

Redmine - Feature #23131

Plugin load order defined by inter-plugin dependencies

2016-06-21 17:29 - Felix Schäfer

Status:	Reopened	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Resolution:	Duplicate		
Description <p>Currently plugins requiring another plugin not yet loaded will cause a hard error, forcing plugin authors to somehow name their plugins to have a name that will be sorted after the names of all the other plugins they rely on in the default sort order (alphabetical I'd guess). See source:tags/3.3.0/lib/redmine/plugin.rb#L236 for the method to require other plugins, which will throw a hard error on line 251 on Plugin.find when the plugin hasn't been loaded yet.</p> <p>This was something handled in ChiliProject, see https://github.com/chiliproject/chiliproject/commit/a24a24eb1980625fbc4558fc44f15498e883d2d9</p> <p>Would something like that be of interest to Redmine? We could take care of adapting that patch to current Redmine trunk.</p>			
Related issues:			
Related to Redmine - Patch #24939: Changing plugin loading order		New	
Related to Redmine - Patch #26439: Plugin load order defined by plugin checks		New	
Is duplicate of Redmine - Defect #6324: requires_redmine_plugin should defer ...		New	2010-09-07

History

#1 - 2016-06-22 23:41 - Mischa The Evil

- Is duplicate of Defect #6324: requires_redmine_plugin should defer loading plugins if not all dependencies are met added

#2 - 2016-06-22 23:41 - Mischa The Evil

- Status changed from New to Closed

- Resolution set to Duplicate

Felix, I think that it would be a good thing if the Redmine core would include *some* solution for this issue. Please see [#6324](#) (authored by Tim Felgentreff) and specifically the patch by Jean-Baptiste Barth in [#6324#note-8](#). Kind regards, Mischa.

#3 - 2016-06-23 10:05 - Felix Schäfer

- Status changed from Closed to Reopened

Thanks, I hadn't seen that one.

This is *not* the same requirement and solution though, as it is not possible to depend on *code* in not-yet-loaded plugins with this solution.

Example: Plugin A requires Plugin B and specifically needs lib/b.rb from that plugin to be require-able, which wouldn't work with the solution in [#6324](#).

The referenced ChiliProject commit allows for this. It is a more generic (and in my opinion better) solution to that problem.

#4 - 2016-09-19 04:32 - Michael Krupp

That is not entirely true. Requiring plugin-external libraries in a to_prepare and/or after_initialize block works just fine. See: <http://guides.rubyonrails.org/configuring.html#initialization-events>

#5 - 2016-10-20 10:26 - Jaap de Haan

Wouldn't a simple topological sort (ruby TSort) in lib/redmine/plugin.rb@self.load do the trick?

Of course provided that we can get information about the dependencies without having to load the plugins first... Chicken/Egg problem.

#6 - 2017-05-26 13:07 - Toshi MARUYAMA

- Related to Patch #24939: Changing plugin loading order added

#7 - 2017-07-17 14:17 - Go MAEDA

- Related to Patch #26439: Plugin load order defined by plugin checks added