Redmine - Defect #6324

requires_redmine_plugin should defer loading plugins if not all dependencies are met

2010-09-07 22:01 - Tim Felgentreff

Status:	New	**Start date:**	2010-09-07
Priority:	Normal	**Due date:**	
Assignee:	Jean-Baptiste Barth	**% Done:**	0%
Category:	Plugin API	**Estimated time:**	0.00 hour
Target version:	Candidate for next major release		
Resolution:		**Affected version:**	

Description

requires_redmine_plugin is a nice facility to track inter-plugin dependencies, however, it is useless if I do not want (or cannot) rename plugins I depend on.

I propose that plugins are deferred and loaded at a later time, if possible.
See attached patch for a proposed solution.

Related issues:

Related to Redmine - Feature #5638: Use Bundler (Gemfile) for gem management	**Closed**	**2010-06-02**
Has duplicate Redmine - Feature #23131: Plugin load order defined by inter-pl...	**Reopened**	

History

#1 - 2010-09-08 01:14 - Tim Felgentreff

- File 0001-implement-redmine-plugin-loader.patch added

- Assignee set to Eric Davis

Remove debugger code

#2 - 2010-09-08 11:29 - Tim Felgentreff

- File 0001-implement-redmine-plugin-loader.patch added

Also, for some reason, the patch I posted doesn't apply with git am. So if you use that, here's one that does. (No code changes)

#3 - 2010-09-08 11:54 - Tim Felgentreff

- File 0001-Add-a-Redmine-plugin-locator-to-be-able-to-query-the.patch added

Just found a bug with my patch where the asset directory wasn't set properly. Changed it to not replace the rails plugin locator, but use Redmine's in addition to the default one. This probably nicer, anyway, in case people want to add locators themselves. In the new implementation, the Redmine plugin locator doesn't load anything, but is only for querying the Rails plugin search path.

#4 - 2010-10-20 19:34 - Eric Davis

- Target version set to Unplanned backlogs

#5 - 2010-11-27 00:40 - Eric Davis

- Assignee deleted (Eric Davis)

I am stepping down from working on Redmine. If someone else is interesting in working on this issue, feel free to reassign it to them.

Eric Davis

#6 - 2011-08-26 00:43 - Toshi MARUYAMA

FYI:
https://www.chiliproject.org/issues/256

#7 - 2014-01-22 13:14 - Massimo Rossello

For anyone not willing to patch the core, not requiring an end user to manage his environment.rb, I found the following workaround to be prepended on the dependent plugin's init.rb:

```
    dep = File.join(Rails.root,'plugins','<plugin name>')
    ActiveSupport::Dependencies.autoload_paths += [File.join(dep, 'lib')]
    require_dependency File.join(dep, 'init')
```

#8 - 2014-08-14 13:08 - Jean-Baptiste Barth

- File 6324_defer_plugin_requirements_evaluation.patch added

- Category set to Plugin API

- Assignee set to Jean-Baptiste Barth

- Target version changed from Unplanned backlogs to Candidate for next major release

Sorry to be late at the party, but I'd like this issue solved as it bothers me from time to time.

I find Tim's proposed implementation a bit complex (and the one integrated in chiliproject is even more complex) where all we'd have to do is defer the requirement check until the end of Redmine::Plugin.load method. Sure it's not as evolved but I think it should be enough.

Here's a proposed patch, I'd like to hear some opinions before committing it, and if you see obvious advantages to Tim's/Chiliproject's approach, let me know.

#9 - 2014-08-26 15:57 - Vincent Robert

It seems to be the simplest fix, straight-forward and easy to understand.
The other implementations, much more complex, are error-prone processes in my opinion.

#10 - 2016-02-03 18:36 - Michael Krupp

- File init.rb added

- File init.rb added

As this issue seems to make no real progress in any direction, I implemented Jean-Baptiste's Patch as a plugin.
Just save them as plugins/000_pluginloader/init.rb and plugins/zzz_pluginloader/init.rb. done.

#11 - 2016-06-22 23:41 - Mischa The Evil

- Has duplicate Feature #23131: Plugin load order defined by inter-plugin dependencies added

#12 - 2016-09-19 04:25 - Michael Krupp

Just a quick update: I merged the two plugin-files into one standalone plugin by abusing the mirror_assets method call in the redmine initializer. This plugin **does not** change the load order of plugins - it only defers the dependency check, as suggested in #6324-8

https://github.com/michaelkrupp/redmine_pluginloader

#13 - 2018-11-06 08:34 - Massimo Rossello

Following approach is simple and does work. Applied in my_plugin's init.rb to check a dependency dependency_plugin.
The check happens after all plugins have been registered.

```
Rails.configuration.to_prepare do
    Redmine::Plugin.find(:my_plugin).requires_redmine_plugin :dependency_plugin, :version_or_higher => '1.0.0'
end
```

#14 - 2023-04-25 00:31 - Takashi Kato

- File 0001-Defer-plugin-requirements-evaluation.patch added

A patch for version 5.0 or later is here.

Files

0001-implement-redmine-plugin-loader.patch	3.92 KB	2010-09-07	Tim Felgentreff
0001-implement-redmine-plugin-loader.patch	3.88 KB	2010-09-07	Tim Felgentreff
0001-implement-redmine-plugin-loader.patch	3.88 KB	2010-09-08	Tim Felgentreff
0001-Add-a-Redmine-plugin-locator-to-be-able-to-query-the.patch	4.77 KB	2010-09-08	Tim Felgentreff
6324_defer_plugin_requirements_evaluation.patch	6.97 KB	2014-08-14	Jean-Baptiste Barth

init.rb	1.39 KB	2016-02-03	Michael Krupp
init.rb	160 Bytes	2016-02-03	Michael Krupp
0001-Defer-plugin-requirements-evaluation.patch	6.74 KB	2023-04-24	Takashi Kato