Redmine - Defect #18832

Activity Stream Filter missing on right hand side due to permission

2015-01-13 22:53 - William Schey

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Jean-Philippe Lang		**% Done:**	0%
Category:	Plugin API		**Estimated time:**	0.00 hour
Target version:	3.0.0			
Resolution:	Fixed		**Affected version:**	2.6.1

Description

When adding a plugins events to the activity stream, the permission is declared as part of the acts_as_activity_provider call. Unfortunately, the Redmine::Activity. Fetcher (lib/redmine/activity/fetcher.rb) method, line 41, requires that permission be called the event type name with "view_" added to the front.

So, if I added the event type :dmsf_file_revision_accesses then there needs to be a permission called :view_dmsf_file_revision_accesses.

If this permission does not exist, the filter on the right hand side of the activity stream will be missing even though the permission works for the stream itself.

This does not seem to be documented anywhere and is not expected behaviour as the permission is declared as part of the acts_as_activity_provider call.

Thanks.

Regards,
Will

Associated revisions

Revision 13895 - 2015-01-18 10:07 - Jean-Philippe Lang

Fixed that Redmine::Activity::Fetcher should consider activity provider permission option if given (#18832).

History

#1 - 2015-01-14 08:24 - Mischa The Evil

- Status changed from New to Confirmed

Confirmed solely based on a quick code-review. This seems to be a remnant of the days where the use of acts_as_activity_provider with an *implicit :permission* option (in fact: "view_#{o}".to_sym) was prevailing and accepted. This has changed already a long time ago in r5325 (Redmine 1.2.0).

I think that Redmine::Activity::Fetcher#event_types should check @user.allowed_to? for three cases:

1. "view_#{o}".to_sym permission [This is currently the only checked case, it seems viable to retain it as a default; in my opinion this does not have to block the actual removal of the long deprecated feature of using acts_as_activity_provider with an implicit :permission option.]
2. a specific :permission that is declared as part of the acts_as_activity_provider call
3. a visible scope on the #{self.name} model

Are there some other things to consider?

#2 - 2015-01-14 08:37 - William Schey

Not that I can think of. This would definitely fix my issue.

Thanks Mischa.

#3 - 2015-01-18 10:10 - Jean-Philippe Lang

- Status changed from Confirmed to Closed

- Assignee set to Jean-Philippe Lang

- Target version set to 3.0.0

- Resolution set to Fixed

This should be fixed in [r13895](). If given, the permission option will be checked in order to display the item in the sidebar, instead of the default "view_*" permission.