Subversion

Pierre-Yves Dupont

Introduction

What is subversion

- SVN (2000). Last version 1.7.6 (August, 15 2012)

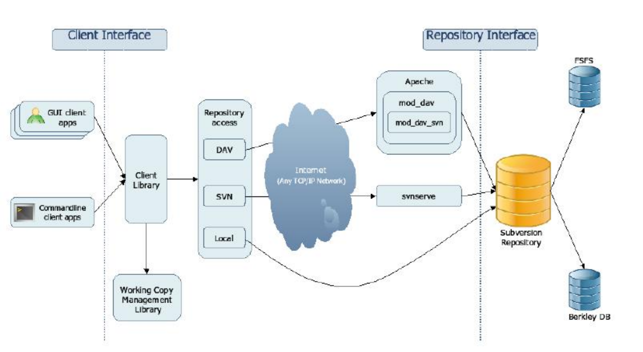
What is subversion

- SVN (2000). Last version 1.7.6 (August, 15 2012)
- revision control system (version control)

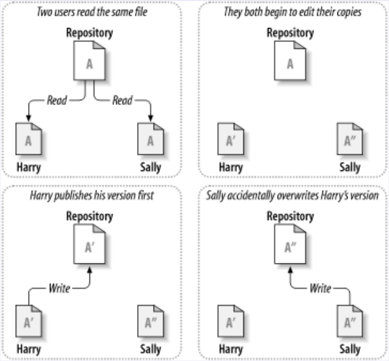
What is subversion

- SVN (2000). Last version 1.7.6 (August, 15 2012)
- revision control system (version control)
- CVS 1986, Hg (Mercurial) 2005, Git 2005

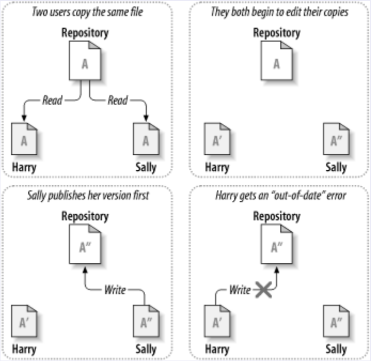
SVN System


Client Interface
Repository Interface
GUI client apps
Commandline client apps
Client Library
Working Copy Management Library
Repository access
DAV
SVN
Local
Internet (Any TCP/IP Network)
Apache
mod_dav
mod_dav_svn
svnserve
Subversion Repository
FSFS
Berkley DB

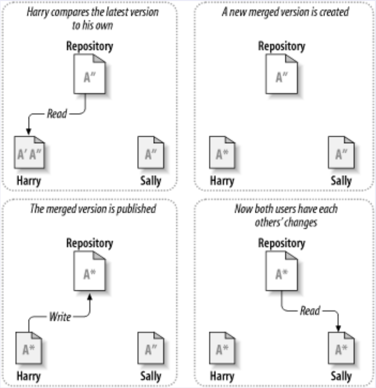
Problem


Two users read the same file
Repository
A
Read
Read
A
Harry
A
Sally
They both begin to edit their copies
Repository
A
A′
Harry
A″
Sally
Harry publishes his version first
Repository
A′
Write
A′
Harry
A″
Sally
Sally accidentally overwrites Harry's version
Repository
A″
Write
A′
Harry
A″
Sally

SVN solution

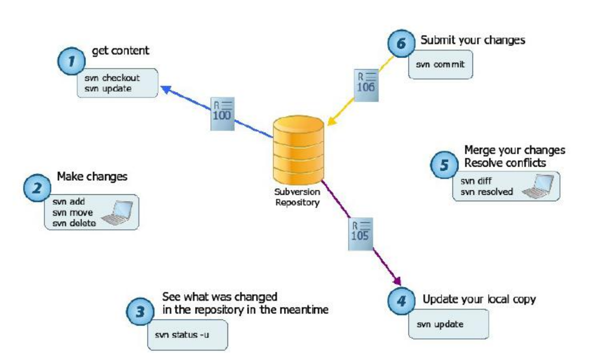

Two users copy the same file
Repository
A
Read
Read
A
A
Harry
Sally
They both begin to edit their copies
Repository
A
A′
A″
Harry
Sally
Sally publishes her version first
Repository
A″
Write
A′
A″
Harry
Sally
Harry gets an "out-of-date" error
Repository
A″
Write
A′
A″
Harry
Sally

