Subversion

What is it?

- Subversion is a open-source version control system similar to CVS, but with more functionality.
- Subversion manages files and directories over time, which means you can always go back to an older version of your code.
- A tree of files is placed into a central repository.
- The repository is much like an ordinary file server, except that it remembers every change ever made to your files and directories.

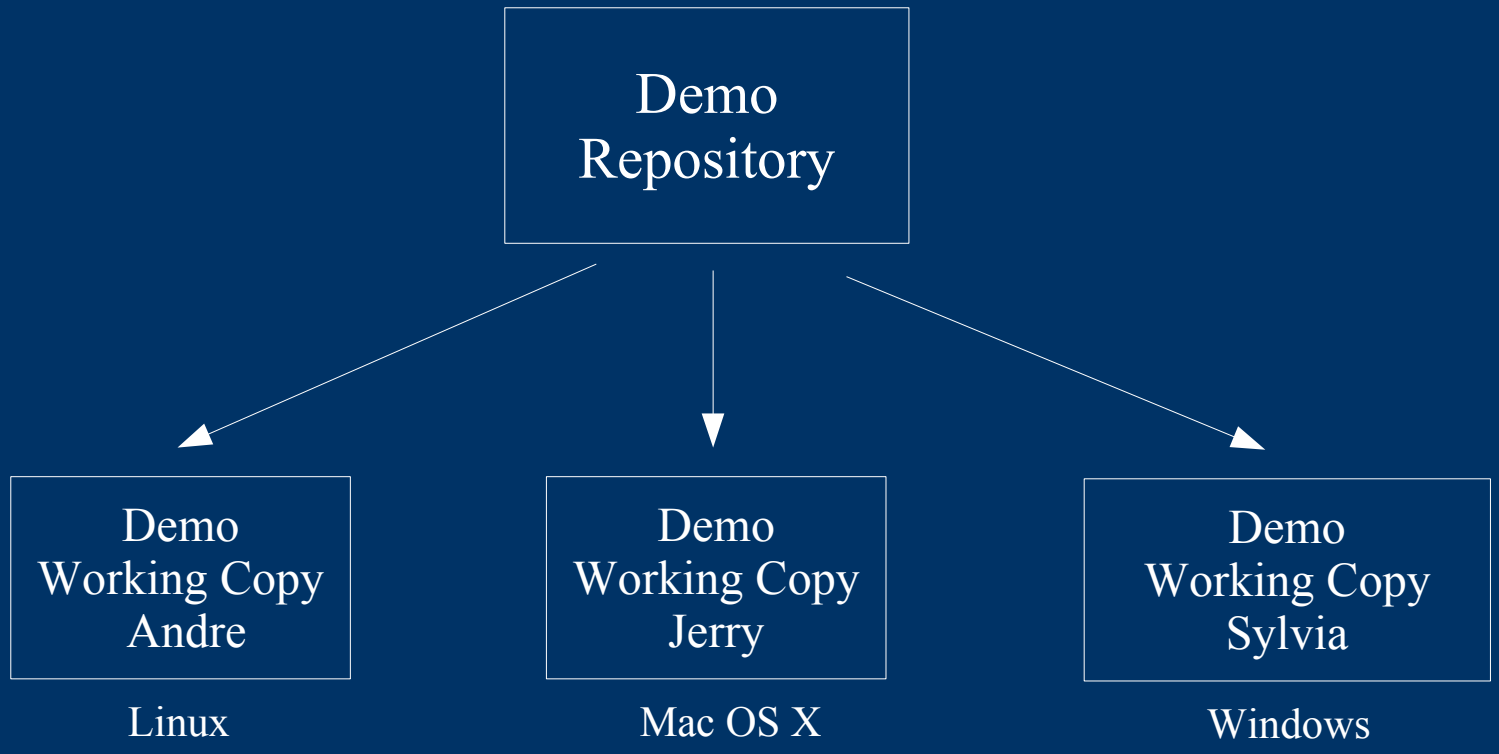
Subversion
Differences with CVS

- Directories, renames, and file meta-data are versioned
- Choice between 'Copy-Modify-Merge' and 'Lock-Modify-Unlock' Solution (from v1.2)
- Commits are truly atomic
- Apache server option (WebDAV/DeltaV protocol)
- Branching and tagging are cheap (constant time) operations
- Efficient handling of binary files (only diffs are handled)

