Subversion

pysvn-workbench

I sometimes use GUI's on my development workstation. For Linux, the best I've found so far is a very old one named pysvn-workbench, which is sufficient for my needs. However, some of the documentation is lacking.

Configuration File

The configuration for pysvn-workbench is in ~/.WorkBench/WorkBench.xml. It contains some global preferences, then a list of all projects with their configurations. When preferences are changed, the file is renamed with a .old suffix, then recreated.

In that directory is also a log file for actions (WorkBench.log) and the most recently used log message (log_message.txt).

The WorkBench.log file appears to be rotated every 100k, ie it will rename the old log file to WorkBench.log.1, then start a new one.

Using Templates

Templates sound like a great idea, but figuring out how to set them up is pretty much undocumented. I finally started looking at the source and found the following:

1. Create a directory
    1. in that directory, place one or more files with the suffix .template
2. Open PySVN
    1. Select a project
    2. Open Project | Settings from the main menu
    3. Under New Template File Folder, enter the directory you created.
3. In the future, every time you create a new file, you can use one of the templates.

Note I have not been able to figure out a way to set one template folder for all projects. You have to set the template folder for each one.

Set up standard layout

1. Create the new repository (aka project)
2. Check out the project
3. Create the following directories
    1. branches
    2. tags
    3. trunk

4. Check the project back in
5. check out the project again, but with the url/trunk
6. Always work in trunk, unless you're working on a branch. tags is for when you want to create a check point.

NOTE: one thing I want to try in the future is creating a fourth directory, stable. This will be a copy of the current stable version of the code. See below for more information.

Using the Caret (^)

As of Subversion 1.6, from a working copy, you can use the caret (^) as a substitute for the root URL of the project. Thus, if your you are in your working copy someplace, the following are equivilent. Note that even if you only checked out trunk, the tags are still accessible since it is the URL that is substituted for the caret. This can greatly reduce typing.

```
http://svn.example.com/project/trunk/subdir1
^/trunk/subdir1
```

```
http://svn.example.com/project/tags/v1.5.0
^/tags/v1.5.0
```

Creating a tag

```
svn copy http://svn.example.com/project/trunk
http://svn.example.com/project/tags/1.0 -m "Release 1.0"
```

OR

```
cd /path/to/project
svn copy ^/trunk ^/tags/1.0 -m "Release 1.0"
```

Moving a repository

- http://svnbook.red-bean.com/en/1.7/svn.ref.svnadmin.c.dump.html
- http://svnbook.red-bean.com/en/1.7/svn.ref.svnadmin.c.load.html

On the old machine

```
svnadmin dump --deltas reposname | bzip2 -c > /tmp/reposname.svn.dump.bz2
```

On the new machine

```
mkdir reposname
svnadmin create reposname
bunzip2 -c /tmp/resposname.svn.dump.bz2 | svnadmin load resposname
```

Maintaining a "stable" tag

I've always been intrigued with people who use subversion with a tag that always points to the most recent stable version. Never figured out how they do it, but thanks to one of my associates, came up with this simple action when a new version has become your most recent stable version.

1. Create a new tag. I usually use a version number
2. Delete the old stable tag if it exists. Be sure and use recursion.
3. recreate the stable tag from the new version

The following list of CLI commands will do this for you on any recent copy of subversion, assuming you are in the root of a checked out version of the project. In other words, in a cli go to your recently perfect, checked in version of your project, then run the following commands.

```
# get a list of all tags
svn ls -v ^/tags
# create a new tag for this version (ie, v3.5.1)
svn copy ^/trunk ^/tags/v3.5.1 -m "Release v3.5.1"
# delete tag stable
svn delete ^/tags/stable -m "Changing latest stable version"
# copy this version to stable tag
svn copy ^/tags/v3.5.1 ^/tags/stable -m 'Setting v3.5.1 as stable'
# list your tags again
svn ls -v ^/tags
```

In the above, I'm using the caret (^) syntax to directly modify the subversion server. You can also do the same thing by explicitly using the URL. To find the URL, issue the command:

```
svn info | grep '^URL:' | cut -d':' -f2-
```

The URL listed after the colon can be used as a replacement for the caret above. Don't forget to remove 'trunk' from the end of that, however, since you are currently working in the trunk directory.

You can now publish your subversion address as
http://svn.example.com/svn/project_name/tags/stable

If you wish, you could also simply create a new subdir on the same level as trunk and tags and do the same thing, I'm guessing (haven't tried it)

Links

- https://stackoverflow.com/questions/851377/how-to-properly-create-an-svn-tag-from-trunk
- http://svnbook.red-bean.com/nightly/en/svn.branchmerge.tags.html
- http://svnbook.red-bean.com/en/1.6/svn.basic.in-action.html