

Redmine - Feature #5863

Remove All Plugins

2010-07-10 01:45 - Sam Bo

Status:	New	Start date:	2010-07-10
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Resolution:			
Description Button, program, function that would correctly and safely remove all plugins (the ones installed by the user not the system). After having gone through an upgrade I've found that the plugins cause all kinds of issues. I would like a way to safely and easily remove all the plugins and any DB changes they might have made which would help easily clean anything that was broken and make the upgrade much easier.			
Related issues: Related to Redmine - Feature #3177: Add "Check for updates" functionality to ... Closed 2009-04-15			

History

#1 - 2013-03-18 14:05 - Ivan Cenov

Removing plugins is pretty easy done by hand. Just move them from /plugins folder to some other folder outside Redmine's root folder and restart Redmine. No need to remove data from the database.

Upgrading process includes upgrading plugins too. If there is no upgraded plugins at the time of upgrading core Redmine, just move the plugin outside and leave data in the database. When a plugin gets updated by its author, copy it into /plugins folder and perform migration of its data.

I think this is the safest way to upgrade plugins asynchronously without losing its data.

#2 - 2013-03-18 14:23 - Daniel Felix

- Due date set to 2013-04-01

- Status changed from New to Needs feedback

Yes this would be some kind of solution. The other way would be to migrate the plugin to version 0, which will truncate all your unneeded data from the database too.

Please give feedback until 1st April, otherwise this issue will be closed.

Best regards,
Daniel

#3 - 2013-03-18 19:54 - Ivan Cenov

Daniel Felix wrote:

Yes this would be some kind of solution. The other way would be to migrate the plugin to version 0, which will truncate all your unneeded data from the database too.

Please give feedback until 1st April, otherwise this issue will be closed.

Best regards,
Daniel

If migration to 0 is performed plugins' data will be lost. I can't imagine a reason to delete them. They may live there waiting for good times in the future.

#4 - 2013-03-18 19:55 - Ivan Cenov

Instead, another idea comes in my mind but I think this should Ruby on rails issue, not Redmine. Why not each plugin has a flag, par example 'active' If this flag is set, plugin is activated. (Something like OS services - active, not active, disabled ...)

#5 - 2013-03-18 20:05 - Daniel Felix

Ivan Cenov wrote:

If migration to 0 is performed plugins' data will be lost.

As described above. ;-)

I can't imagine a reason to delete them. They may live there waiting for good times in the future.

Well as there is just useless dump in the database. If I'm sure that those plugins won't come back, I delete them to have a clean database.

#6 - 2013-03-18 20:07 - Daniel Felix

Ivan Cenov wrote:

Instead, another idea comes in my mind but I think this should be a Ruby on Rails issue, not Redmine. Why not each plugin has a flag, for example 'active' If this flag is set, plugin is activated. (Something like OS services - active, not active, disabled ...)

I had the same idea as I described in the message above. But otherwise there must be a restart, a migrate etc. this is cleaner if the user runs this in his commandline.

#7 - 2013-03-19 15:30 - Dipan Mehta

Actually I do use and experiment a lot of plugins - and sometimes if things don't work you need to hunt down where the issue is.

Many systems such as Drupal, GLPI, allow activating or deactivating the plugin - that way it can be much easier than logging in to the server and moving the plugin folder to a temporary place and bringing it back if you need!

#8 - 2013-03-19 17:05 - Ivan Cenov

Dipan Mehta wrote:

Actually I do use and experiment a lot of plugins - and sometimes if things don't work you need to hunt down where the issue is.

Many systems such as Drupal, GLPI, allow activating or deactivating the plugin - that way it can be much easier than logging in to the server and moving the plugin folder to a temporary place and bringing it back if you need!

I agree with you. It would be better to have a systematic way to enable/disable plugins. Also, it should be possible to activate Redmine without plugins nevertheless they are activated or not. This is similar to the 'safe' mode of Windows for example.

#9 - 2013-03-19 17:35 - Daniel Felix

Yes, this idea of a safe mode would be good. Or maybe a way that Redmine has a small bootstrap.

Just an example:

1. Redmine starts
2. Administrator searches for a plugin and installs this
3. Redmine could be rebooted by webinterface.
4. Redmine runs tests, which are included in the plugin
5. Redmine checks if the latest installed plugin has any impact on performance/stability
6. If the tests fail or maybe the stability shows some problems, the plugin will be marked as critical and will be deactivated. Redmine will reboot after this, without this plugin
7. If there is no issue. Everything is okay and the plugin will be marked as stable.
8. If the plugin is marked as critical, the administrator has the right to go to the plugin backend, select this plugin and set it to "forced load". Which will ignore the critical state, restart Redmine and will use this plugin. This could cause that Redmine won't run correctly (the same behaviour as if you install an outdated plugin currently in your plugin folder)
9. The plugin backend should allow to upgrade the current plugins and hold back the old (stable) version of the plugin, which gives the admin the ability to revert the update by webinterface.
10. In case, that there is a critical bug, which prevents Redmine from starting, there is the safe mode, which could be accessed and give the admin the ability to access Redmine without any plugins. This could be good, if the admin wants to disable a certain plugin, which he previously has set to forced.

These are just some ideas how this could be improved.

I've had a much bigger idea, learned from TYPO3. But this needs much more improvements.

Something like the TER from TYPO3. A global server (maybe with some mirrors), which holds the different versions of a Redmine plugin. The server just serves those plugins as tar.gz or .zip files.

Redmine has a backend which could search the server (via json) and get the plugins and their versions. The administrator chooses which plugin should be installed and Redmine receives them via wget or whatever to the local server. After that, Redmine just unzips these archives and restarts (as described in the example) above.

