suited to a plug-in. It would need to add an extra field to each tracker's settings, and then that field would need to be checked against with whatever routine runs to refresh the form when you change the tracker, as well as making the usual permission checks.

It's a compelling idea, but it feels like it's only half of a good user experience. Once you can set trackers to private/public by default, users will want the option to hide private-by-default trackers from members who don't have privacy permissions, so they can't create any of those issues in the first place. And THAT is much more complicated.

+1. It would be a very useful feature for absent minded people like me.

+1 for this

@Joshua I have tested out your code and it is great . I am looking to do the same thing for Private Notes too . I have no experiance with Ruby on Rails . Can u please help me.

My use case is almost same as yours . Private notes should be checked by default for users having "Set notes as private" privilege.

---

@jai -- I don't have a 2.x server to play with right now, so I can't say for sure if this works. You'll want to look at /app/views/issues/\_edit.html.erb though, right around line 34, which in stock 2.2 reads:

```
<%= f.check_box :private_notes, :no_label => true %> <%= I(:field_private_notes) %>
```

You should be able to do something very similar to what I did before, adding to the check\_box definition so it ends up something like:

```
<%= f.check_box :private_notes, :checked => true, :no_label => true %> <%= I(:field_private_notes) %>
```

Since this line is preceded by a permission check, it should only affect users allowed to post private comments. Since the check box doesn't appear when editing an existing comment later, there's no need for some kind of 'new comment' check, either (like with the private issue earlier), and a simple ':checked => true' should suffice.

Let me know if this works for you!

The original issue describes wanting to use these settings on a per-project basis. Here's one way to do it, step by step...

1. Start with a db migration: You need a place to store the privacy settings. You'll want to create a new /db/migrate/\*.rb file, let's say 20130226022500\_add\_default\_privacy\_checkboxes.rb. You can have either or both private issues/comments checked by

default:

```
class AddDefaultPrivacyCheckboxes < ActiveRecord::Migration
  def self.up
    add_column :projects, :private_issues_is_checked, :boolean, :default => false, :null => false
    add_column :projects, :private_notes_is_checked, :boolean, :default => false, :null => false
  end
end
```

```
def self.down
  remove_column :projects, :private_issues_is_checked
  remove_column :projects, :private_notes_is_checked
end
end
```

It's pretty readable, so you can figure out which lines to remove if you don't want both values. After this, from your console run:  
rake db:migrate RAILS\_ENV=production

Now you get to edit a few files.

1. /app/views/issues/\_form.html.erb From comment 22 above, this is where you get the private issue box checked by default. We're adding an extra check to it:

```
<%= f.check_box :is_private, :checked => (@issue.new_record? && @project.private_issues_is_checked?) || @issue.is_private? :no_label => true %>
```

2. /app/views/issues/\_edit.html.erb From comment 29 right above this, where I explain how to check private comment by default. Instead of :checked => true, you want the following:

```
:checked => @project.private_notes_is_checked?
```

3. /app/views/projects/\_form.html.erb Now to add check boxes to the project Settings page. Around line 17 you'll see the code to create a checkbox for :is\_public. Insert a new line (or two) after it, for the new setting(s):

```
<%= f.check_box :private_issues_is_checked %>
```

```
<%= f.check_box :private_notes_is_checked %>
```

4. /config/locales/en.yml Those checkboxes need labels, so add them. Around line 300, you'll find a block of all the field\_\* messages. You need to add two new messages for field\_private\_issues\_is\_checked and/or field\_private\_notes\_is\_checked. For example:

```
field_private_issues_is_checked: Private issues by default
field_private_notes_is_checked: Private notes by default
```

5. /app/models/project.rb Add the new fields to the safe\_attributes list, otherwise Redmine won't save changes to the check boxes on the project Settings page. These are around line 632. Append or insert the field names, making sure to add comma separators where necessary. My final looked like:

```
safe_attributes 'name',
  'description',
  'homepage',
  'is_public',
  'private_issues_is_checked',
  'private_notes_is_checked',
  'identifier',
  'custom_field_values',
  'custom_fields',
  'tracker_ids',
  'issue_custom_field_ids'
```

Restart Redmine, and that does it. You now have a project-by-project setting for whether issues and/or comments are checked 'Private' by default for all users with permission to

set it private. Users without permission to do so will still only post publicly.

Thanks Joshua

It worked perfectly well . Just the kind of feature i was looking for.

Hello Joshua,

Is it possible to put these changes in a plugin ? In case if I want to upgrade my redmine,plugin will be better to port these changes to new redmine instance right?

I am thinking about makig a plugin , but dont know if its even possible.

Maybe. I have zero experience building plugins. I prefer to tinker and play in the code directly. If I ever figure plugins out, I'll probably use something simple like this as a practice run.

