An Introduction to Debian Packaging

Tom Most

Board Member at Large, NBLUG

February 11, 2014

Outline

What is Packaging?

- *Package* — a file containing software
- Distributions manage software *repositories* of packages
- *Packaging* is taking software and putting it in the file format used by a distribution

Package Formats

- Fedora, Red Hat, SUSE: `.rpm`, via `yum`
- Arch: `pkgbuild`, via `pacman`
- Debian, Ubuntu, Mint: `.deb`, via `dpkg` and `apg-get`/`aptitude`

What's in a .deb?

```
$ apt-get download hello
Get:1 Downloading hello 2.8-4 [28.1 kB]
Fetched 28.1 kB in 0s (77.9 kB/s)

$ ar t hello_2.8-4_amd64.deb
debian-binary
control.tar.gz
data.tar.gz

$ ar x hello_2.8-4_amd64.deb
$ file debian-binary
debian-binary: ASCII text
$ cat debian-binary
2.0
```

What's in a .deb?

```
$ tar -xvf control.tar.gz
./
./control
```

What's in a .deb?

```
Package: hello
Version: 2.8-4
Architecture: amd64
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.u
Original-Maintainer: Santiago Vila <sanvila@debian.org>
Installed-Size: 108
Depends: libc6 (>= 2.14), dpkg (>= 1.15.4) | install-info
Section: devel
Priority: optional
Homepage: http://www.gnu.org/software/hello/
Description: The classic greeting, and a good example
 The GNU hello program produces a familiar, friendly greet
```

What's in a .deb?

```
$ tar -tf data.tar.gz
./
./usr/
./usr/bin/
./usr/bin/hello
./usr/share/
./usr/share/info/
./usr/share/info/hello.info.gz
./usr/share/doc/
./usr/share/doc/hello/
./usr/share/doc/hello/NEWS
./usr/share/doc/hello/copyright
./usr/share/doc/hello/changelog.Debian.gz
./usr/share/man/
./usr/share/man/man1/
./usr/share/man/man1/hello.1.gz
```

What's in a .deb?

```
# dpkg -i hello_2.8-4_amd64.deb
Selecting previously unselected package hello.
(Reading database ... 270621 files and directorie
Unpacking hello (from hello_2.8-4_amd64.deb) ...
Setting up hello (2.8-4) ...
Processing triggers for install-info ...
Processing triggers for man-db ...

$ hello
Hello, world!
```

From Whence the .deb Came?

Upstream project releases	`hello_1.2.3.tar.gz`

Debian maintainer creates *source package*	`hello_1.2.3-1.dsc`
	`hello_1.2.3.orig.tar.gz`
	`hello_1.2.3-1.debian.tar.gz`

Binary packages are built	`hello_1.2.3-1_i386.deb`
	`hello_1.2.3-1_amd64.deb`
	`hello_1.2.3-1_armel.deb`

What Might the Maintainer Change?

- Metadata like dependencies — `debian/control`
- Instructions for the build system — `debian/rules`
- Changes to fix bugs or conform to Debian policy — patches, man pages, init scripts
- Scripts run on install/deinstall — `debian/preinst`, `debian/postinst`, `debian/prerm`, `debian/postrm`

Debian Policy

- Documentation on how all this works, found in the `debian-policy` package or online
- Not the enemy!
- ...but not always useful outside of the Debian project, either.

Building Your Own Packages

- It doesn't have to be as hard as all that
- Meet `debhelper` and friends
- `# apt-get install debhelper dh-make devscripts build-essential`

Make Me an Upstream!

