Getting svn to ignore files and directories

superchlorine.com/2013/08/getting-svn-to-ignore-files-and-directories/

superchlorine

August 27, 2013

Who knew it would be so hard to get svn (Subversion) to ignore some files and directories?

I'm working on an Android project, and I wanted svn to stop looking at me questioningly regarding files and directories that were automatically generated every time I built my source code. Basically, I needed svn to completely ignore the following:

- bin/ and gen/: directories with generated code
- proguard/: directory generated by my editor, Eclipse
- .classpath and .project: Eclipse project files
- local.properties: local config file
- Thumbs.db: annoying Windows thumbnail database files that are EVERYWHERE
- All built Android files, which have the extension .apk or .ap_
- All Java class files, which have the extension .class

I didn't think it would too difficult to get svn to ignore some files for me, but it turns out that svn really likes to pay attention to my files. Hours passed before I finally got svn to relax and ignore what I wanted it to. Below the cut, I share my newfound wisdom with you.

The svn:ignore property

svn has properties, which let you specify how your repository should be handled. One of these properties is `svn:ignore`. How this works is that you use the command `svn propset` to set the property `svn:ignore` on a particular directory. You give svn:ignore a value, which is a file name pattern. Then, svn will ignore all items in this directory whose name matches the pattern. For example:

```
svn propset svn:ignore *.class .
```

Here, you're telling svn to set the `svn:ignore` property, and what you want ignored are all files in the current directory (.) with the extension .class.

If you do `svn status` after executing that line, you'll find that svn will not show you any *.class files in the current directory that are not under version control. Ah, so much less output to sift through now. If you want `svn status` to tell you about the ignored files as well, you can do:

```
svn status --no-ignore
```

On ignoring directories

A short note that when specifying a directory to be ignored, you must not put any slashes before or after it! To ignore the directory “bin”, just type “bin”. “/bin” or “bin/” will crash and burn, and your “bin” directory will not be ignored. In short:

```
svn propset svn:ignore bin .     # yes
svn propset svn:ignore /bin .    # nope
svn propset svn:ignore bin/ .    # nope
```

Recursive property setting with -R

So, that was simple enough right? However, the command we used above only sets `svn:ignore` for the current directory: svn will not ignore *.class files in subdirectories! Fortunately, if we want *.class to be ignored in all subdirectories as well, we just need to add the `-R` (or `--recursive`) flag to specify that the command should be applied recursively:

```
svn propset svn:ignore -R *.class .
```

Like magic, *.class files are now ignored! Life is beautiful.

Ignoring multiple file types and items

Say you execute the following lines of code to ignore *.class, *.apk, and Thumbs.db files:

```
svn propset svn:ignore -R *.class .
svn propset svn:ignore -R *.apk .
svn propset svn:ignore -R Thumbs.db .
```

Now, you do `svn status`, and you see...what?! *.class files? *.apk files? In fact, the only files that are being ignored are those pesky Thumbs.db files. As I discovered, whenever you set `svn:ignore`, you’re *overwriting* whatever value was previously set. So, when we did `svn propset` for *.apk, we overwrote the `svn:ignore` setting for *.class, and the `svn:ignore` setting for *.apk was overwritten when we did `svn propset` for Thumbs.db!

To ignore multiple items, you’ll need to be tricky. Use quotations and put each name pattern to be ignored on its own line, and you’ll get to ignore a long list of items:

```
svn propset svn:ignore -R "*.class
> *.apk
> Thumbs.db" .
```

The > are, of course, just my shell prompts. You don’t type those in.

“svn, ignore all the names on this blacklist!”: the -F flag

If you added a new subdirectory after setting `svn:ignore`, your new subdirectory and everything in it **will not** be subject your `svn:ignore` settings! You will have to run `svn propset` again. Clearly, it’s not good to set `svn:ignore` manually like this every time:

you're wasting time, you might make typos, you might forget to include a pattern, and if you're working on a team, you can't share your awesome `svn:ignore` settings with everyone else!

Have no fear though, for the `-F` (`--file`) flag is here to rescue you! With `-F` , you can specify a file that contains a list of file name patterns to ignore. For example, this was my file for my Android project:

```
bin
gen
proguard
.classpath
.project
local.properties
Thumbs.db
*.apk
*.ap_
*.class
*.dex
```

I saved this file as .svnignore, and then I did:

```
svn propset svn:ignore -R -F .svnignore .
```

What did this do? Starting from the current directory, it recursively set `svn:ignore` with all of the patterns listed in .svnignore for each directory. Magic! And the best thing is that you can commit .svnignore to your repository as well, so that you and/or your team can use it again it the future.

Things to watch out for with this approach

Great power comes with great responsibility, and this is no exception. Notice that every pattern in the file was ignored for every subdirectory. This means that any directories called "bin", "gen", or "proguard" will be ignored by svn in my example, no matter how deeply these directories are nested. This is fine for my project, but if that's not true for you, you need to remember to go to those directories and change the `svn:ignore` setting. (You might want to write a shell script to help you with all this!)

Living with your ignorance

Now that you have `svn:ignore` settings applied, here are some commands to help you live with them more comfortably:

List all properties (including `svn:ignore`) set for the current directory. You can optionally specify a path, or use `-v` (`--verbose`) will list all the file patterns being ignored.

```
svn proplist -v [PATH]
```

Remove all your `svn:ignore` settings for the current directory. You can optionally specify a path, or use `-R` (`--recursive`) to delete the property recursively.

```
svn propdel svn:ignore [PATH]
```

As mentioned earlier, this will include all ignored files in your `svn status` output:

```
svn status --no-ignore
```