Redmine - Feature #5306

Separation of core and Redmine plugins

2010-04-13 19:32 - Jean-Philippe Lang

Status:	Closed	**Start date:**	2010-04-13
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Plugin API	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

Currently, Redmine plugins are mixed with Rails plugins used by the core in /vendor/plugins.
It's not very clean and it causes the following problems:

- people somtimes move plugin directories incorrectly when upgrading
- some Redmine plugins get loaded before core plugin

A solution would be:

1. to move plugins used by the core into a new directory (eg. /vendor/core) and leave Redmine plugins in /vendor/plugins
2. to override Rails plugin loading mechanism so that Redmine plugins get loaded after all the others

Feedback is welcome from plugin developpers.

History

#1 - 2010-04-13 19:32 - Jean-Philippe Lang

- Category changed from Code cleanup/refactoring to Plugin API

#2 - 2010-04-23 02:51 - Eric Davis

I don't like the idea of putting Rails plugins or Redmine plugins in a different path. It would be even more code that we would need to support and maintain (and with Rails 3, who knows what would break). I think Radiant went this path by not using Engines and now they are trying to migrate back to how we are doing things.

What I recommend is:

1. Redmine plugins are prefixed with "redmine" - the generators already do that and legacy plugins can be renamed
2. Install Redmine plugins as Rubygems - this works really good with config/additional_environment.rb. I found one issue I'm working on and that's getting a gem's static assets to load.

Another option is to have Redmine manage the load order of it's "core" plugins itself. This is supported by Rails and would fix any loading issues. The :all option can be used to pick up all of the Redmine plugins.

```
# config/environment.rb

  # Only load the plugins named here, in the order given (default is alphabetical).
  # :all can be used as a placeholder for all plugins not explicitly named
  config.plugins = [ :core_1, :core_2, :all ]
```

#3 - 2010-04-23 08:55 - Holger Just

I'm in support of separating core Redmine plugins and user-defined ones. This would make upgrading using tar.gz files so much easier. And we would not have to abandon engines, which would be a bad idea indeed.

The seperatio could be transparently supported by moving all core-plugins to something like vendor/core_plugins and then configuring something like this:

```
# config/environment.rb

  # Insert /path/to/redmine/vendor/core_plugins at the top of the plugin load paths
  config.plugin_paths.insert(0, File.join(RAILS_ENV, "vendor", "core_plugins"))
```

From my understanding of the default plugin locator implementation in /path/to/rails/railtis/lib/rails/plugin/locator.rb, the order of the plugin load paths properly defines the load order. Using this approach, we would not have to update the environment.rb on plugin additions. This works great, if you only have to make sure that the core plugins are loaded before user plugins.

I have no idea how the RubyGem plugins have to be handled using this or the current approach.

#4 - 2011-02-21 14:38 - Ryan Cross

the approach by Holger seems like the best. The adjustment of load order by Eric could be used in addition to this approach as well if it is ever needed.

#5 - 2013-03-17 16:04 - Ivan Cenov

This issue is outdated. Redmine 2.x.x uses /plugins folder for plugins.
I propose this issue to be closed.

#6 - 2013-03-17 16:19 - Daniel Felix

- Status changed from New to Closed

Well this seems to done in the 2.x branches.
If there is any regression to close this, please reopen this issue. Otherwise this one is done.