But this needs much more man power, another server, some handwork of a TER admin or maybe some TER admins. And finally... The help of each plugin developer, to upload a dedicated version of his plugin to this server.

#10 - 2013-03-19 19:08 - Dipan Mehta

11. The logs can be visible (or downloadable) in the web interface. I use [plugin logs](#) for that, but ideally it should be the key feature of Redmine.

But,

Something like the TER from TYPO3. A global server (maybe with some mirrors), which holds the different versions of a Redmine plugin. The server just serves those plugins as tar.gz or .zip files.
Redmine has a backend which could search the server (via json) and get the plugins and their versions. The administrator choose which plugin should be installed and Redmine receives them via wget or whatever to the local server. After that, Redmine just unzip these archive and restarts (as described in the example) above.

Now this is quite big really. I think the trouble is that most plugins are not well managed in terms of versions - many original contributors no longer active and then you see some half a dozen forks in github trying to figure out which would work - along side which others on which versions of plugin and which Redmine version!

This is not quite straight forward. But never the less if I download a .zip / .tar.gz file from somewhere and manage plugin as you describe in step 1 - 10 still is a lot of help.

#11 - 2013-03-20 05:22 - Mischa The Evil

Daniel Felix wrote:

Yes, this idea of a safe mode would be good. Or maybe a way that redmine has a small bootstrap.

These are just some ideas how this could be improved.

I don't see something like this getting implemented soon. Too much troubles to overcome which are due to the Rails nature quite hard to get right. For example:

3. Redmine could be rebooted by webinterface.

Considering the amount of ways to deploy Redmine (or in general: Rails apps) this is quite difficult to develop.

4. Redmine runs tests, which are included in plugin

Most plugins don't come with tests.

5.Redmine checks if the latest installed plugin has any impact on performance/stability

Considering the amount of test frameworks for Rails, this could become a burden too.

9. Those plugin backend should allow to upgrade the current plugins and hold back the old (stable) version of the plugin, which gives the admin the ability to revert the update by webinterface.

This is difficult with the current plugin API and plugin implementation in general.

Just my two cents...

#12 - 2013-03-20 15:52 - Dipan Mehta

I agree with Mischa on each of the above points. I think, a very simple thing to begin with would be to have following:

1. uplaod the .tar.gz or zip through web interface : equivalent of upload and move the folder.
2. install : amounts to plugin:migrate rake tasks running in background.
3. configure : as it is right now
4. activate / deactivate button : this is equivalent to redmine skipping the directory even if it is present.

That much would be enough to start with. It would be nice to have uninstall (migrate to VERSION=0) as well, however, I can still live without it.

If I am not wrong, this can be achieved through external set of script without even changing the API - but I might just be wrong.

My point is that - in order to get max out of this feature we should start with least initial requirements that can be implemented usefully.

#13 - 2013-03-25 04:01 - Dipan Mehta

Add related - [#3177](#)

#14 - 2013-03-25 07:21 - Ivan Cenov

Dipan Mehta wrote:

1. activate / deactivate button : this is equivalent to redmine skipping the directory even if it is present.

Does this need restart of Redmine?

Also, an administrator should be sure that no any other user uses Redmine at the moment of deactivating/activating plugin. There is another plugin, https://github.com/edavis10/redmine_maintenance_mode

#15 - 2013-04-10 12:17 - Daniel Felix

- Due date deleted (2013-04-01)

- Status changed from Needs feedback to New

Removed due date. This one still needs some further investigation.

#16 - 2013-04-10 13:23 - Dipan Mehta

Ivan Cenov wrote:

Dipan Mehta wrote:

1. activate / deactivate button : this is equivalent to redmine skipping the directory even if it is present.

Does this need restart of Redmine?

No. I think the plugin mechanism/API should be such that it shouldn't require Redmine to restart. This is quite well achieved in say [Drupal](#) or [GLPI](#) achieves this. Basically, when the plugin is de-activated, all hook calls registered by the plugin are dropped and not executed. This would be a great help when we want to troubleshoot which plugin/hook is causing specific behavior!

#17 - 2013-04-10 13:50 - Dipan Mehta

Mischa The Evil wrote:

Daniel Felix wrote:

3. Redmine could be rebooted by webinterface.

Considering the amount of ways to deploy Redmine (or in general: Rails apps) this is quite difficult to develop.

If we think about it, restart of Redmine essentially an equivalent of restart of the webserver it is hosting- be it Webbrick, Apache or any other. The trouble is, Redmine doesn't know the settings of webserver to be re-started from the browser! There is some situation, touch tmp/restart.txt might do the job but I can't help when server is Apache. So I agree that the issue is

What is possible however, is the following:

1. As part of the installation, there can be a script such as 'redmine-control' or something containing commands such as '/etc/init.d/apache restart' or 'mongrel_rails restart' etc.
2. A cron job wakes up on a period and checks either a file in /tmp/restart.txt OR in DB, and if the status is true, executes redmine-control script with proper environment
3. On the web-page there is a button which (without expecting the full page to refresh) will apply the necessary command
4. After a while, the redmine will restart and after 5 mins or so and reflect the time at which Redmine has restarted.

Understand that redmine-control script needs to exclusively re-written as the installation changes so it is not a general purpose functionality. However, once done right Redmine can be restarted from the browser arbitrary number of times.

Note that we do similar cron job command setting for email-receiving.

Open issue: yes, what if you want to restart only redmine but not the whole webserver hosting other sites? Well even in the command line environment I don't know how to do this. But something in line of this, this above can also be achieved.