

Debian Package Creation Workflow

 coursera.org/learn/linux-tools-for-developers/supplement/EjX8S/debian-package-creation-workflow

Building a Debian package requires many more files than does the RPM system, where almost all the complications go into the **spec** file.

The most authoritative documentation is the Debian New Maintainers' Guide.

At first glance, it is almost frighteningly complicated. However, there exist auxiliary utilities such as **debuild** and **cdb**s which can automate many tasks, provide templates for files which do not often have to be complex, and give command line interfaces for updating the various configuration files.

We will begin by explaining the basic package building workflow:

First you will need the upstream package source, provided as a tarball in the form:

1

`package-version.tar.gz`



such as **someprogram-1.0.tar.gz**. Note the use of the hyphen (-) between the package name and the version number. The system is very picky about the name; it must contain only lower case letters, digits, plus and minus signs and periods. It cannot contain dashes or underscores.

- You will also need any patches that need to be made to the upstream source.
- The source will be expanded into **someprogram-1.0** and the original source tarball will be preserved in:

1

`package_version.orig.tar.gz`



Note the use of the underscore which has replaced the hyphen in the name; this can be very confusing!

- A directory **someprogram-1.0/debian** is created, and will be filled with all the files needed by the package building system.
- Source and binary packages are built from these ingredients; the binary package has the **.deb** suffix; we will talk about what a source package looks like later in an exercise (unlike RPM, where the source package is just one file, Debian source packages contain several files).



Completed
