Redmine - Feature #9432

Default value for the private issue flag

2011-10-17 11:31 - Martin G

Status:	New	**Start date:**	2011-10-17
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Issues	**Estimated time:**	0.00 hour
Target version:	Candidate for next major release		
Resolution:			

Description

It would be useful to have the possibility, to set the default value for the "private issues"-flag for each project.

Related issues:

Related to Redmine - Feature #7414: Private issues	**Closed**	**2011-01-22**
Related to Redmine - Feature #17132: add ability set "private note" visibilit...	**New**	
Related to Redmine - Feature #12089: Hide Issue History	**New**	
Related to Redmine - Defect #13522: Private field is shown as required	**New**	
Has duplicate Redmine - Feature #14390: Notes private by default	**Closed**	
Has duplicate Redmine - Feature #20102: Possibility to set new issues in priv...	**Closed**	
Has duplicate Redmine - Feature #24636: All issues private - Setting by project	**Closed**	

History

#1 - 2011-10-17 15:55 - Etienne Massip

- Category set to Issues

#2 - 2011-10-17 19:04 - ruslan sharipov

- Status changed from New to Resolved

- % Done changed from 0 to 100

#3 - 2011-10-17 19:28 - Etienne Massip

- Status changed from Resolved to New

- % Done changed from 100 to 0

#4 - 2011-10-22 13:42 - liku xiong·

- % Done changed from 0 to 10

#5 - 2011-10-22 13:49 - liku xiong·

- % Done changed from 10 to 100

#6 - 2011-10-28 09:04 - Etienne Massip

- Target version set to Candidate for next major release

#7 - 2011-12-06 23:38 - Bruno Medeiros

- % Done changed from 100 to 0

I tried one more time to create a patch for this, but failed :(

There is a mechanism to set safe_attributes, and I could understand it because I'm ruby/rails noob. If someone have a clue, please post it!

#8 - 2012-02-17 10:39 - Geert Smit

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.

#9 - 2012-03-03 17:35 - (OPENTIA) Lucas J. González

- File 20120303053333_issues_is_private_by_default.rb added

Geert Smit wrote:

> It happens a lot that developers forget to choose private for internal issues.

It's just a matter of upgrading the issues model.

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.

Greetings,

#10 - 2012-03-03 17:36 - (OPENTIA) Lucas J. González

- Status changed from New to Resolved

#11 - 2012-03-03 17:45 - (OPENTIA) Lucas J. González

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.

#12 - 2012-03-05 09:45 - Etienne Massip

- Status changed from Resolved to New

#13 - 2012-04-30 20:59 - William Roush

+1 on this, really useful if you're running a hybrid environment (behind closed doors development + open tickets among users).

#14 - 2012-05-02 05:40 - Bruno Medeiros

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?

#15 - 2012-05-02 16:51 - Mischa The Evil

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.

#16 - 2012-05-09 05:39 - Bruno Medeiros

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.

#17 - 2012-05-09 16:56 - Mischa The Evil

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...

#18 - 2012-06-20 06:12 - Bruno Medeiros

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!

#19 - 2012-10-18 01:52 - Anonymous

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.

#20 - 2012-10-18 08:52 - Anonymous

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.

#21 - 2012-10-20 13:30 - Anonymous

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:

1. Split your roles in two sets, some external (customer), some internal.
2. Allow internal roles to view all issues, public and private, and restrict external roles to non-private issues.
3. 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.

#22 - 2012-10-23 05:12 - Anonymous

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.

#23 - 2012-10-30 14:47 - Bruno Medeiros

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**.

#24 - 2012-10-30 23:27 - Anonymous

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.

#25 - 2012-11-28 15:31 - Zube Zahn

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.

#26 - 2012-12-06 20:03 - Anonymous

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.

#27 - 2013-02-12 10:02 - François Bélingard

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

#28 - 2013-02-26 07:22 - Jai prakash

+1 for this

Joshua Khoele 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" privilage.

#29 - 2013-02-26 09:19 - Anonymous

Jayaprakash S -- 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:

```
<label for="issue_private_notes"><%= f.check_box :private_notes, :no_label => true %> <%= l(:field_private_notes) %></label>
```

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:

```
<label for="issue_private_notes"><%= f.check_box :private_notes, :checked => true, :no_label => true %> <%= l(:field_private_notes) %></label>
```

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!

