

SCM Creator (+Github) - Feature #1707

Git support

21 Apr 2011 20:09 - Andriy Lesyuk

<b>Status:</b>	Closed	<b>Start date:</b>	21 Apr 2011
<b>Priority:</b>	Normal	<b>Due date:</b>	28 May 2011
<b>Assignee:</b>	Andriy Lesyuk	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	0.1.0		
<b>External issue:</b>			
<b>Description</b>			
First of all <b>thanks to Bertrand Baudinet</b> for his great patch adding Git support to the plugin!			
I believe adding other repositories support is a very cool feature! However this changes much and I have some "items" to think:			
<div>1. The plugin name is to be changed perhaps (including, of course, description);</div> <div>2. File subversion.yml should support configuration for different repositories, e.g.:</div>			
<pre>production:   svnpath: /var/lib/svn   svnadmin: /usr/bin/svnadmin   gitpath: /var/lib/git   git: /usr/bin/git</pre>			
<div>This way the plugin will allow to “manage” several repositories on the same installation...</div> <div>3. Configuration class SvnConfig should be renamed (again)?</div>			

History

#1 - 22 Apr 2011 19:09 - Andriy Lesyuk

- Target version set to 0.1.0
- % Done changed from 0 to 40

#2 - 22 Apr 2011 19:10 - Andriy Lesyuk

- File deleted (add\_git\_support.diff)

#3 - 22 Apr 2011 19:11 - Andriy Lesyuk

- Status changed from Open to In Progress

#4 - 26 May 2011 21:44 - Andriy Lesyuk

Maybe even so:

```
production:
  svn:
    path: /var/lib/svn
    svnadmin: /usr/bin/svnadmin
  git:
    path: /var/lib/git
    git: /usr/bin/git
```

**#5 - 28 May 2011 11:15 - Andriy Lesyuk**

- Due date set to 28 May 2011

- % Done changed from 40 to 100

It would be great if someone tested this...

**#6 - 30 May 2011 16:14 - Jean-Sébastien Bour**

No real problem here; “destroy” doesn’t destroy the repository but it’s not related to git, it just doesn’t look implemented yet

We have a problem with file permissions on disk, but it’s related to Ruby people who seem to think that it’s cool to put “File.umask 0000 # Insure sensible umask” all around daemonize.rb, because security is for boring people I guess I think I’ll add a “chmod o-a” after the system call to git

**#7 - 30 May 2011 16:30 - Andriy Lesyuk**

No real problem here; “destroy” doesn’t destroy the repository but it’s not related to git, it just doesn’t look implemented yet

I guess this is going to remain this way... Destroying just a repository in Redmine is one thing and destroying the repo itself... not sure if this is good/wanted. Let it be “backup”...

We have a problem with file permissions on disk, but it’s related to Ruby people who seem to think that it’s cool to put “File.umask 0000 # Insure sensible umask” all around daemonize.rb, because security is for boring people I guess I think I’ll add a “chmod o-a” after the system call to git

This is interesting! Bertrand used chmod (or maybe chown - don’t remembder) also... When I tested I figured out that chmod is not needed. Where did you find this line?

**#8 - 30 May 2011 16:42 - Jean-Sébastien Bour**

Andriy Lesyuk wrote:

No real problem here; “destroy” doesn’t destroy the repository but it’s not related to git, it just doesn’t look implemented yet

I guess this is going to remain this way... Destroying just a repository in Redmine is one thing and destroying the repo itself... not sure if this is good/wanted. Let it be “backup”...

Indeed. However I thought about this from a “use auto\_\* features and let the machine handle EVERYTHING” point of view. And remember I speak on git fans behalf, so who really cares about the “server” repository when everyone has a copy?

This is interesting! Bertrand used chmod (or maybe chown - don't remember) also... When I tested I figured out that chmod is not needed. Where did you find this line?

```
~# grep -hn umask /usr/lib/ruby/gems/1.8/gems/daemons-1.0.10/lib/daemons/daemonize.rb
119:   File.umask 0000 # Insure sensible umask
176:       File.umask 0000 # Insure sensible umask
219:   File.umask 0000 # Insure sensible umask
```

and umask command ran from a shell with our redmine system account outputs 0022, while ran from a system() call I added in a Redmine controller just to see, outputs 0000...

It ends up with a repo full of files with 777 UNIX rights, which is not a real huge security flaw ATM since we use group-shared git repos and everyone is in the dev group (or, just doesn't have access to the server), but in other situations it might...

#### #9 - 01 Jun 2011 15:44 - Andriy Lesyuk

- Status changed from In Progress to Closed

Jean-Sébastien Bour wrote:

Indeed. However I thought about this from a "use auto\_\* features and let the machine handle EVERYTHING" point of view. And remember I speak on git fans behalf, so who really cares about the "server" repository when everyone has a copy?

Created issue [#1770](#)... This means that perhaps I will implement this in future...

It ends up with a repo full of files with 777 UNIX rights, which is not a real huge security flaw ATM since we use group-shared git repos and everyone is in the dev group (or, just doesn't have access to the server), but in other situations it might...

This can be (and perhaps should be) solved by setting the correct umask or whatever somewhere else (e.g. in system - I don't know). I have access to several Redmine installation using different distros and did not see such permission issue... What makes me think this is an issue not related to the plugin but to the system (even if you solve this in plugin you potentially have a security "bug" which can lead to 777 files created by other applications)!

**#10 - 09 Jun 2011 09:18 - Jean-Sébastien Bour**

I can confirm this umask problem is not a bug in the system, we have a 0022 umask set at system level, and the Redmine system user creates files with correct permissions from the shell for example. This 0000 umask is clearly set by the ruby daemonize library. Maybe you don't have a setup with this particular version and they fixed it?

However it is indeed not a problem in your plugin.

**#11 - 09 Jun 2011 14:01 - Andriy Lesyuk**

This 0000 umask is clearly set by the ruby daemonize library. Maybe you don't have a setup with this particular version and they fixed it?

I'm using Redmine from Debian repository... I guess Debian package maintainer fixed it.