SVN solution


Harry compares the latest version to his own
Repository
A″
Read
A′ A″
Harry
A″
Sally
A new merged version is created
Repository
A″
A*
Harry
A″
Sally
The merged version is published
Repository
A*
Write
A*
Harry
A″
Sally
Now both users have each others' changes
Repository
A*
Read
A*
Harry
A*
Sally

SVN commands

SVN workflow



Proxy configuration

Edit the file:

```
~/.subversion/servers
```

Find the “[global]” section and uncomment the following lines:

```
[global]
http-proxy-host = tur-cache1.massey.ac.nz
http-proxy-port = 8080
http-proxy-username = <user name>
http-proxy-password = <password>
```

Save the correct informations

Getting the data: checkout

```
%> svn checkout --username "username" \
 --password "pwd" http://mysvn.com/repository
```

No need to put the password in the command line
Same as:

```
%> svn co --username "username" http://mysvn.com/repository
```

checkout

Pull an SVN tree from the server. Should be done only once.

Updating the data: update

The command must be launched in the directory to update. Or give the path of the directory

```
%> svn update
```

Same as:

```
%> svn up
```

The informations about the repository, the user, etc. . . are in the .svn directory.

update

Synchronize your local copy with the server: add the new files, delete the removed files, merge the modified files.

Make changes on svn tree: add, move, . . .

```
%> svn add FILE_OR_DIR
%> svn move SRC DEST
%> svn mkdir DIR
```

Delete files or directories

```
%> svn delete FILE_OR_DIR
%> svn remove FILE_OR_DIR
%> svn del FILE_OR_DIR
%> svn rm FILE_OR_DIR
```

Same command

add move delete mkdir

Use the svn commands to act on files and directories in your local copy

Sending the changes to the repository: commit

```
%> svn commit files_or_dirs_to_commit -m 'a comment please!'
```

Same as:

```
%> svn ci files_or_dirs_to_commit -m 'a comment please!'
```

commit

Commit your local copy with all the changes you made on the repository. Don't forget to update first!

get informations: status

```
%> svn status -uv FILE_OR_DIR
```

Same as:

```
%> svn st -uv FILE_OR_DIR
```

’u’ option is used to compare the files with the repository, ’v’ is used to display all files and directories even if they are up to date.

status

Obtaining informations about the status of files or directories. The status of each file is given by a letter: added (A), deleted (D), conflicted (C), Ignored (I), modified (M), unversioned (?), missing (!), up to date ( )

Examples

svn co

```
%>svn co file:///var/svn/projet1
A    projet1/foo
Checked out revision 11.
```

svn status

```
%>svn status -uv
M               11        8 pydupont     foo
                11       11 pydupont     .
Status against revision:     11
```

svn log -v

```
r7 | pydupont | 2012-09-13 15:02:05 +1200 (Thu, 13 Sep 2012) | 1 line
Changed paths:
   D /bar

delete one file
------------------------------------------------------------------------
r6 | pydupont | 2012-09-13 15:01:15 +1200 (Thu, 13 Sep 2012) | 1 line
Changed paths:
   A /baz

add one file
------------------------------------------------------------------------
r5 | pydupont | 2012-09-13 14:53:15 +1200 (Thu, 13 Sep 2012) | 1 line
Changed paths:
   M /foo

conflict resolved
------------------------------------------------------------------------
```

svn add, svn up, svn commit

```
%>touch bar
%> svn add bar
A         bar
%> svn up
At revision 11.
%> svn ci -m "add bar"
Adding         bar
Transmitting file data .
Committed revision 12.
```

