How to Understand

Debian Packaging

Rev 1 2010-04-17

A visual guide by Martin Owens

Requirements

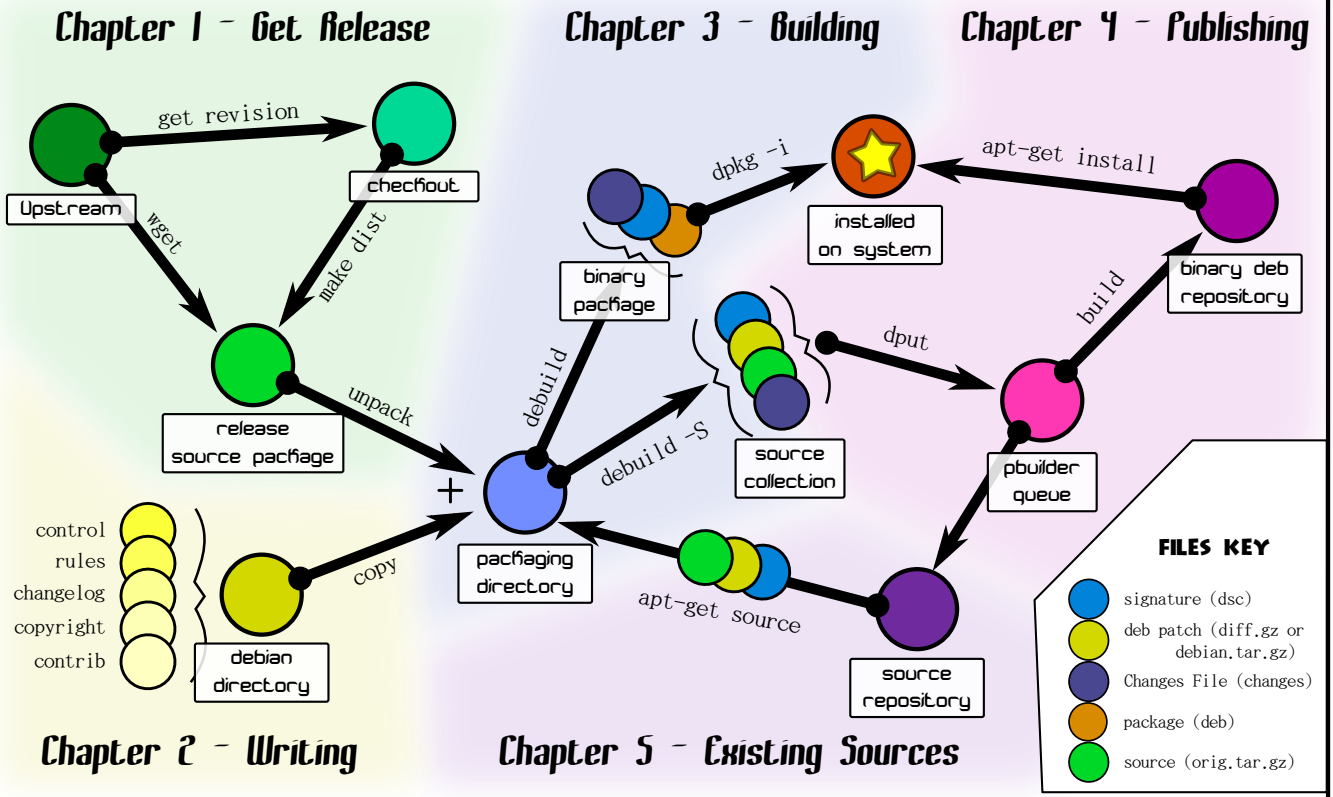
Confident familarity with the Bash command line.

Experence of installing packages and managing repositories.

Some understanding of where installed files go in a posix/fdo standard system.

Awareness of copyright licenses and how it effects redistribution rights and packaging.

Contents


Chapter 1 - Get Release
Chapter 3 - Building
Chapter 4 - Publishing
get revision
Upstream
checkout
wget
make dist
release source package
unpack
control
rules
changelog
copyright
contrib
debian directory
copy
+
packaging directory
debuild
debuild -S
binary package
dpkg -i
installed on system
source collection
dput
pbuilder queue
build
binary deb repository
apt-get install
apt-get source
source repository
Chapter 2 - Writing
Chapter 5 - Existing Sources
FILES KEY
signature (dsc)
deb patch (diff.gz or debian.tar.gz)
Changes File (changes)
package (deb)
source (orig.tar.gz)

Get Release

Chapter 1

The simple way: Download an offical release

Foo Website

1. Go to offical upstream url

2. Download release source package

Foo Release 1.7.2

3. Unpack the source code into It's own versioned directory.

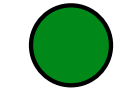
Foo Release 1.7.2

4. You're ready to add the debian directory to make it a package.
see Chapter 2

Releases are very important when it comes to debian packages, they define what is supposed to be in the source package and should never include developer files.

Thus never commit the debian directory to your upstream revision control.

Get Release



The source control way: Make your own release.

Foo Trunk Branch

1. Download target source repository

My Foo Checkout

2. Using the build tools, make a new release source package.

see table right

Foo Release

1.7~20100504

3. Unpack the source code into It's own versioned directory.

Foo Package

1.7~20100504

4. You're ready to add the debian directory to make it a package.

see Chapter 2

Each type of source code will have different commands to produce a new release. This is a short table of common release commands:

C/C++: `make dist`
py: `./setup.py sdist`
perl: unknown
ruby: unknown