

## Redmine - Defect #18832

### Activity Stream Filter missing on right hand side due to permission

2015-01-13 22:53 - William Schey

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Jean-Philippe Lang	<b>% Done:</b>	0%
<b>Category:</b>	Plugin API	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	3.0.0	<b>Affected version:</b>	2.6.1
<b>Resolution:</b>	Fixed		
<b>Description</b>			
<p>When adding a plugins events to the activity stream, the permission is declared as part of the <code>acts_as_activity_provider</code> call. Unfortunately, the <code>Redmine::Activity::Fetcher</code> (<code>lib/redmine/activity/fetcher.rb</code>) method, line 41, requires that permission be called the event type name with "view_" added to the front.</p> <p>So, if I added the event type <code>:dmsf_file_revision_accesses</code> then there needs to be a permission called <code>:view_dmsf_file_revision_accesses</code>.</p> <p>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.</p> <p>This does not seem to be documented anywhere and is not expected behaviour as the permission is declared as part of the <code>acts_as_activity_provider</code> call.</p> <p>Thanks.</p> <p>Regards, Will</p>			

#### 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 version#27).

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.