Our “upstream” tarball, `hello-py-1.0.tar.gz`, contains a single file:

```
#!/usr/bin/python2.7

print "Hello, world!"
```

Make Me a Package!

```
$ tar -xvf hello-py-1.0.tar.gz
hello-py-1.0.0/
hello-py-1.0.0/hello-py

$ cd hello-py-1.0
```

Make Me a Package!

```
$ dh_make -f ../hello-py-1.0.0.tar.gz
Type of package: single binary, indep binary, multiple binary,
library, kernel module, kernel patch?
 [s/i/m/l/k/n] i

Maintainer name  : Tom Most
Email-Address    : twm@...
Date             : Tue, 11 Feb 2014 01:25:06 -0800
Package Name     : hello-py
Version          : 1.0.0
License          : blank
Type of Package  : Independent
Hit <enter> to confirm:
Currently there is no top level Makefile. This may require
additional tuning. Done. Please edit the files in the debian/
subdirectory now. You should also check that the hello-py
Makefiles install into $DESTDIR and not in / .
```

The template debian directory

```
$ ls
debian/  hello-py*

$ ls debian/
changelog                init.d.ex         prerm.ex
compat                   manpage.1.ex      README.Debian
control                  manpage.sgml.ex   README.source
copyright                manpage.xml.ex    rules*
docs                     menu.ex           source/
hello-py.cron.d.ex       postinst.ex       watch.ex
hello-py.default.ex      postrm.ex
hello-py.doc-base.EX     preinst.ex
```

debian/control

```
Source: hello-py
Section: unknown
Priority: extra
Maintainer: Tom Most <twm@...>
Build-Depends: debhelper (>= 8.0.0)
Standards-Version: 3.9.4
Homepage: <insert the upstream URL, if relevant>

Package: hello-py
Architecture: all
Depends: ${misc:Depends}
Description: <insert up to 60 chars description>
 <insert long description, indented with spaces>
```

debian/changelog

```
hello-py (1.0.0-1) unstable; urgency=low

  * Initial release (Closes: #nnnn)
    <nnnn is the bug number of your ITP>

 -- Tom Most <twm@...>  Tue, 11 Feb 2014 01:25:06 -0800
```

Other `debian` control files

Required files:

- `debian/compat` — version of this source package
- `debian/copyright` — license information
- `debian/rules` — Makefile which does everything (but really just calls `dh`)

You can delete the rest.

Finishing Up

- No, not yet.
- `hello-py` should go in `/usr/bin`
- `dh_install(1)` is the right tool
- Define `debian/install`: `hello-py usr/bin`

Building the Package

```
$ dpkg-buildpackage -uc -us
dpkg-buildpackage: source package hello-py
dpkg-buildpackage: source version 1.0.0-1
dpkg-buildpackage: source changed by Tom Most <twm@...>
dpkg-buildpackage: host architecture amd64
 dpkg-source --before-build hello-py-1.0.0
 fakeroot debian/rules clean
dh clean
   dh_testdir
   dh_auto_clean
   dh_clean
 dpkg-source -b hello-py-1.0.0
dpkg-source: info: using source format ‘3.0 (quilt)’
dpkg-source: info: building hello-py using existing ./hello-py_1.0.0.orig.tar.gz
dpkg-source: info: building hello-py in hello-py_1.0.0-1.debian.tar.gz
dpkg-source: info: building hello-py in hello-py_1.0.0-1.dsc
 debian/rules build
dh build
   dh_testdir
   dh_auto_configure
   dh_auto_build
   dh_auto_test
 fakeroot debian/rules binary
dh binary
   dh_testroot
   dh_prep
   dh_auto_install
   dh_install
   dh_installdocs
   dh_installchangelogs
   dh_perl
   dh_link
   dh_compress
   dh_fixperms
   dh_installdeb
```

Building the Package

```
$ ls ..
hello-py-1.0.0/
hello-py_1.0.0-1_all.deb
hello-py_1.0.0-1_amd64.changes
hello-py_1.0.0-1.debian.tar.gz
hello-py_1.0.0-1.dsc
hello-py_1.0.0.orig.tar.gz
hello-py-1.0.0.tar.gz
```

Use `dpkg-buildpackage -uc -us -b` to only build the .deb.

What if I'm Upstream?

- Just drop a minimal `debian` directory into your source tree
- The Debian folks will ignore it

Releasing the Next Version

- Use `dch` to add a new entry to `debian/changelog`
- "New upstream release." is the stock comment

Additional Resources

- `debhelper(7)` — the essential reference
  - `dh_installinit(1)` — init scripts/Upstart jobs
  - `dh_installdirs(1)`
- `pbuilder` for clean builds and easy 32-bit builds
- `man deb-triggers` — triggers let packages communicate at install time