#30 - 2013-02-26 11:06 - Anonymous

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

  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.

2. **/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 %>
```

3. **/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?
```

4. **/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):

```
<p><%= f.check_box :private_issues_is_checked %></p>
<p><%= f.check_box :private_notes_is_checked %></p>
```

5. **/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
```

6. **/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.

#31 - 2013-02-26 12:13 - Jai prakash

Thanks Joshua

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

#32 - 2013-03-19 14:48 - Jai prakash

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.

#33 - 2013-03-19 20:59 - Anonymous

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.

#34 - 2013-07-11 01:38 - Toshi MARUYAMA

- Has duplicate Feature #14390: Notes private by default added

#35 - 2013-11-26 00:23 - JayD -

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

#36 - 2014-06-22 16:26 - Toshi MARUYAMA

- Related to Feature #17132: add ability set "private note" visibility for specific role added

#37 - 2014-08-05 02:47 - Nate Swinburne

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

2. **/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 %>
```

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

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

4. **/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
```

5. **/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'
```

6. **Reboot Redmine for the changes to take effect.**

#38 - 2014-10-23 16:48 - Barry Fisher

- File add_default_private_checkbox__core.patch added

- File add_default_private_checkbox__with_redmine_track_control_plugin.patch added

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_*privacy*_checkbox.rb to 20140801_add_default_*private*_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), 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!

#39 - 2015-06-28 10:48 - Jean-Philippe Lang

- Has duplicate Feature #20102: Possibility to set new issues in private by default added

#40 - 2016-12-18 09:37 - Go MAEDA

- Has duplicate Feature #24636: All issues private - Setting by project added

#41 - 2017-01-11 10:14 - Go MAEDA

- Related to Feature #12089: Hide Issue History added

#42 - 2017-03-23 21:08 - Luis Roa

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.

#43 - 2017-12-13 00:04 - Philip Heise

- File add_default_private_checkbox__core.patch added

Hi,

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

regards

#44 - 2018-03-10 01:41 - Anudeep Junuthula

Hi Philip,

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

Thanks

#45 - 2021-06-18 14:46 - damien cuvillier

Luis Roa wrote:

> Hi, I wrote a plugin for this feature:
>
> 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

#46 - 2021-08-22 11:18 - Mischa The Evil

- Related to Defect #13522: Private field is shown as required added

#47 - 2023-04-15 08:51 - Go MAEDA

- File clipboard-202304151547-2huct.png added

- File clipboard-202304151549-spuf4.png added

- File 0001-Allow-the-default-value-of-the-private-field-to-be-c.patch added

I wrote a patch that allows you to set a default value for the private field per tracker basis.

The patch adds a "Private by default" checkbox to trackers. On the New issue page, the "Private" checkbox is checked by default when you select a tracker that "Private by default" is set. Of course, you can manually uncheck the "Private" checkbox even for such trackers.

clipboard-202304151547-2huct.png
clipboard-202304151549-spuf4.png

#48 - 2023-04-18 02:32 - Go MAEDA

- Target version changed from Candidate for next major release to 5.1.0

Setting the target version to 5.1.0.

#49 - 2023-04-22 07:59 - Go MAEDA

- Target version changed from 5.1.0 to Candidate for next major release

Go MAEDA wrote in #note-47:

> I wrote a patch that allows you to set a default value for the private field per tracker basis.

The patch has a bug that members without both "Set project public or private" and "Set own issues public or private" permissions can create private issues.

Files

20120303053333_issues_is_private_by_default.rb	271 Bytes	2012-03-03	(OPENTIA) Lucas J. González
add_default_private_checkbox__with_redmine_track_control_plugin.patch	3.64 KB	2014-10-23	Barry Fisher
add_default_private_checkbox__core.patch	2.88 KB	2014-10-23	Barry Fisher
add_default_private_checkbox__core.patch	2.99 KB	2017-12-12	Philip Heise
clipboard-202304151547-2huct.png	144 KB	2023-04-15	Go MAEDA
clipboard-202304151549-spuf4.png	74.6 KB	2023-04-15	Go MAEDA
0001-Allow-the-default-value-of-the-private-field-to-be-c.patch	5.34 KB	2023-04-15	Go MAEDA