Subversion - A Summary Cheat Sheet - Learn svn in 10 minutes

From http://jwamicha.wordpress.com/2008/05/29/subversion-a-summary-cheat-sheet-learn-svn-in-10-minutes/
This post is a summary of the subversion book, only that the summary takes you straight in. Please find the subversion book here: http://svnbook.red-bean.com/
If you have not done so already, begin by installing Subversion on your system. For Fedora/CentOS/Redhat users, this is explained in another post here: http://jwamicha.wordpress.com/2008/04/25/quick-svn-trac-installation-on-centosfedora/

1 Subvserion commands summary

1.1 Checkout

Checkout the code and do an update in case of any changes made since your last update (We assume that you are using apache dav server to access your code and not svnserve):

```
    $svn checkout http://192.168.0.54/svn/repos/server\_code server\_code
```

If your repository requires authentication:

```
    $svn checkout —username my\_username http://192.168.0.54/svn/repos/server\_code
    server\_code
```

Update your working copy:

```
    $svn update
    (update from current)
    $svn update —r BASE server\_code
    (update foo from base revision)
    $svn update —r 1200 server\_code (update foo from revision number 1200)
```

1.2 Make changes

```
    $svn add eg svn add new\_directory
    (add a new directory foo)
    $svn delete
    $svn copy directory1 directory2
    (copy directory directory1 to directory2)
    $svn move directory2 renamed\_directory
    (rename?)
```

1.3 Examine your changes

Can be done even with no network access to the subversion repository

```
    $svn status
    (To get an overview of all your changes)
    eg
    A stuff/loot/bloo.h # file is scheduled for addition
    C stuff/loot/lump.c # file has textual conflicts from an update
    D stuff/fish.c # file is scheduled for deletion
    M bar.c # the content in bar.c has local modifications

    $svn diff
    (to show changes between current working directory and the same directory in the
    repository)
```

1.4 Possibly undo some changes

Can also be done even with no network access to the subversion repository

```
    $svn revert
    After running svn revert as a way to resolve local conflict with the repository copy,
    Run:

    $svn resolve
    To inform svn that the conflict has been resolved. You will now be able to successfully
    run svn update in case of previous conflicts.
```

1.5 Resolve Conflicts

Merge Others' Changes

```
    $svn update
    $svn resolved
```

1.6 Commit your changes

```
    $svn commit
    eg
    $svn commit -m "Removed out of mem errors."
    or
    $svn commit -F comment.txt
    or
    $svn commit -file comment.txt
```

1.7 Logs

```
    $svn log (use current working directory as the default target)
    $svn log server\_code
    (current working directory/file is server\_code)
    $svn log -r 5:19
    (shows logs 5 through 19 in chronological order of working directory)
    $svn log -r 19:5
    (shows logs 5 through 19 in reverse order of working directory)
    $svn log -r 8
    (shows log for revision 8 of working directory)
    $svn log -r 8 -v
    (shows verbose? log for revision 8 of working directory)
```

1.8 Diffs (Changes)

```
    $svn diff
    $svn diff −r 3 rules.txt
    (or svn diff −revision 3 rules.txt)
    $svn diff −r 2:3 rules.txt
    (revisions 2 and 3 are directly compared)
    $svn diff −c 3 rules.txt
    (compare changes between current revision and revision 2)
```

1.9 Browse a file directly

```
    $svn cat −r 2 rules.txt
    $svn cat −r 2 rules.txt > rules.txt.v2 (send cat output directly to a file)
```

1.10 Browse a folder directly

```
    svn list http://svn.collab.net/repos/svn
    svn list −v http://svn.collab.net/repos/svn
```

1.11 Fetching older repository snapshots

```
    $svn checkout −r 1729
    (Checks out a new working copy at r1729)
    $svn update −r 1729
    (Updates an existing working copy to r1729)
```

1.12 Export

If you're building a release and wish to bundle up your files from Subversion but don't want those pesky .svn directories in the way, then you can use svn export to create a local copy of all or part of your repository sans .svn directories. As with svn update and svn checkout, you can also pass the - -revision switch to svn export:

```
    $svn export http://svn.example.com/svn/repos1
    (Exports latest revision)
    $svn export http://svn.example.com/svn/repos1 −r 1729
    (Exports revision r1729)
```

1.13 Cleanup

Cleanup if a Subversion operation is interrupted (if the process is killed, or if the machine crashes, for example), the log files remain on disk. By re-executing the log files, Subversion can complete the previously started operation, and your working copy can get itself back into a consistent state.

```
    $svn cleanup
```