Subversion

The Repository – An Example

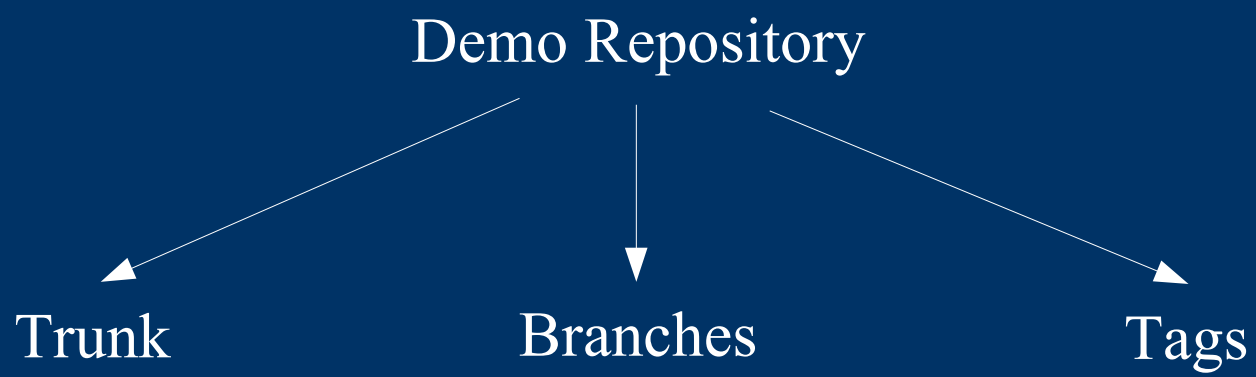
http://repository.eng.utep.edu/svn/demo



Subversion

Proposed Directory Structure

- Trunk – main line of development.
- Branches – trying out new things on copies of the trunk code.
- Tags – released code (stable version, e.g. 1.0.0, 2.3.1, etc).



Subversion
Clients

Linux

- Command Line Client (CLI)
- RapidSVN (Java-based GUI Browser)
- KSvn (Konqueror integration)

Windows

- TortoiseSVN (Explorer Integration)
- RapidSVN (GUI Browser)
- AnkhSVN (VB .NET plugin)

Mac OS X

- Command Line Client (CLI)
- SCPlugin (Finder Integration)

Multi-Platform

- Web Browser (HEAD only)
- Subclipse (Eclipse plugin)

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn import

Uploading your files to the repository for the first time.


```
/home/andre/dev/demo/readme.txt      http://.../svn/demo/trunk/readme.txt
/home/andre/dev/demo/mycode.c        http://.../svn/demo/trunk/mycode.c
```

```
svn import demo http://repository.eng.utep.edu/svn/demo/trunk
-m “Initial import of demo files”
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn export

Downloading files from the repository to your local workstation. No history information is kept in your local copy (no .svn subdirectories).


```
/home/andre/dev/demo2/readme.txt
/home/andre/dev/demo2/mycode.c
```

```
http://.../svn/demo/trunk/readme.txt
http://.../svn/demo/trunk/mycode.c
```

```
svn export http://repository.eng.utep.edu/svn/demo/trunk ./demo2
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn checkout

Checking out files from the repository to your local workstation.
History information is kept with your local working copy (.svn subdirectories).


```
/home/andre/dev/demo3/readme.txt
/home/andre/dev/demo3/mycode.c
```

```
http://.../svn/demo/trunk/readme.txt
http://.../svn/demo/trunk/mycode.c
```

```
svn checkout http://repository.eng.utep.edu/svn/demo/trunk ./demo3
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn status

Finding out what the status is of files in your working copy as compared to the files in the .svn subdirectory (the last update from the repository). If the '-u' parameter is used, the repository is checked for updates.

Andre's Linux Workstation

```
svn status <working copy> (after modifying mycode.c)
M       mycode.c


svn -uv status <working copy> (after modifying mycode.c)
M                2        2 akerstens    mycode.c
                 2        2 akerstens    readme.txt
                 2        2 akerstens    .
Status against revision:      2
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn commit

Sending your local changes to the repository. Only changes are send, not complete files. The .svn subdirectories are updated with the latest information as well.


```
/home/andre/dev/demo3/readme.txt
/home/andre/dev/demo3/mycode.c      http://.../svn/demo/trunk/mycode.c
```

```
svn commit <working copy> -m “Changed the hello string in mycode”
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn update

Getting the latest changes from the repository merged into your local working copy. If local files have been changed, this can create conflicts or a local/remote merge. The .svn subdirectories are updated with the latest information as well.

`/home/andre/dev/demo4/mycode.c` `http://.../svn/demo/trunk/mycode.c`

```
svn update <working copy>
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn resolve

Letting the repository know that a local conflict has been resolved. This should always be followed by a commit to make sure that the repository is updated with the solution as well. This process involves communication with your co-developers!


```
/home/andre/dev/demo3/mycode.c.mine
/home/andre/dev/demo3/mycode.c.rOLDrev
/home/andre/dev/demo3/mycode.c.rNEWrev
```

```
svn resolve mycode.c
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion
svn add

Place files and directories that are not under version-control yet under control of Subversion. The svn add command works in your local working copy and should at some point be followed by a commit.