The conflicts

What is a conflict

A conflict occurs when there is two different modifications (by two users) on a same line in the same file

```
%>svn update
Conflict discovered in 'foo'.
Select: (p) postpone, (df) diff-full, (e) edit,
        (mc) mine-conflict, (tc) theirs-conflict,
        (s) show all options: p
C    foo
Updated to revision 4.
Summary of conflicts:
  Text conflicts: 1
```

What is a conflict

```
%>svn status -uv
C                4        4 pydupont     foo
?                                        foo.r3
?                                        foo.r4
?                                        foo.mine
                 4        4 pydupont     .
Status against revision:      4
```

foo

```
%>more foo
<<<<<<< .mine
First line - conflict
=======
Other line
>>>>>>> .r4
Second line
```

Solution

Conpare the files

foo

```
%>more foo
<<<<<<< .mine
First line - conflict
=======
Other line
>>>>>>> .r4
Second line
```

foo.r4

```
%>more foo.r4
Other line
Second line
```

foo.mine

```
%>more foo.mine
First line - conflict
Second line
```

foo

```
%>more foo
<<<<<<< .mine
First line - conflict
=======
Other line
>>>>>>> .r4
Second line
```

foo.r4

```
%>more foo.r4
Other line
Second line
```

foo.mine

```
%>more foo.mine
First line - conflict
Second line
```

Solution

Conpare the files

Comparing files

Tools to compare the files

- command lines: svn diff, diff . . .

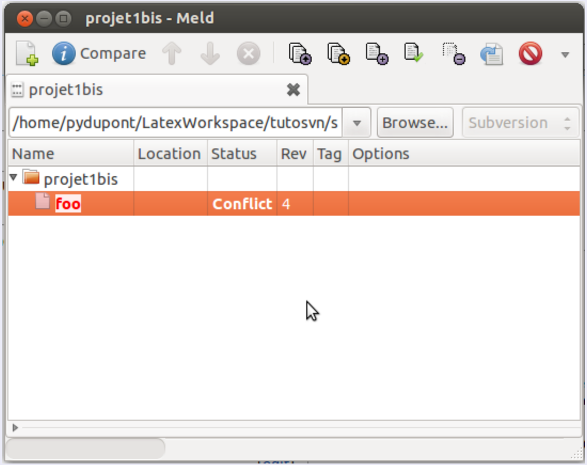
Tools to compare the files

- command lines: svn diff, diff . . .
- GUI: xxdiff, gvim -d, meld, . . .

Tools to compare the files

- command lines: svn diff, diff . . .
- GUI: xxdiff, gvim -d, meld, . . .
- In Eclipse: Subclipse

