

Help Topic: The SVN info and log Commands

Spring 2021

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

The SVN `info`, and `log` Commands

The `svn info` and `svn log` commands provide useful information about a given repository. They are both executed on the command line somewhere within the directory of the locally checked out repository.

- The `svn info` command may be invoked offline with no network connection to the Subversion server.
- The `svn log` command requires a network connection to the server and perhaps a password if it is not an anonymous read-only repository.

More info can always be found by Googling "Subversion book" and reading the full PDF online free, or just typing `svn help info` anytime on the command line.

The `svn info` command

The basic usage of the `svn info` command is simple:

```
$ svn info          (invoked somewhere within previously checked out tree)
```

The `svn info` command provides a quick look at some key properties of your locally checked out repository. By "quick" I mean the information is stored locally and doesn't involve communication with the (most likely remote) server. Below is example output from my own locally checked out version of the moos-ivp tree:

```
$ cd moos-ivp
$ svn info
Path: .
Working Copy Root Path: /Users/mikerb/Research/moos-ivp
URL: svn+ssh://18.38.2.158/home/svn/repos/moos-ivp/trunk
Relative URL: ^/trunk
Repository Root: svn+ssh://18.38.2.158/home/svn/repos/moos-ivp
Repository UUID: 7b6de600-d9ab-45c9-a1ca-e8145d3c6daf
Revision: 6048
Node Kind: directory
Schedule: normal
Last Changed Author: mikerb
Last Changed Rev: 6046
Last Changed Date: 2013-10-18 14:53:54 -0400 (Fri, 18 Oct 2013)
```

You can read in the subversion documentation the full meaning, but the two pieces of information I find most useful are:

- The Revision: This information can help you determine quickly if your repository is up to date with a lab partner. If you're Revision numbers are the same, you're working on the same code. Otherwise, one of you may need to run the `svn update` command.
- The URL: The content of this line can be cut and pasted back into the command line for a fresh checkout. Perhaps to check out an older revision.

Note that if you invoke this command from within different subdirectories of your local repository, you will get "Last Changed" information local to that subtree. For example, the overall tree/repository may have had its last change yesterday and be at revision 99, but the a subtree may have had its last change two months ago and that change may have represented revision 22. So if you really want to see the "global" revision number and last-changed date, move to the top level of the repository and execute the `svn info` command. Furthermore, this information only represents your local checkout. If you want to know the latest revision on the svn server, do an `svn update` first.

The `svn log` command

The `svn log` command provides a log of changes made to the repository across the group of developers. It is incremental in date and revision number, and tells you who made the change and when. It should also tell you, in a short description, what changes or fixes were involved for a given log entry. The basic format is this:

```
$ svn log          (invoked somewhere within previously checked out tree)
```

If there's more than a handful of log entries in the history, this will quickly scroll by your screen. Since the most common entry of interest is the most recent entry, this will have scrolled off your screen first since log entries are reported in descending by revision number. You can use your terminal application to scroll up, but I tend to do one of the following instead:

```
$ svn -l 10 log    (Shows only the 10 most recent entries)
or
$ svn log | more   (Pipes it to the more utility)
```

Here is an example from my locally checked out copy of the moos-ivp tree.

```
$ cd moos-ivp/
$ svn -l 10 log
-----
r6052 | k_w | 2013-10-30 14:40:00 -0400 (Wed, 30 Oct 2013) | 1 line
updated ulv.sh to handle multiple uLogViewIPF alog file inputs
-----
r6051 | mikerb | 2013-10-28 16:00:44 -0400 (Mon, 28 Oct 2013) | 1 line
mod to OSX README
-----
r6050 | mikerb | 2013-10-28 15:17:53 -0400 (Mon, 28 Oct 2013) | 1 line
-----
r6049 | mikerb | 2013-10-28 11:52:27 -0400 (Mon, 28 Oct 2013) | 1 line
minor fixes after compiling on Mavericks
-----
r6048 | mikerb | 2013-10-18 16:43:34 -0400 (Fri, 18 Oct 2013) | 1 line
-----
r6047 | mikerb | 2013-10-18 15:43:50 -0400 (Fri, 18 Oct 2013) | 1 line
-----
r6046 | mikerb | 2013-10-18 14:53:54 -0400 (Fri, 18 Oct 2013) | 1 line
more tweaks to the build-ivp.sh script
-----
r6045 | mikerb | 2013-10-18 14:30:59 -0400 (Fri, 18 Oct 2013) | 1 line
modified the build-ivp.sh script to make it easier to invoke debug/release/none
options. Default is no longer debug, but rather none
-----
r6044 | mikerb | 2013-10-18 13:25:37 -0400 (Fri, 18 Oct 2013) | 1 line
-----
r6043 | k_w | 2013-10-18 13:24:14 -0400 (Fri, 18 Oct 2013) | 1 line
updated uFldCollisionDetect to base collisions on CPA since last node report
rather than current distance to contacts.
-----
```

Note that the revisions are perfectly in order without skipping a number. If this command were invoked within a subdirectory of the tree, you would see skips in the revision number.

The comments in the log entries are important, and are generated upon a *commit* to the tree. They are not mandatory, and even in our example above, some entries do not have comments. You have to use your own judgement in striking a balance between quick and convenient commits without comments against verbose commits which take more time. At some point you may need the comments of your log to help you revert changes or just remind yourself when something key

changed in the history.

Knowing exactly what files changed for a given revision/log entry

In the `svn log` example above, the entry for revision number 6049 looks like:

```
-----  
r6049 | mikerb | 2013-10-28 11:52:27 -0400 (Mon, 28 Oct 2013) | 1 line  
minor fixes after compiling on Mavericks  
-----
```

This shows the comment entered by the person committing the change, but *what really changed* in this commit? It would be good to at least know what files are involved. This can be done with the following:

```
$ svn log --verbose -r 6049  
-----  
r6049 | mikerb | 2013-10-28 11:52:27 -0400 (Mon, 28 Oct 2013) | 1 line  
Changed paths:  
  M /trunk/ivp/src/lib_geometry/XYPatternBlock.cpp  
  M /trunk/ivp/src/uLogView/LogPlotViewer.cpp  
  
minor fixes after compiling on Mavericks  
-----
```