

Redmine - Feature #1536

Using libsvn

2008-06-27 12:11 - Pierre Paysant-Le Roux

Status:	New	Start date:	2008-06-27
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	90%
Category:	SCM	Estimated time:	0.00 hour
Target version:	Unplanned backlogs		
Resolution:			
Description Tigris provides a high level ruby api named libsvn (subversion/bindings/swig/ruby in the sources of subversion). It could maybe simplify the way Redmine handles subversion repositories by avoiding out shelling. Have someone tested this lib ? Is this lib voluntary not used in Redmine ?			
Related issues: Related to Redmine - Defect #4448: Subversion password cleanly visible in the... New 2009-12-18			

History

#1 - 2008-06-29 10:50 - Jean-Philippe Lang

The main reason why it's not used in Redmine is that there's no binaries available for windows.
I don't know if this lib has all the necessary functionalities to be used in Redmine, can you point to the documentation of this API please ?

#2 - 2008-07-03 11:52 - Pierre Paysant-Le Roux

There is no documentation specific to the ruby bindings. [Here](#) is the documentation of the subversion API. I think that the lib has more than all the necessary functionalities the be used in Redmine. I am not convinced that maintaining a Windows compatibility for a server software like Redmine is crucial.

However, it's also possible to have an adapter based on libsvn for Unices and an adapter that uses command line calls for Windows or non libsvn users. I can try to implement this adapter to show if there is a real benefit (memory footprint, cpu or io usage, code simplicity).

Moreover, no ruby bindings exists for all api (Mercurial for example).

#3 - 2008-07-05 11:51 - Jean-Philippe Lang

For sure, it would be faster.
Having an alternate adapter that uses libsvn would be nice indeed.

#4 - 2009-05-03 19:34 - Pierre Paysant-Le Roux

- File `patch_libsvn` added

- % Done changed from 0 to 90

I finally succeed in implementing a subversion adapter based on libsvn. I wrote some non regression tests (should be completed). The adapter pass functional and unit tests of the adapter based on svn command line. At the moment, I made no benchmarking. I will write something like parsing a repository with 1000 changesets to see if there is some speed improvement.

#5 - 2009-06-17 13:46 - Pierre Paysant-Le Roux

A simple benchmark : calling `Adapter#revisions` method on a local repository with 1000 simple changesets :

	user	system	total	real
libsvn	0.150000	0.020000	0.170000 (0.230447)
svn command	8.150000	0.480000	8.750000 (8.987669)

#6 - 2009-06-17 14:38 - Jens Goldhammer

+10! Great. Please integrate it into trunk

#7 - 2009-06-23 22:54 - Jean-Philippe Lang

Looks great! I just found a small bug in the diff view, it seems you need to swap `identifier_to` and `identifier_from` in `SubversionLibsvnAdapter#diff`.

But I think we shouldn't introduce a new Repository class (eg. SubversionLibsvn) since this is not a new kind of repository, but just another way to read a Subversion repository. I think your patch could be released as a plugin that overrides the default adapter. Existing repositories will use this new adapter as well.

#8 - 2009-07-28 14:06 - Pierre Paysant-Le Roux

Jean-Philippe Lang wrote:

I think your patch could be released as a plugin that overrides the default adapter. Existing repositories will use this new adapter as well.

Because plugins are loaded first, it's hard to override methods with a plugin. Do you have an idea on how to implement that ?

#9 - 2009-07-31 02:12 - Eric Davis

Pierre Paysant-Le Roux wrote:

Because plugins are loaded first, it's hard to override methods with a plugin. Do you have an idea on how to implement that ?

You can have the plugin load the required files it needs to override. For example:

```
require_dependency 'subversion_adapter' # Might need to use redmine/scm/adapters/subversion_adapter

module Redmine
  module Scm
    module Adapters
      class SubversionAdapter < AbstractAdapter
        # Override code here.
      end
    end
  end
end
```

You can also use the module include pattern if you don't want to reopen the SubversionAdapter class.

#10 - 2009-08-17 17:56 - Pierre Paysant-Le Roux

I created a plugin. It is available on github :

<http://github.com/pplr/redmine-libsvn/tree/master>

Suggestions and beta testers are welcome !

#11 - 2009-12-20 12:28 - Jean-Philippe Lang

I've just tested your plugin and found that #properties uses infinite depth, which is wrong and horribly slow (13s on Redmine repository root). Here is a fix:

```
--- subversion_libsvn_adapter.rb      Sun Dec 20 12:31:16 2009
+++ subversion_libsvn_adapter_fix.rb  Sun Dec 20 12:28:55 2009
@@ -100,7 +100,7 @@
   def properties(path, identifier=nil)
     identifier = (identifier and identifier.to_i > 0) ? identifier.to_i : "HE
AD"
     properties = {}
-    ctx.proplist(target(path), identifier) do |path, prop_hash|
+    ctx.proplist(target(path), identifier, nil, false) do |path, prop_hash|
       properties.merge!(prop_hash)
     end
     properties
```

Apart from that, it's really faster than the current implementation.

#12 - 2010-01-14 19:25 - Pierre Paysant-Le Roux

Thanks for feedback. I pushed a bugfix for #properties.

#13 - 2010-01-14 19:26 - Pierre Paysant-Le Roux

- Status changed from New to Resolved

#14 - 2011-10-18 22:13 - Mischa The Evil

- Status changed from Resolved to New

Resolved is used for merging-purposes.

#15 - 2011-10-18 22:15 - Mischa The Evil

- Target version set to Unplanned backlogs

#16 - 2012-01-24 20:48 - Rafi Greenberg

I would like to see libsvn work for the stable/trunk versions of Redmine. Somewhere along the line the pplr plugin <http://github.com/pplr/redmine-libsvn/tree/master> broke.
I wish this would be part of the official build! Anyone else agree?

#17 - 2012-05-02 03:58 - Bruno Medeiros

I agree.

#18 - 2012-06-07 19:09 - Rafi Greenberg

Anyone interested in porting this to Redmine 2.0.x?

#19 - 2012-07-11 09:10 - Ki Won Kim

+1

#20 - 2012-12-04 19:32 - William Baum

Jean-Philippe Lang wrote:

The main reason why it's not used in Redmine is that there's no binaries available for windows.

I've actually been running the libsvn plugin on Windows for years with win32svn binaries available in the win32svn project on sourceforge: <http://sourceforge.net/projects/win32svn/> And yes, it's dramatically faster.

So if the lack of Windows binaries was the primary obstacle, can we get libsvn into the core now?

#21 - 2014-04-03 12:39 - Sen Haerens

I updated the init.rb for Redmine 2.x and the plugin seems to work fine with Redmine 2.5.1.

<https://github.com/SenH/redmine-libsvn>

Files

patch_libsvn	15.7 KB	2009-05-03	Pierre Paysant-Le Roux
--------------	---------	------------	------------------------