Using easy_install

Once you have **setuptools** installed, you can use **easy_install**. You can find it installed in your Python installation's **Scripts** folder. Be sure to add the Scripts folder to your system path so you can call easy_install on the command line without specifying its full path. Try running the following command to learn about all of easy_install's options:

```
easy_install -h
```

When you want to install a package with easy_install, all you have to do is this:

```
easy_install package_name
```

easy_install will attempt to download the package from PyPI, compile it (if necessary) and install it. If you go into your Python's **site-packages** directory, you will find a file named **easy-install.pth** that will contain an entry for all packages installed with easy_install. This file is used by Python to help in importing the module or package.

You can also tell easy_install to install from a URL or from a path on your computer. It can also install packages directly from a tar file. You can use easy_install to upgrade a package by using **-upgrade** (or **-U**). Finally, you can use easy_install to install Python eggs. You can find egg files on PyPI and other locations. An egg is basically a special zip file. In fact, if you change the extension to .zip, you can unzip the egg file.

Here are some examples:

```
easy_install -U SQLAlchemy
easy_install http://example.com/path/to/MyPackage-1.2.3.tgz
easy_install /path/to/downloaded/package
```

There are some issues with easy_install. It will try to install a package before it's finished downloading. There is no way to uninstall a package using easy_install. You will have to delete the package yourself and update the easy-install.pth file by removing the entry to the package. For these reasons and others, there was movement in the Python community to create something different, which caused **pip** to be born.