SVN Totorial

J. Farquhar

MMM Thursday Talk, 2008

bg-logo

Why SVN?

Version Control

- Is a database which holds a complete history of what has happened to a file and the reasons why it changed.
- A system for managing changes to files to ensure a single consistent view of changes comming from multiple locations.

How does it work?

- keeps a copy of all registered files
- keeps a history for these files
- keeps info on who changed the file
- keeps a log, provided by the modifier, of what the change was supposed to do
- allows multiple local copies of a project
- automatically merges changes from different locations/people/branches to generate a consistent project

bg-logo

Why SVN?

The big idea

Allows multiple people with multiple local-copies of a project to edit files in the same project and keep all these copies consitent.

Does this by;

- checking if someone has edited the same file
- forcing you to resolve the different changes
- before allowing your changes to enter the repository

bg-logo

Nomclameture

repository	the actual location where the “real” copy of the files lives
checkout	the process of making a local copy of the repository
update	the process of ensuring your local copy is up-to-date with the current repository state
commit	the process of merging your local changes into the repository

bg-logo

Typical project usage

1. checkout an initial copy of the repository
   U1: svn co svn+ssh://mmmxserver.nici.ru.nl/Volumes/Xserver_RAID/BCI_code/svn/BCI P1
   U2: svn up svn+ssh://mmmxserver.nici.ru.nl/Volumes/Xserver_RAID/BCI_code/svn/BCI P2
2. update local copies to get everyone elses changes
   U1: cd project; svn up
   U2: cd project; svn up
3. edit files on the local copy
   U1: edit P1/file1.m
   U2: edit P2/flie2.m
4. add new files to the local copy
   U1: svn add P1/file11.m
   U2: svn add P2/flie21.m
5. commit local changes to the repository (when tested and working)
   U1: svn ci P1/file1.m -m "fixed bug 1"
   U2: svn ci P2/file2.m -m "fixed bug 2"
6. goto 2 and repeat

bg-logo

Suggested workflow

9:00am	Get to work, ;-)
9:01am	SVN update (Finder->(ctrl-click)->more->Subversion->Update)
9:02am	Get coffee (while update runs)
9:10am	Make lots of changes
9:50am	Fix lots of bugs
10:20am	test the fix
11:20am	SVN commit the bug fix with a comment to say what you've done

- you can do this direct from Matlab File->source control->commit OR
- Finder->(ctrl-click)->more->Subversion->commit

don't forget to add any new files you've created

......

5:00pm	SVN commit before you go home, just to check you haven't missed something!

bg-logo

Mac-OS SVN setup

1. Install SCPlugin (Applications/Utilities/version_control/SCPlugin-0.7.1.dmg)
2. Install Matlab plugin[1] – add this to Matlab path (BCI code/toolboxes/utilities/svn_integeration/customverctrl.m)
3. Setup password-less ssh – run from terminal (BCI code/toolboxes/utilities/svn_integeration/setup_server_pwless_ssh.sh)
4. Checkout the BCI code repository:
   - svn co "svn+ssh://mmmxserver.nici.ru.nl/Volumes/Xserver_RAID/BCI_code/svn/BCI"
     OR
   - Finder->(ctrl-click)->more->Subversion->Checkout
     From:
     svn+ssh://mmmxserver.nici.ru.nl/Volumes/Xserver RAID/BCI code/svn/BCI

[1] Optional – only if you want to commit direct from matlab

bg-logo

SVN got-yah's

- SVN only knows about registered files – you need to add any new files you make
- the same applies when you move files – to SVN this looks like a delete old+add new
- file deletion is treated as an (extreem) edit – you need to commit deleted files to really delete them