2 Using the revision specifiers

- HEAD: The latest (or ”youngest”) revision in the repository.
- BASE: The revision number of an item in a working copy. If the item has been locally modified, the ”BASE version” refers to the way the item appears without those local modifications.
- COMMITTED: The most recent revision prior to, or equal to, BASE, in which an item changed.
- PREV: The revision immediately before the last revision in which an item changed. Technically, this boils down to COMMITTED-1.

```
    $svn diff -r PREV:COMMITTED main.c
    (shows the last change committed to main.c)

    $svn log -r HEAD
    (shows log message for the latest repository commit)

    $svn diff -r HEAD
    (compares your working copy with all of its local changes to the latest version of that
    tree in the repository)

    svn diff -r BASE:HEAD main.c
    (compares the unmodified version of foo.c with the latest version of foo.c in the
    repository)

    $svn log -r BASE:HEAD
    (shows all commit logs for the current versioned directory since you last updated

    $svn update -r PREV main.c
    (rewinds the last change on foo.c, decreasing foo.c's working revision)

    $svn diff -r BASE:14 main.c
    (compares the unmodified version of foo.c with the way foo.c looked in revision 14)
```

2.1 Checkout based on revisions

```
    $svn checkout -r {'2006-02-17'}
    $svn checkout -r {'15:30'}
    $svn checkout -r {'15:30:00.200000'}
    $svn checkout -r {'2006-02-17 15:30'}
```

2.2 Logs based on revisions

```
    $svn log -r {2006-11-28}
    $svn log -r {2006-11-20}:{2006-11-29}
```

3 Properties of files

```
    $svn propset copyright '(c) 2006 Red-Bean Software' calc/button.c
    property 'copyright' set on 'calc/button.c'

    $svn propset license -F /path/to/LICENSE calc/button.c
    property 'license' set on 'calc/button.c'

```

```
$svn propedit copyright calc/button.c
No changes to property 'copyright' on 'calc/button.c'

$svn propset copyright '(c) 2006 Red-Bean Software' calc/*
property 'copyright' set on 'calc/Makefile'
property 'copyright' set on 'calc/button.c'
property 'copyright' set on 'calc/integer.c'

$svn proplist calc/button.c
Properties on 'calc/button.c':
copyright
license

$svn propget copyright calc/button.c
(c) 2006 Red-Bean Software

$svn proplist -v calc/button.c

$svn propset license " calc/button.c
$svn propdel license calc/button.c

And specify the revision whose property you wish to modify

$svn propset copyright '(c) 2006 Red-Bean Software' calc/button.c -r11 -revprop
```

4 Locking files

```
$svn lock banana.jpg -m "Editing file for tomorrow's release."
'banana.jpg' locked by user 'harry'.

$svn status
K banana.jpg

$svn info banana.jpg
Path: banana.jpg
Name: banana.jpg
URL: http://svn.example.com/repos/project/banana.jpg
Repository UUID: edb2f264-5ef2-0310-a47a-87b0ce17a8ec
Revision: 2198
Node Kind: file
Schedule: normal
Last Changed Author: frank
Last Changed Rev: 1950
Last Changed Date: 2006-03-15 12:43:04 -0600 (Wed, 15 Mar 2006)
Text Last Updated: 2006-06-08 19:23:07 -0500 (Thu, 08 Jun 2006)
Properties Last Updated: 2006-06-08 19:23:07 -0500 (Thu, 08 Jun 2006)
Checksum: 3b110d3b10638f5d1f4fe0f436a5a2a5
Lock Token: opaquelocktoken:0c0f600b-88f9-0310-9e48-355b44d4a58e
Lock Owner: harry
Lock Created: 2006-06-14 17:20:31 -0500 (Wed, 14 Jun 2006)
Lock Comment (1 line):
Editing file for tomorrow's release.

$svnadmin lslocks /usr/local/svn/repos
$svnadmin rmlocks /usr/local/svn/repos /project/raisin.jpg
Force out someone else's lock:

$svn unlock -force http://svn.example.com/repos/project/raisin.jpg
Force a lock over someone else's
$ svn lock -force raisin.jpg
```

5 Creating branches

```
    $svn checkout http://svn.example.com/repos/calc bigwc
    A bigwc/trunk/
    A bigwc/trunk/Makefile
    A bigwc/trunk/integer.c
    A bigwc/trunk/button.c
    A bigwc/branches/
    Checked out revision 340.
```

