SUBVERSION QUICK REFERENCE CARD

TYPICAL WORKFLOW

- **Checkout / Update working copy**
  - › *svn co svn://svn.ssg.uab.edu/qtlbim/pkg/trunk*
  - › *(or) svn update*
- **Make changes**
- **Review what you did**
  - › *svn status (summary output)*
  - › *svn diff (detailed output)*
- **Review what others did**
  - › *svn status –u (summary output)*
  - › *svn diff –r BASE:HEAD (detailed output)*
- **Update, for latest changes by others**
  - › *svn update*
- **Build package**
- **Run tests**
- **Commit with descriptive log message**
  - › *svn commit –m "Log message"*

GENERAL INFO

- **Getting Help (help)**
  - › *svn help <command name>*

  This gives you information on the command syntax and the various switches that go with it. By far, it's the most useful and handy svn reference.
- **Repository URLs:**
  - › *<scheme>://<repo location>/<subdirectory>*
  - › *svn://svn.ssg.uab.edu/qtlbim/pkg/trunk/*
- **Referring to revisions**

  Precede references to a revision(s) with a *-r* or a - - r*evision* switch. Alternatively, revision numbers can be suffixed to the url with an '@' symbol. For e.g.:
  - › *svn://svn.ssg.uab.edu/qtlbim/pkg/trunk/ -r 851*
  - › *svn://svn.ssg.uab.edu/qtlbim/pkg/trunk/@851*

  by Revision number

  Simply use the revision number (e.g. 5). Remember the revision number applies to the whole project as such and *not* to a file

  by Keyword

  Head: Latest revision in the repository
  Base: Checked out revision of the working copy
  Committed: Last revision in which an item changed
  Prev: The revision just before '*Committed*'

  by Date
  - › *{ 2006-06-18 }* *{ 2006-11-25 15:30 }*

  by Range of revisions:
  - › *265:269* *PREV:BASE* *BASE:HEAD*

REPOSITORY TRANSACTIONS

- **Check out (co)**

  The latest revision
  - › *svn checkout svn://svn.ssg.uab.edu/qtlbim/pkg/trunk*

  A specific revision
  - › *svn checkout svn://svn.ssg.uab.edu/qtlbim/pkg/trunk –r REV*

  A Branch
  - › *svn checkout svn://svn.ssg.uab.edu/qtlbim/pkg/branches/branch_name*

  A Tagged snapshot
  - › *svn checkout svn://svn.ssg.uab.edu/qtlbim/pkg/tags/tag_name*

  Note that the mainline, branches and tags are differentiated only by the path that you supply. Also, an older revision of any of these can be specified.
- **Update (up)**

  Update to the latest revision
  - › *svn update*

  Update to a specific revision
  - › *svn update –r*

  Update symbols
  1. A – item Added
  2. D – item Deleted
  3. U – item Updated
  4. G – item merGed successfully
  5. C – item in Conflict
- **Commit (ci)**
  - › *svn commit*

  Please check the final output message to see if the commit was successful.

REVIEWING

- **Status (st)**

  Checking for local modifications
  - › *svn status*

  Checking for repository modifications
  - › *svn status –u*
- **Diff (di)**

  Your changes to working copy
  - › *svn diff*

Others' changes to repository

› *svn diff –r BASE:HEAD*

Changes between two revisions

› *svn diff –r REV1:REV2 url*

Changes between working copy and a revision

› *svn diff –r REV url*

Changes between two URLs
svn diff url1 url2

- **Log**

Prints commit log messages, author and revision dates. By default, prints logs till revision in which the item was created. Specify revisions / ranges for more specific logs. Refer to the '*General Info*' section.

All logs for current directory

› *svn log*

All logs for current item

› *svn log <item_name>*

All logs for specified url

› *svn log url*

All logs just for a branch

› *svn log --stop-on-copy branch_url*

- **List (ls)**

Browse the repository. Use it like the ls command in unix

› *svn ls –r svn://svn.ssg.uab.edu/qtlbim/trunk*

EDITING FILE CONTENT

Modify file content as you please, as you normally do. Subversion does not affect the way you edit code, compile, build etc.

EDITING DIRECTORY CONTENT

Just like with editing file content, these operations will *require a commit* before they are reflected in the repository. This is because only the 'safe' ways of doing things are listed here.

- **Add**

Bring new files / directories under the version control umbrella

› *svn add name(s)*

- **Delete (rm)**

Remove files from the latest revision

› *svn delete name(s)*

- **Copy (cp)**

Copies an item with its *previous* history.

› *svn copy SRC DEST*

- **Move (mv) / Rename (ren)**

Moves (renames) an item with its *previous* history. Note that both these actions mean the same thing to svn. Hence the commands are just different names for the same action.

› *svn move SRC TARGET*

› *svn rename SRC TARGET*

BRANCHING AND TAGGING

Subversion treats both branching and tagging identically. It does *NOT* know the difference between the two. This means that the same command is used to create both. The distinction between a branch and a tag is in how *YOU* treat it.

- **Branch**

› *svn copy branch_source branch_path/branch_name*

Uses the *svn copy* command to create a branch by just making a copy of the source revision. This *requires a commit* for the branch to become part of the repository.

› *svn checkout(or) svn update to the required revision*

› *svn copy ./pkg/trunk ./pkg/branches/branch_name*

› *svn commit*

- **Tag**

› *svn copy tag_source tag_path/tag_name*

Uses the *svn copy* command to make a copy of the items that you want to snapshot. *Requires a commit* to incorporate the tag in the repository.

› *svn checkout(or) svn update to the required revision*

› *svn copy ./pkg/trunk ./pkg/tags/tag_name*

› *svn commit*

MERGING CHANGES

- **Merge**

Merging a bug-fix onto the trunk

› *svn checkout / update copy of the trunk*

› *svn merge tag_path/pre_tag tag_path/post_tag*

› *svn commit*

- **Revert**

› *svn revert item_name*

Reverts your working copy to the 'Base' revision. It's like an undo for all your changes

- **Resolved**

› *svn resolved item_name*

Indicates to Subversion that you have resolved a conflict that it had flagged earlier. This removes locks that were put in place and will permit you to commit your changes that resolve the conflict.

REFERENCES

1. http://subversion.tigris.org
2. http://svnbook.red-bean.com/
3. Pragmatic Version Control with SVN: (2nd ed., Pragmatic Bookshelf)

Authors: Ramprasad Venkataraman, Tapan Mehta @ SSG, UAB