Subversion Centralised version control

CONCEPTS

Revisions are repository-wide numbers: r42 is the whole tree at one point, not one file.

Working copy your files
↓ `svn commit`
Repository central server, revisions
↓ `svn update`
Working copy everyone else's

Branches and tags are ordinary directories made with svn copy; the layout trunk/ branches/ tags/ is only convention.

WORKING COPY

`svn checkout <url> [<dir>]`
Check out a working copy

`svn update`
Bring working copy up to date

`svn update -r <rev>`
Update to a specific revision

`svn info`
Show URL, revision, repo root

`svn cleanup`
Release locks after a crash

`svn switch <url>`
Point working copy at another branch

`svn export <url> <dir>`
Clean tree without .svn

CHANGES

`svn status`
List local changes

`svn status -u`
Also show server-side changes

`svn add <path>`
Schedule files for addition

`svn delete <path>`
Schedule removal

`svn move <src> <dst>`
Rename, keeping history

`svn copy <src> <dst>`
Copy, keeping history

CHANGES (cont.)

`svn mkdir <dir>`
Create a versioned directory

`svn revert -R <path>`
Undo local changes recursively

`svn commit -m "<msg>"`
Commit local changes

INSPECT

`svn diff`
Local changes against base

`svn diff -c <rev>`
Changes made in one revision

`svn diff -r <a>:<b> <path>`
Compare two revisions

`svn log -l <n>`
Last n log entries

`svn log -v`
Include changed paths

`svn log -r <a>:<b>`
Log for a revision range

`svn blame <file>`
Who last changed each line

BRANCH AND MERGE

`svn copy <trunk-url> <branch-url> -m "<msg>"`
Create a branch

`svn switch <branch-url>`
Work on the branch

`svn merge <branch-url>`
Merge branch changes into here

CONFLICTS

`svn resolve --accept working <path>`
Keep your edited file

A conflict leaves .mine and .rNN files next to the file; resolve removes them.

PROPERTIES AND LOCKS

`svn propset svn:ignore "<pattern>" <dir>`
Ignore files in a directory