
Subversion HOWTO

Software Development Group CLLRnet <sdg@cllrnet.ca>

2004-09-09

This document will help the understand what subversion is and how to use subversion for their own projects.

Table of Contents

Introduction	1
Copyright and License	1
Disclaimer	1
What Is SVN	1
Where To Download SVN	2
Key Features	2
A Simple Subversion Tutorial	3
Configuring Subversion for Multiple developers	3
Sample subversion commands	4
Checking out a file	4
Finding the difference between two files	4
Uploading Changes To Your Repository	4
Updating Your Working Copy	4
Checking Status Of Files	5
Other Commans	5
A. Useful Web Sites	5

Introduction

This document will introduce Subversion repository system and why should you use it. Give a sample project and walk the reader throught it. We will also show the reader how to create a repository were many users can access it, for project with multiple developers.

Copyright and License

Copyright (c) 2003, 2004 Eric Meyerhoff. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts A copy of the license is included in the section entitled "GNU Free Documentation License"

Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although it is highly unlikely that accidents will happen because of following advice or procedures described in this document, the author(s) take no responsibility for damage claimed to be caused by doing so.

All copyrights are held by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements. No liability for the contents of this document can be accepted.

What Is SVN

SVN is the project name for subversion which was developed to replace the Concurrent Version System CVS. For those people who are less familiar with CVS, CVS is a "*source control*" or "*revision control*" tool designed to keep track of source changes made by groups of developers working on the same files, allowing them to stay in sync with each other as each individual chooses. The great thing about SVN though is that it has all the same functionality as CVS, but SVN is designed to be more like a file system rather than a simple revision control.

Where To Download SVN

The main website for subversion is at <http://subversion.tigris.org> [<http://subversion.tigris.org>] and a great book called Version Control with Subversion [<http://svnbook.red-bean.com>] can be read online.

Subversion has three very good software applications that you can use.

- A basic text based interface
- A graphical interface that is embedded into your file explorer. (TortoiseSVN [<http://tortoisesvn.tigris.org>])
- Another graphical interface, but this application can be used on many Operating Systems (RapidSVN [<http://rapidsvn.tigris.org>])

Key Features

- support most current CVS features
- directories, renames, and file metadata are versioned
- commits are atomic
- Three types of server options.
 - WebDAV/ with DeltaV protocol
 - Stand alone svn server
 - Using ssh to tunnel svn connection to server
- branching and tagging are constant time operations
- client/server protocol sends diffs in both directions
- time complexity of operation is proportional to the change not the data size
- efficient handling of binary files
- parseable output

A Simple Subversion Tutorial

Example 1. Creating a Repository

Subversion stores all versioned data in a control repository to begin, create a new repository.

```
# mkdir svnroot
# svnadmin create svnroot/project_name
```

This command creates a new directory `svnroot/project_name` which contains a subversion repository. *Never edit this directory manually*

Example 2. Creating your Work Space

A common practice for a working tree look like this:

```
project_dir +-- branches
              |
              +-- modules
              |
              +-- trunk
```

- The trunk directory is where you will do most of your work. This directory should contain all your up-to-date working copy of each file needed for the project.
- The branches directory is a copy of the trunk directory but only at a point where you would build a release of your project.
- The modules directory is where you would divide your project into separate modules... (plugins) each plugin would have its own (branches/trunk) directories.

Configuring Subversion for Multiple developers

In order to let other users use your repository. The permission on the root repository directory and all the files in the root repository of your project will have to give access to all developers of that project.

Add a group to your Linux server and call this group the same name as your project. Add all developers of this project to that group. Then change the group of each directory and files in your project repository to that group.

```
chown root.group_name svnroot/project_name
```

Then you will need to change the access bit on all the files and directory in your project's repository.

```
chmod g+wS svnroot/project_name
```

When a developer uses svn locally, he must make sure that he as **umask 022** set.

When using using SVN with ssh. You should create a wrapper script so that when **svnserver** is call it set the umask 022 prior. Here is a sample svnserver script.

```
#!/sbin/sh

umask 022
/usr/bin/svnserver $*
```

Sample subversion commands

Checking out a file

```
# svn co svn+ssh://user@server.com/svnroot/project_name
```

This command get every file in your repository and store them on your local drive, creating yourself your own working copy of the project.

Once you have your own working copy of the project. You can start editing any file you want.

Finding the difference between two files

```
# svn diff filename
```

The command allows you to see the unified difference between the file in your working copy and the file in the repository.

Uploading Changes To Your Repository

```
# svn ci
```

```
# svn ci filename
```

The checkin command is used when you want to send your changes that you have made to your working copy to your repository. *No matter what command or file you edit, Your repository will not have been changed unless you execute this command.*

Updating Your Working Copy

If you work on a project with multiple developer, on occasion you will have to merge or update your working copy with their code that they have summited to the repository.

```
# svn up
```

the update command will try to merge all changes into your working copy. Sometime this is not possible and subversion will let you know by creating a file showing you the difference between the two files.

When performing a update subversion displays a status of each file it update.

Checking Status Of Files

If you want to know what files has been changed in your working copy the status command will tell you.

```
# svn status
```

This will show you a list of all your files in your working copy with a character indicating the status of this file. To a list of all status characters goto Section 4.3.1 of the Subversion book [<http://svnbook.red-bean.com/svnbook/book.html#svn-ch3-sect-4.3.1>]

Other Commans

For other svn commands go to: <http://svnbook.red-bean.com/svnbook/book.html> [<http://svnbook.red-bean.com/svnbook/book.html>] or use the help system.

```
# svn help
```

```
# svn help add
```

A. Useful Web Sites

These Web sites may be useful in getting more information on Subversion:

- Subversion Official Web Site (subversion.tigris.org) [<http://subversion.tigris.org>]
- RapidSVN Web Site (rapidsvn.tigris.org) [<http://rapidsvn.tigris.org>]
- TortoiseSVN Web Site (tortoisesvn.tigris.org) [<http://tortoisesvn.tigris.org/>]