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.

Unlike git, commit goes straight to the server and needs network access.

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 relocate *<new-url>*
Repository moved to a new URL

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

CHANGES (cont.)

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

svn mkdir *<dir>*
Create a versioned directory

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

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

svn commit -F *<file>*
Read log message from file

svn changelist *<name> <path>*
Group files for a commit

svn commit --changelist *<name>* **-m "***<msg>***"**
Commit only a changelist

svn add --force *<dir>*
Add all unversioned files in a tree

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 diff --git
Output in git diff format

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 log --stop-on-copy
Stop at branch creation

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

svn cat -r *<rev> <file>*
Print a file at a revision

svn list *<url>*
List a repository directory

BRANCH AND MERGE

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

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

BRANCH AND MERGE (cont.)

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

svn merge -c *<rev> <url>*
Merge a single revision

svn merge --dry-run *<url>*
Preview a merge

svn mergeinfo --show-revs eligible *<url>*
Revisions not yet merged

svn copy *<trunk-url> <tag-url>* **-m "***<msg>***"**
Create a tag

CONFLICTS

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

svn resolve --accept theirs-full *<path>*
Take the incoming file

svn resolve --accept mine-full *<path>*
Take your version of the 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

RECIPES

Daily loop

```
svn update
svn status
svn diff
svn commit -m "Fix typo in
README"
```

Feature branch and merge back

```
svn copy ^/trunk ^/branches/feat
-m "Branch feat"
svn switch ^/branches/feat
# ...work, commit...
svn switch ^/trunk
svn merge ^/branches/feat
svn commit -m "Merge feat"
```

Undo a committed revision

```
svn update
svn merge -c -<rev> .
svn commit -m "Revert r<rev>"
```