```
/home/andre/dev/demo3/readme.txt   http://.../svn/demo/trunk/readme.txt
/home/andre/dev/demo3/mycode.c     http://.../svn/demo/trunk/mycode.c
/home/andre/dev/demo3/manual.doc
```

```
svn add manual.doc
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn copy

Copy files from one directory to another. This command is also used to create branches or tags. It can operate on your local working copy or directly on the repository itself.


```
/home/andre/dev/demo3/readme.txt          http://.../svn/demo/trunk/readme.txt
/home/andre/dev/demo3/mycode.c            http://.../svn/demo/trunk/mycode.c
/home/andre/dev/demo3/manual.doc          http://.../svn/demo/tags/1.0/readme.txt
/home/andre/dev/demo3/test/readme.txt     http://.../svn/demo/tags/1.0/mycode.c
```

```
svn copy readme.txt <subdirectory>
svn copy http://.../svn/demo/trunk http://.../svn/demo/tags/1.0
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn delete / svn move

svn delete removes files or directories from your local working copy. svn move moves files or directories to another location in your working copy. Both commands can also work on the repository directly.

Andre's Linux Workstation

Before

```
/home/andre/dev/demo3/readme.txt
/home/andre/dev/demo3/mycode.c
/home/andre/dev/demo3/manual.doc
/home/andre/dev/demo3/test/readme.txt
```

After

```
/home/andre/dev/demo3/mycode.c
/home/andre/dev/demo3/manual.doc
/home/andre/dev/demo3/test/readme.txt
```

```
svn delete test/readme.txt
svn move readme.txt test
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn blame

The blame command shows you exactly who changed what at which revision in a (text) file. This command cannot operate on binary files.

Andre's Linux Workstation

```
svn blame mycode.c

     2  akerstens #include <stdio.h>
     2  akerstens void main()
     2  akerstens {
     3  akerstens      printf("Hello World! How are you?\n");
     2  akerstens }
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn revert

Undo any changes made to your local working copy since the last update and before a commit.

Andre's Linux Workstation

```
svn revert mycode.c

Reverted 'mycode.c'
```

Subversion

Day to Day Tasks

- Importing data into the repository – svn import
- Exporting data from the repository – svn export
- Checking out a working copy from the repository – svn checkout
- Getting status information – svn status
- Sending your changes to the repository – svn commit
- Updating a working copy with changes of other people – svn update
- Resolving conflicts – svn resolve
- Adding new files and directories – svn add
- Copying files and directories – svn copy
- Deleting and moving – svn delete, svn move
- Finding out who made which changes – svn blame
- Turning back a local change – svn revert
- Finding out the history of a file – svn log

Subversion

svn log

Shows the revision history of your working copy or a specified file or a directory. This can be very useful to find out why a certain change to a file has been made, but it is important that meaningful comments are provided during commits.

Andre's Linux Workstation

```
svn log mycode.c
------------------------------------------------------------------------
r3 | akerstens | 2005-07-14 23:04:07 +0000 (Thu, 14 Jul 2005) | 1 line

Changed the hello string in mycode
------------------------------------------------------------------------
r2 | akerstens | 2005-07-14 22:46:49 +0000 (Thu, 14 Jul 2005) | 1 line

Lets load up some files into the trunk
------------------------------------------------------------------------
```

Subversion
Tips and Tricks

- Make sure that what you commit to the trunk in the repository is tested and works at all times. This way you are not forcing non-working code onto other people (who have to find out why it is not working).
- In case you need to test out something that potentially might fail and needs a rollback, create a branch (copy trunk to the branch directory) and use this for your experiments.
- If you want to release a stable revision of your code, tag it: which means copying it to the tag directory and giving it a release nr.
- Before you starting hacking code after coming into your office in the morning, update your working copy. If somebody else made changes overnight, you make sure you work with the latest version.

Subversion
More Information

- The Subversion Home Page - http://subversion.tigris.org/
- The Subversion Book - http://svnbook.red-bean.com/
- TortoiseSVN Home Page - http://tortoisesvn.tigris.org/
- Subclipse Home Page - http://subclipse.tigris.org/
- SCPlugin Home Page - http://scplugin.tigris.org/
- College of Eng Subversion - http://repository.eng.utep.edu/