From what I've seen, plugins can become very buggy after a Redmine upgrade, potentially breaking itself or large chunks of Redmine. I've come to like having a list of manual tweaks to apply after an upgrade instead, because I can adapt to new code as I go. I realize this is NOT an option for everyone, though, so I'll see what I can do.

Joshua DeClercq wrote:

The original issue describes wanting to use these settings on a per-project basis. Here's one way to do it, step by step...

Why wouldn't you create a patch together with a pull-request on github for such a useful feature?

-> <https://github.com/redmine/redmine>

Here's a way to set new issues private by default on a tracker-specific basis (as opposed to for an entire project). I've made a few small changes to Joshua's instructions in comment #30 above to accomplish this.

1. Start with a db migration. Create a new `/db/migrate/*.rb` file, e.g., `20140801_add_default_privacy_checkbox.rb`.

```
class AddDefaultPrivateCheckbox < ActiveRecord::Migration
  def self.up
    add_column :trackers, :private_issues_is_checked, :boolean, :default => false, :null => false
  end

  def self.down
    remove_column :trackers, :private_issues_is_checked
  end
end
```

After this, from the command line run: `rake db:migrate RAILS_ENV=production`

1. `/app/views/issues/_form.html.erb` Change the `:is_private` code to this:

```
<%= f.check_box :is_private, :checked => (@issue.new_record? && issue.tracker.private_issues_is_checked?) || (!@issue.new_record? && @issue.is_private?), :no_label => true %>
```

2. `/app/views/trackers/_form.html.erb` Add the checkbox to the tracker form. Insert after the `:is_in_roadmap` line.

```
<%= f.check_box :private_issues_is_checked %>
```

3. `/config/locales/en.yml` Add the following line to the section with all the `field_*` messages (e.g., line 324)

```
field_private_issues_is_checked: Private issues by default
```

4. /app/models/project.rb Add the new field to the safe\_attributes list (around line 632). It should look something like:

```
safe_attributes 'name',
'description',
'homepage',
'is_public',
'private_issues_is_checked',
'identifier',
'custom_field_values',
'custom_fields',
'tracker_ids',
'issue_custom_field_ids'
```

5. Reboot Redmine for the changes to take effect.

Thanks to Nate and Joshua for sharing your solutions.

A couple of notes further to Nate's post... I've rolled up his described steps into a single patch that worked for me. I had to change 20140801\_add\_default\_privacycheckbox.rb to 20140801\_add\_defaultprivate\_checkbox.rb. I'm not entirely sure why the migration failed because of the difference is name- perhaps it has a magic meaning in Rails (noob question?!)

I also was running the Redmine Tracker Control Plugin ([http://www.redmine.org/plugins/redmine\\_track\\_control](http://www.redmine.org/plugins/redmine_track_control)), and as this plugin overrides the form partial a further amendment was required- which I've added in another patch named: add\_default\_private\_checkbox\_\_with\_redmine\_track\_control\_plugin.patch

I'll be interested to hear how people get on applying the patch. Of course, don't forget to run rake db:migrate RAILS\_ENV=production ...and reboot redmine. I just restart apache (with Passenger), which make this good to go!

Note: to apply the patch, download the patch to your base redmine directory and run:

```
patch -p1 < add_default_private_checkbox__core.patch
```

Or:

```
patch -p1 < add_default_private_checkbox__with_redmine_track_control_plugin.patch
```

Cheers!

Hi, I wrote a plugin for this feature:

```
https://github.com/luisr9/redmine\_private\_issues
```

Steps

1. Install plugin.
2. go to setting of plugin and check "enabled".

After enabled the plugin, all new issue will be private.

I tested the plugin with redmine 3.1 and 3.2. I don't know if it works and other versions.

Note: If a user will update a non private issue, the plugin will change to private the issue.

Hi,

I updated the patch from comment 38 as it contained one error.

regards

Hi Philip,

What version of Redmine are you running? None of the patches seem to work on 3.3.3.stable

## Thanks

Luis Roa wrote:

Hi, I wrote a plugin for this feature:

[https://github.com/luisr9/redmine\\_private\\_issues](https://github.com/luisr9/redmine_private_issues)

....

Just tested it.

On my point of view, it's too global

Improvement ideas :

- It would be great that default choice would be private, but if I uncheck checkbox, it goes public
- Edition not working : that means that I'm unable to make public any ticket
- It would be great that enabling would be each project scoped

Besides, a great behavior would be checkbox linked to real result.

Currently, even if it's unchecked, issue are posted as private

---

related\_issues

relates,Closed,7414,Private issues

relates,New,17132,add ability set "private note" visibility for specific role

relates,New,12089,Hide Issue History

relates,New,13522,Private field is shown as required

duplicates,Closed,14390,Notes private by default

duplicates,Closed,20102,Possibility to set new issues in private by default

duplicates,Closed,24636,All issues private - Setting by project

---

## 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