Redmine - Patch #24939

Changing plugin loading order

2017-01-27 20:00 - Lucas Arnaud

Status:	New		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:	Plugin API		**Estimated time:**	0.00 hour
Target version:				

Description

Hi,

I have one plugin that depends on another, and I needed to specify the order plugins are loaded so the classes and migrations are loaded in the proper order, but the dependent plugin is loaded first because of alphabetical order of their names.

I made this patch on lib/redmine/plugin.rb so I can determine plugins loading order by setting its priorities on a Hash in the file config/additional_environment.rb.

This sorting relies that plugin directory name is equals to its id.

Regards,

Related issues:

Related to Redmine - Feature #23131: Plugin load order defined by inter-plugi...		**Reopened**

History

#1 - 2017-01-27 20:01 - Lucas Arnaud

I forgot to mention, I am using Redmine 3.3 version.

#2 - 2017-01-29 15:02 - Go MAEDA

- Category set to Plugin API

#4 - 2017-04-26 23:31 - Jaap de Haan

The plugin load order is a headache, I fully agree.

Instead of making a kind of ad-hoc fix, I believe the best solution involves:

- an optional declarative statement "depends_on" that can be added to the init.rb of a plugin
- a topological sort based on these dependency relationships

This has the benefit to scale and be dynamic... Your solution is overriding by giving a hard coded list... which I consider being a workaround and not a clean solution. Of course the clean solution means more efforts :-)

I'll try to come with that solution...

Relates to #23131

#5 - 2017-05-03 16:10 - Lucas Arnaud

Hi Jaap de Haan,

I know that is not the ideal solution, but it is sufficient for most of cases, and the as you already said, the best solution would take a lot of additional effort.

Thanks for considering.

Best regards,

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

- Related to Feature #23131: Plugin load order defined by inter-plugin dependencies added

#7 - 2017-08-23 11:10 - Tiago Gonçalves

We could try loading the plugin, have it check dependencies and, if they are not met, raise a specific exception to defer loading until the next plugin loading iteration. If at the end of a plugin loading iteration there isn't any progress, we know that the remaining plugins have unmet dependencies, like, missing plugins and/or circular dependencies.

#8 - 2017-09-19 00:54 - Lucas Arnaud

Hi Jaap,

I've met another use case when the topological order defined by plugin dependencies won't work.

As Redmine allows you to override views, model class methods among other things in code, the order the plugins are loaded are crucial to define the application behavior.

Sometimes one use two or more independent plugins and you need to define the order they are loaded to meet your requirements and you can't just declare one as dependent on the other.

Regards,

Jaap de Haan wrote:

> The plugin load order is a headache, I fully agree.
>
> Instead of making a kind of ad-hoc fix, I believe the best solution involves:
>
> - an optional declarative statement "depends_on" that can be added to the init.rb of a plugin
> - a topological sort based on these dependency relationships
>
> This has the benefit to scale and be dynamic... Your solution is overriding by giving a hard coded list... which I consider being a workaround and not a clean solution. Of course the clean solution means more efforts :-)
>
> I'll try to come with that solution...
>
> Relates to [#23131](#23131)

Files

0001-Making-possible-to-change-plugins-loading-order.patch	2.43 KB	2017-01-27	Lucas Arnaud