

# Installing from Source

Installing from source is a great skill to have. There are easier ways, which we'll be getting to later on in the article. However, there are some packages that you have to install from source. For example, to use **easy\_install**, you will need to first install **setuptools**. To do that, you will want to download the tar or zip file from the Python Package Index and extract it somewhere on your system. Then look for the **setup.py** file. Open up a terminal session and change directories to the folder that contains the setup file. Then run the following command:

```
python setup.py install
```



If Python isn't on your system path, you will receive an error message stating that the *python* command wasn't found or is an unknown application. You can call this command by using the full path to Python instead. Here's how you might do it if you were on Windows:

```
c:\python34\python.exe setup.py install
```



This method is especially handy if you have multiple versions of Python installed and you need to install the package to different ones. All you need to do is type the full path to the right Python version and install the package against it.

Some packages contain C code, such as C header files that will need to be compiled for the package to install correctly. On Linux, you will normally already have a C/C++ compiler installed and you can get the package installed with minimal headaches. On Windows, you will need to have the correct version of Visual Studio installed to compile the package correctly. Some people say you can use MingW too, but I have yet to find a way to make that work. If the package has a Windows installer already pre-made, use it. Then

work. If the package has a windows installer already pre made, use it. Then you don't have to mess with compiling at all.