

A Short SVN Tutorial

January 4, 2009

1 Introduction

A repository is a storage area containing all files and directories under version control. The repositories can be local pathnames if all users of the repository have access to that filesystem. They can be remote repositories with different syntax. For this course your repository syntax is:

- `/home/svn/490.2009/groupname` for in department access,
So for checking out your project you need to type:
`svn checkout file:///home/svn/490.2009/groupname`
- `/username@login.ceng.metu.edu.tr/home/svn/490.2009/groupname`
for remote access via ssh.
So for checking out your project you need to type:
`svn checkout svn+ssh://username@login.ceng.metu.edu.tr/home/svn/490.2009/groupname`

In this document we will assume that two users `user1` and `user2` use the system, and the group name is `project`.

2 Creating the Initial Module

For starting to use svn, first you need to import your project. For example, if you have the following listing in your project directory:

```
$ cd /home/user1/mytree
$ ls -R
.:
src/ web/

./src:
main.c main.h Makefile

./web:
index.html
```

Then you can import your project using the following command:

```
$ cd /home/user1/
$ svn import mytree file:///home/svn/490.2009/project -m "Initial import"
Adding          mytree/src
```

```
Adding      mytree/src/main.c
Adding      mytree/src/main.h
Adding      mytree/src/Makefile
Adding      mytree/web
Adding      mytree/web/index.html
```

Committed revision 1.

You can list the files in the repository by using the command:

```
$ svn list file:///home/svn/490.2009/project
src/
web/
```

Thus, as can be seen the files under the directory `mytree` is copied under the directory `project` in the repository. To start on working on the working copy of the project, you first need to `checkout` your project.

```
$ svn checkout file:///home/svn/490.2009/project
A   project/src
A   project/src/main.c
A   project/src/main.h
A   project/src/Makefile
A   project/web
A   project/web/index.html
Checked out revision 1.
```

By the `checkout` command a working copy of your project is copied to your current directory under the directory `project`.

3 Basic Usage

Most of the time you will use the following steps:

- Update your copy (if you do not have a working copy you need to `checkout` as given in the previous section).
- Make changes to the files.
- Commit your changes.

A sample session may be as the following:

```
$ cd /home/user1/project
$ svn update
U   src/main.c
U   web/index.html
Updated to revision 6.
$ vi web/index.html
<...Make your changes...>
$ vi src/main.h
<...Make your changes...>
$ svn commit -m "changes by user1"
Sending      src/main.h
Sending      web/index.html
Transmitting file data ..
Committed revision 7.
```

4 Multiple Developers

When multiple developers work on the same project, it is likely that a file is changed by another developer until you commit your changes. Assume that in the following scenario `user1` and `user2` updated the same version of the project. They both started to modify `main.c`. `user2` commits her/his changes before `user1`. Then `user1` tries to commit her/his own changes.

```
$ svn commit
Sending          src/main.c
svn: Commit failed (details follow):
svn: Out of date: '/project/src/main.c' in transaction '9-1
```

`user1` should first update her/his working copy to resolve the conflicts.

```
$ svn update
C    src/main.c
Updated to revision 9.
```

When `user1` updates her/his copy there are four different version of `main.c`. `main.c.mine`, the file that contains the changes by `user1`. `main.c.r8`, 8th version of the file (previous version). `main.c.r9`, 9th version of the file (the version that `user2` committed). `main.c`, the file which contains both changes by `user1` and `user2`.

```
$ cat src/main.c
#include <stdio.h>

int main (int argc, char **argv)
{
<<<<<<<< .mine
    int a = 7;
    printf("%d\n", a);
=====
    int a = 5;

    printf("%d\n", a);
>>>>>>> .r9
    return 0;
}
```

`user1` should modify the portion between `<<<<<<<< .mine` and `>>>>>>> .r9` to resolve the conflict. After the conflicts are resolved `user1` proceeds as follows:

```
$ svn resolved src/main.c
Resolved conflicted state of 'src/main.c'
$ svn commit -m "Discarding the changes by user2"
Sending          src/main.c
Transmitting file data .
Committed revision 10.
```

5 Adding, Removing, Renaming a File

If you want to make a change in the tree structure, you cannot do it just by using a `commit` command. For example, `user1` creates a new file `src/project.c` and in the working directory first `adds` and then `commits` the file.

```
$ svn add src/project.c
A          src/project.c
$ svn commit
Adding          src/project.c
Transmitting file data .
Committed revision 11.
```

For creating directories you first create the directory and then `add` that directory to the repository:

```
$ mkdir doc
$ vi doc/project.doc
<...Edit your file...>
$ svn add doc
A          doc
A          doc/project.doc
$ svn commit
Adding          doc
Adding          doc/project.doc
Transmitting file data .
Committed revision 12.
```

Or you can directly create a directory in the repository:

```
$ svn mkdir lib
A          lib
$ svn commit
Adding          lib
Committed revision 13.
```

For removing files or directories you can use the `delete` command:

```
$ svn delete lib
D          lib
$ svn commit
Deleting          lib
Committed revision 14.
```

For moving files or directories you can use the `move` command:

```
$ svn move src/main.c src/Main.c
A          src/Main.c
D          src/main.c
$ svn commit
Adding          src/Main.c
Deleting          src/main.c
Committed revision 15.
```