

Additional Information

I just wanted to mention a few other features of [coverage.py](#) without going into a lot of detail. First, [coverage.py](#) supports configuration files. The configuration file format is your classic “.ini” file with sections demarcated by the fact that they are surrounded with square braces (i.e. [my_section]). You can add comments to the config file using the following # or ; (semi-colon).

[Coverage.py](#) also allows you to specify what source files you want it to analyze via the configuration file we mentioned previously. Once you have the configuration set up the way you want it, then you can run [coverage.py](#). It also supports a “-source” command-line switch. Finally you can use the “-include” and “-omit” switches to include a list of file name patterns or exclude them. These switches have matching configuration values that you can add to your configuration file too.

The last item that I want to mention is that [coverage.py](#) supports plugins. You can write your own or download and install someone else’s plugin to enhance [coverage.py](#).

Wrapping Up

You now know the basics of [coverage.py](#) and what this special package is useful for. [Coverage.py](#) allows you to check your tests and find holes in your test coverage. If you aren’t sure you’ve got your code tested properly, this package will help you ascertain where the holes are if they exist. Of course, you are still responsible for writing good tests. If your tests aren’t valid but they pass anyway, [coverage.py](#) won’t help you.