

## Redmine - Feature #783

### Real Plugin-System

2008-03-04 14:25 - Martin Herr

<b>Status:</b>	Closed	<b>Start date:</b>	2008-03-04
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	0.8		
<b>Resolution:</b>			
<b>Description</b> <p>I think it would be extremely useful to have a "real plugin-system" integrated into redmine core. Based on this, it should be possible to modify routing, controllers, models, views, helpers and even core functions.</p> <p>At the moment it's hard to build extensions without patching the core and even the engines-plugin is just an unstable workaround for me :(</p> <p>I'm one of the developers of <a href="http://forge.typo3.org">forge.typo3.org</a> and would like to contribute my redmine modifications - but I can't do this by contributing hundreds of patches. There should be a clean way for everyone to extend redmine by developing plugins.</p> <p>Maybe it makes sense to look at the mephisto guys and how they implemented their plugin-system: <a href="http://mephistoblog.com/">http://mephistoblog.com/</a> <a href="http://jamescrisp.blogspot.com/2007/04/tips-for-developing-mephisto-plugins.html">http://jamescrisp.blogspot.com/2007/04/tips-for-developing-mephisto-plugins.html</a></p> <p>Any other ideas?</p>			
<b>Related issues:</b>			
Related to Redmine - Feature #1143: Hooks for plugins		<b>Closed</b>	<b>2008-04-29</b>
Related to Redmine - Patch #1677: Plugin hook API		<b>Closed</b>	<b>2008-07-24</b>

### History

#### #1 - 2008-03-04 17:48 - Eric Davis

Actually Mephisto is in the process of throwing their plugin system away and using [Engines instead](#). You should be able to create a Redmine or Rails plugin to override anything in Redmine without changing the core.

#### #2 - 2008-03-04 18:04 - Martin Herr

Hey Eric,

I really tried to overload existing redmine-views and -controllers with engines for several days - but couldn't get it to work.. :(

Will try it again :)

@Jean-Philippe: Would it be possible that I could get write access to <http://www.redmine.org/wiki/redmine/Plugins> that I can add some documentation about my experiences writing redmine plugins?

#### #3 - 2008-03-04 18:15 - Rocco Stanzione

+1. As the Engines docs say, you can't really clobber models with a plugin - you'd have to copy the whole existing model into the plugin, which kind of defeats the purpose. I was able to get it to work for one page load with some evil hackery, but at the end Rails insists on loading the original plugin.

#### #4 - 2008-03-05 15:45 - Brian Palmer

-1.

I have to disagree. I'm working on a Redmine plugin and the engines system has worked very well, I think a "real" plugin system would require a huge amount of effort to even reach the same level. Models can be a problem, true, but it's not difficult to inject code into existing models in the plugin initialization code. I added a new has\_many association to projects in this way. Some problems can occur in development mode because of reloading, but they can be worked around.

#### #5 - 2008-03-08 18:12 - Jean-Philippe Lang

[Martin Liu](#): you should be able to edit the wiki now.

#### #6 - 2008-07-11 06:46 - Chaoqun Zou

Hi, Brian.

Could you tell us how to do this? "I added a new has\_many association to projects in this way."

Thanks.

#### #7 - 2008-07-21 17:56 - Eric Davis

chaoqun zou wrote:

Hi, Brian.

Could you tell us how to do this? "I added a new has\_many association to projects in this way."

This code is working great for me. As you can see in the ClassMethods and InstanceMethods modules, you can easily add or override methods in the Redmine core.

```
# init.rb
require_dependency 'issue_patch'

# issue_patch.rb
require_dependency 'issue'

module IssuePatch
  def self.included(base) # :nodoc:
    base.extend(ClassMethods)

    base.send(:include, InstanceMethods)

    # Same as typing in the class
    base.class_eval do
      belongs_to :my_plugin_class
    end
  end

  module ClassMethods
    # Methods to add to the Issue class
  end

  module InstanceMethods
    # Methods to add to specific issue objects
  end
end

# Add module to Issue
Issue.send(:include, IssuePatch)
```

#### #8 - 2008-10-02 02:50 - Eric Davis

- Status changed from New to Closed
- Assignee deleted (Jean-Philippe Lang)
- Target version set to 0.8
- % Done changed from 0 to 100

I'm going to close this issue. [#1677](#) added some plugin hooks so plugins can start to add in their own HTML or controller logic. [r1723](#) lets any plugin override a view by creating the same named view file.

If any more hooks are needed; open a issue describing where you would like it, what data you would like, and assign it to me.

Thanks.

#### #9 - 2015-02-04 07:24 - Thiagarajan Thulasi

Hi,

In my redmine application i added some plugins while overriding a same patch in different plugin but it take priority with the top level i unable to run my bottom plugin patch.

For Example:

```
Plugin 'A'    Am writing patch for issues controller and issue before_save callBack
plugin 'B'    Am overriding same issue controller patch and model before_save callBack
```

The above scenario Plugin 'A' patch's only running plugin 'B' code doesn't run

Any one have any solution for this?

**#10 - 2015-02-04 10:54 - Toshi MARUYAMA**

Thiyagarajan Thulasi wrote:

(snip)

FTR: [#19019](#).