Redmine - Feature #8420

Feature # 9609 (Closed): Upgrade to Rails 2.3.14

Consider removing #7013 workaround

2011-05-21 15:39 - Etienne Massip

Status:	Closed	**Start date:**	2011-05-21
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Code cleanup/refactoring	**Estimated time:**	0.00 hour
Target version:	1.3.0		
Resolution:	Fixed		

Description

As mentionned in config/environment.rb :

```
# TODO: Workaround for #7013 to be removed for 1.2.0
# Loads i18n 0.4.2 before Rails loads any more recent gem
# 0.5.0 is not compatible with the old interpolation syntax
# Plugins will have to migrate to the new syntax for 1.2.0
require 'rubygems'
begin
  gem 'i18n', '0.4.2'
rescue Gem::LoadError => load_error
  $stderr.puts %(Missing the i18n 0.4.2 gem. Please `gem install −v=0.4.2 i18n`)
  exit 1
end
```

Related issues:

Related to Redmine - Defect #7013: See {{count}} in page display rather than ...	**Closed**	**2010-12-01**

History

#1 - 2011-05-22 12:37 - Jean-Philippe Lang

I'm no longer sure that we should remove it.
If a new i18n release breaks the API again, Rails 2.3.11 will load it anyway.

#2 - 2011-05-22 21:32 - Etienne Massip

Isn't it the same kind of issue with Rails 2.3.11 not supporting RubyGems > 1.6 ?

What would you think of adding Bundler support (#5638) to add some dependencies check at startup such as rack, rubyGems, i18n, etc., maybe that would also prevent some support issues to be posted ?

#3 - 2011-05-28 14:41 - Terence Mill

Budnler support is adressed for next major, so can this get closed in favour for #5638 ?

#4 - 2011-05-29 09:34 - Jean-Philippe Lang

- Target version changed from 1.2.0 to Candidate for next major release

Etienne Massip wrote:

> Isn't it the same kind of issue with Rails 2.3.11 not supporting RubyGems > 1.6 ?

No, rubygems is not a gem.

> What would you think of adding Bundler support (#5638) to add some dependencies check at startup such as rack, rubyGems, i18n, etc., maybe that would also prevent some support issues to be posted ?

The only problem is i18n and the way the Rails dependency is done. In active_support/vendor.rb we have:

```
gem 'i18n', '>= 0.4.1'
```

which will load the more recent i18n version available. I think that the following would be safer:

```
gem 'i18n', '~> 0.4.1'
```

#5 - 2011-11-21 18:12 - Etienne Massip

- Parent task set to #9609

#6 - 2011-11-21 18:12 - Etienne Massip

- Status changed from New to Resolved

- Target version changed from Candidate for next major release to 1.3.0

- Resolution set to Fixed

#7 - 2011-11-29 19:02 - Jean-Philippe Lang

- Status changed from Resolved to Closed

As Rails 2.3.14 dependency is still '>= 0.4.1', I hope that i18n won't break the API in the future.