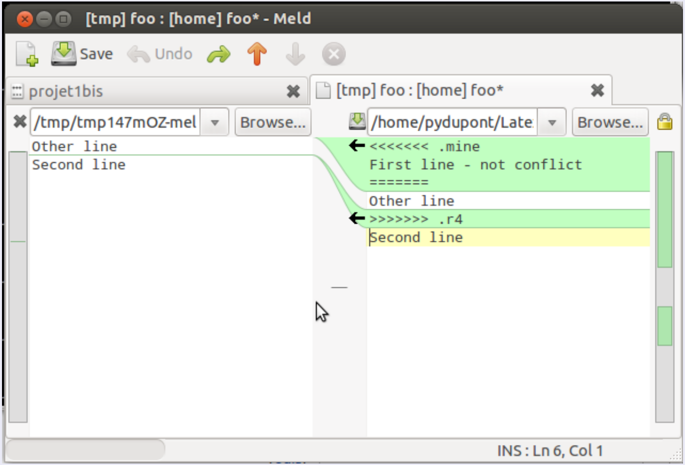
Meld: display the conflicted files



/home/pydupont/LatexWorkspace/tutosvn/s

Name	Location	Status	Rev	Tag	Options
projet1bis					
foo		Conflict	4		

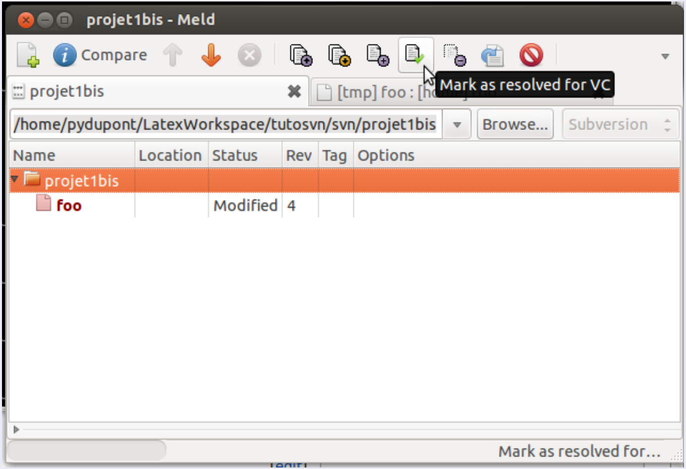
Meld: edit the conflited files



```
Other line
Second line
```

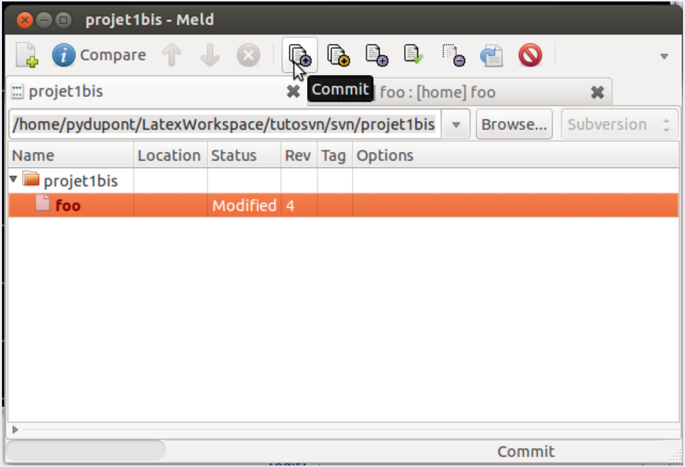
```
<<<<<<< .mine
First line - not conflict
=======
Other line
>>>>>>> .r4
Second line
```

Meld: resolve the conflicts



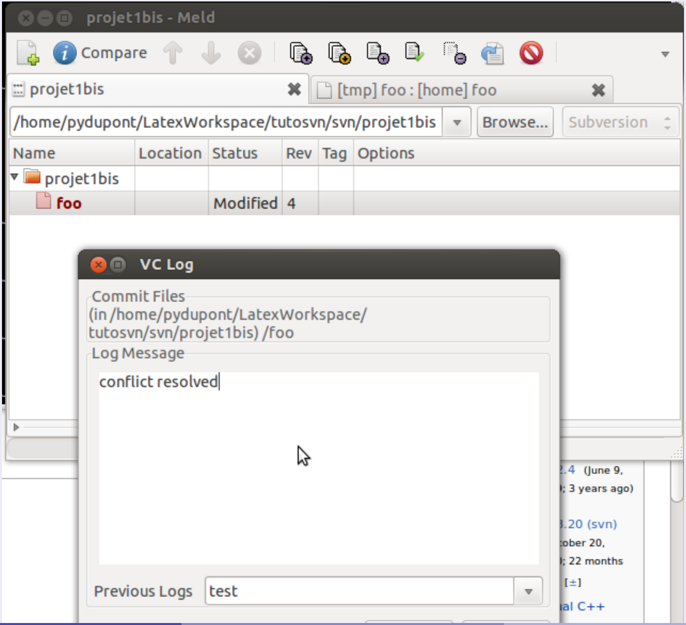
Name	Location	Status	Rev	Tag	Options
projet1bis					
foo		Modified	4		

