Help Topic: The SVN Commit Command

Spring 2021

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

The SVN `commit` Command

The `svn commit` command sends changes from your working copy to the respository. A log message (even if empty) must be provided. The message can be given on the command line, from a file, or an editor may be launched as the commit proceeds. More info on `svn commit` can always be found by Googling "Subversion book" and reading the full PDF online free, or just typing `svn help commit` anytime on the command line.

Basic usage of the svn `commit` command

The shortest basic syntax for the svn `commit` command below. The shorthand version for `svn commit` is `svn ci`.

```
$ svn -m "" ci
```

The `svn commit` command *does* require a network connection to the server, and you will be prompted for a password (unless you have something liki ssh keys in use).

The `-m ""` option is essentially allowing the commit to proceed without any textual log file comment. This is generally not a good idea, especially the bigger and more complex the commit. But I often find this is reasonable especially if it's just me working on a development branch. I'll save the complex comment for the commit that merges the branch back into the trunk. The `-m ""` option could instead have been `--message "Fixing the butterfly bug"`.

Committing just a single file or directory

In the above example, by leaving any file or directory unspecified, the commit is applied to every outstanding pending change. The pending changes are viewable with the `svn status` command. You could have instead proceeded by:

```
$ svn commit -m "fixed the hanging input bug" KeyboardItems.cpp
$ svn commit -m "undid Paul's broken fix" Blackboard.cpp
$ svn commit -m "updated with italian additions" UtilitiesFolder/
```

After each commit, the file(s) committed are removed from the list of pending changes. So you can still finish things off with a `svn -m "" ci` after you've applied all the comments worth logging.

Configuring SVN with an editor of your choosing

Getting back to that issue of launching an editor to enter commit comments as described above. ”Which editor?” you may ask. The editor chosen is determined by how you have certain environment variables set in your shell configuration. It will first look for the value of the `SVN_EDITOR` variable, and fall back to the value for the `EDITOR` environment variable. If these are not set, svn may not allow you to select the (e) option.

If you’re a `tcsh` user, add the following lines to your `.cshrc` file, with the editor of your choosing:

```
setenv EDITOR 'emacs'
setenv SVN_EDITOR 'emacs'
```

If you’re a `bash` user, add the following lines to your `.bashrc` file, with the editor of your choosing:

```
export EDITOR=emacs
export SVN_EDITOR=emacs
```

To check the prevailing value of an environment variable, e.g., `EDITOR`, just type this on the command line: `echo $EDITOR`.