Now create the branch;

```
    $cd bigwc
    $svn copy trunk branches/my-calc-branch
    $svn status
    A + branches/my-calc-branch

    $svn commit -m "Creating a private branch of /calc/trunk."
    Adding branches/my-calc-branch
    Committed revision 341.
```

You can do all the above in one step (Recommended way):

```
    $svn copy http://svn.example.com/repos/calc/trunk \
    http://svn.example.com/repos/calc/branches/my-calc-branch \
    -m "Creating a private branch of /calc/trunk."
    Committed revision 341.
```

Merging branch to main trunk (Assuming you are in the working branch directory)

```
    $svn merge -c 344 http://svn.example.com/repos/calc/trunk (merge change revision number
    344 on your working directory branch)
    U integer.c

    $svn status
    M integer.c
```

Merging while specifying the destination and target:

```
    $svn merge -c 344 http://svn.example.com/repos/calc/trunk my-calc-branch
    U my-calc-branch/integer.c

    $svn merge http://svn.example.com/repos/branch1@150 \
    http://svn.example.com/repos/branch2@212 \
    my-working-copy

    $svn merge -r 100:200 http://svn.example.com/repos/trunk my-working-copy

    $svn merge -r 100:200 http://svn.example.com/repos/trunk
```

Previewing merges:

```
    $svn merge - -dry-run -c 344 http://svn.example.com/repos/calc/trunk
    U integer.c
    (- -dry-run is a double dash without spaces. Word press munges the double dash into one
    when put together.)

    $svn status
    (nothing printed, working copy is still unchanged)
```

Merging branch changes into trunk:

```
    $cd calc/trunk
    $svn update
```

```
At revision 405.

$svn merge -r 341:405 http://svn.example.com/repos/calc/branches/my-calc-branch
U integer.c
U button.c
U Makefile

$svn status
M integer.c
M button.c
M Makefile
```

Examine the diffs, compile, test, etc...

```
$svn commit -m "Merged my-calc-branch changes r341:405 into the trunk."
Sending integer.c
Sending button.c
Sending Makefile
Transmitting file data ...
Committed revision 406
```

Undo a merge:

```
$svn merge -c -303 http://svn.example.com/repos/calc/trunk
or
$svn merge -revision 303:302 http://svn.example.com/repos/calc/trunk
U integer.c

$svn status
M integer.c

$svn diff
(Verify that the change is removed)

$svn commit -m "Undoing change committed in r303."
Sending integer.c
Transmitting file data .
Committed revision 350.
```

Merging from branch to trunk:

```
$cd trunk-working-copy

$svn update
At revision 1910.

$svn merge http://svn.example.com/repos/calc/trunk@1910 \
  http://svn.example.com/repos/calc/branches/mybranch@1910

U real.c
U integer.c
A newdirectory
A newdirectory/newfile
```

Resurrecting deleted items:

```
$svn copy -r 807 \
http://svn.example.com/repos/calc/trunk/real.c ./real.c

$ svn status
A + real.c

$svn commit -m "Resurrected real.c from revision 807, /calc/trunk/real.c."
Adding real.c
Transmitting file data .
Committed revision 1390.
```

Traversing branches:

```
    $cd calc

    $svn info | grep URL
    URL: http://svn.example.com/repos/calc/trunk

    $svn switch http://svn.example.com/repos/calc/branches/my-calc-branch
    U integer.c
    U button.c
    U Makefile
    Updated to revision 341.

    $svn info | grep URL
    URL: http://svn.example.com/repos/calc/branches/my-calc-branch
```

Making releases using tags (snapshot of a directory at a given instant in time)

```
    $svn copy http://svn.example.com/repos/calc/trunk \
    http://svn.example.com/repos/calc/tags/release-1.0 \
    -m "Tagging the 1.0 release of the 'calc' project."

    Committed revision 351.
```

Remove your branch after merge:

```
    $svn delete http://svn.example.com/repos/calc/branches/my-calc-branch \
    -m "Removing obsolete branch of calc project."

    Committed revision 375.
```

6 More commands

Commit a log message correction:

```
    $echo "Here is the new, correct log message" > newlog.txt
    $svnadmin setlog myrepos newlog.txt -r 388
```

Migrate repository: Create the dump files first:

```
    $svnadmin dump myrepos -r 23 > rev-23.dumpfile
    $svnadmin dump myrepos -r 100:200 > revs-100-200.dumpfile
```

Load the dump files into the new repository:

```
    $svnadmin dump myrepos -r 0:1000 > dumpfile1
    $svnadmin dump myrepos -r 1001:2000 -incremental > dumpfile2
    $svnadmin dump myrepos -r 2001:3000 -incremental > dumpfile3
```