Meld: commit



Name	Location	Status	Rev	Tag	Options
projet1bis					
foo		Modified	4		

Meld: add a log message to the commit



/home/pydupont/LatexWorkspace/tutosvn/svn/projet1bis

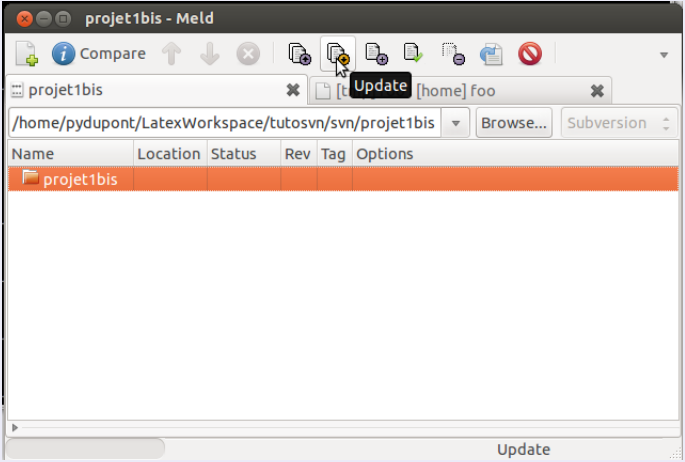
Name	Location	Status	Rev	Tag	Options
projet1bis					
foo		Modified	4		

Commit Files
(in /home/pydupont/LatexWorkspace/tutosvn/svn/projet1bis) /foo

Log Message: conflict resolved

Previous Logs: test

Meld: update


projet1bis - Meld
Compare
Update
projet1bis
[t
[home] foo
/home/pydupont/LatexWorkspace/tutosvn/svn/projet1bis
Browse...
Subversion
Name
Location
Status
Rev
Tag
Options
projet1bis
Update

Setup your own svn repository

Make a folder for the new repository

```
%>sudo mkdir /var/svn
```

Start svn server with your system

```
%>sudo gedit /etc/init.d/svnserve
```

```
 1 #!/bin/sh
 2 do_start () {svnserve -d -r /var/svn --pid-file /var/run/svnserve.pid}
 3 do_stop () {start-stop-daemon --stop --quiet --pidfile /var/run/svnserve.pid}
 4 case "$1" in
 5 start)
 6    do_start
 7   ;;
 8   stop)
 9   do_stop
10   exit $?
11   ;;
12   restart)
13   do_stop
14 sleep 1s
15   do_start
16   ;;
17 reload|force-reload)
18 echo "Error: argument '$1' not supported" >&2
19 exit 3
20 ;;
21 *)
22 echo "Usage: $0 start|stop|restart" >&2
23 exit 3
24 ;;
25 esac
```

Change line 2 to setup the correct location

Start svn server with your system

```
%>sudo chmod +x /etc/init.d/svnserve
%>sudo update-rc.d svnserve defaults
%>sudo /etc/init.d/svnserve start
```

Create a repository

```
%>sudo svnadmin create /var/svn/my_repository_name
```

Setup the repository

Uncomment the following lines in file

`/repository/path/conf/svnserve.conf`

```
anon-access = none
auth-access = write
password-db = passwd
```

Setup the access to repository

Edit the file

`/repository/path/conf/passwd`

```
[users]
user1_login = user1_password
user2_login = user2_password
```

Documentation

web sites

Ubuntu:

1. RabbitVCS `http://wiki.rabbitvcs.org/wiki/install/ubuntu`
2. SVN book `http://svnbook.red-bean.com/en/1.7/svn-book.html`
3. Wikipedia `http://en.wikipedia.org/wiki/Apache_Subversion`
4. Ubuntu doc `https://help.ubuntu.com/community/Subversion`

MAC OSX:

- Installation `http://www.rubyrobot.org/tutorial/subversion-with-mac-os-x`
- GUI `http://www.lachoseinteractive.net/en/community/subversion/svnx/features/`