Debian packaging
tutorial for newcomers

Tiago Bortoletto Vaz

What is Debian?

And then I realised. My kind of people ... I am surrounded by my kind of people.

Debconf 24
Busan, South Korea

Next year:
Debconf 25
Brest, France

Over 1,000 volunteers

- package maintainers
- bug reporters
- wiki editors
- translators
- designers
- package sponsors / mentors / reviewers
- press contributors
- sysadmins
- outreaching
- quality assurance
- list moderators
- debconf attendees
- debconf speakers
- local gathering organizers
- testers
- … and so on

What is a Debian package?

"A very powerful and convenient way to distribute software to users"

Concerns when installing (or removing) a software:

security

dependencies / conflicts

licensing

stability

compatibility

upgradability

How Debian address them:

Debian Policy

Debian Free Software Guidelines

Files System Hierarchy Standard

Clear distinction between "unstable", "testing" and "stable"

Auto testing / QA (debCI, autopkgtest, lintian, piuparts, etc)

Rigorous New Maintainer process

Web of trust and extensive checks of signatures

12 architecture ports

User-friendly tools

...

Why learn how to package for Debian

Be part of an amazing community of volunteers.

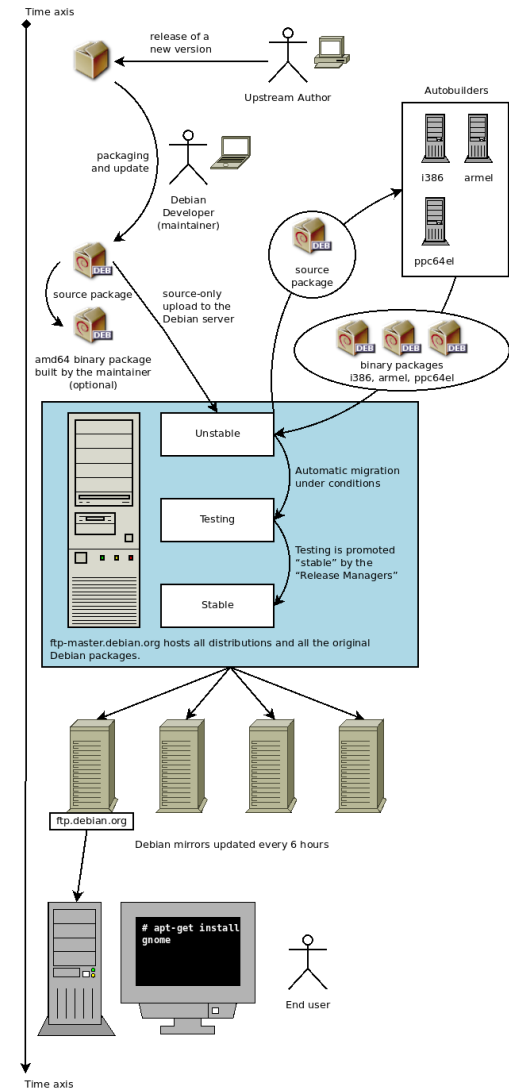Acquire a culture of technical excellence (and fun)

…and bring it to your daily life!

Learn a lot about Debian in general when packaging

…which can help you later performing other Debian work.

General packaging workflow

from "https://debian-handbook.info"
by Raphaël Hertzog and Roland Mas

Anatomy of a Debian package

Main files:

control – meta-data about the package (dependencies, etc.)
rules – specifies how to build the package
copyright – copyright information for the package
changelog – history of the Debian package

Other files:

- compat
- watch
- dh_install* targets
- *.dirs, *.docs, *.manpages, ...
- maintainer scripts
- *.postinst, *.prerm, ...
- source/format
- patches/ – if you need to modify the upstream sources

https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial
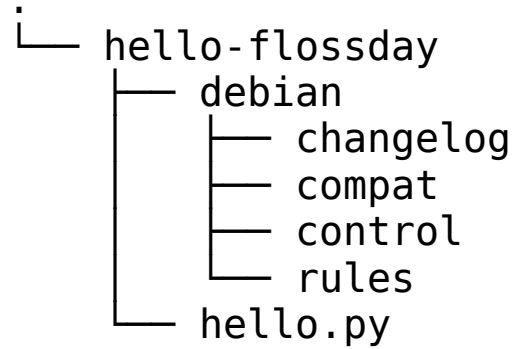
Anatomy of a Debian package

Main files:

control – meta-data about the package (dependencies, etc.)
rules – specifies how to build the package
copyright – copyright information for the package
changelog – history of the Debian package
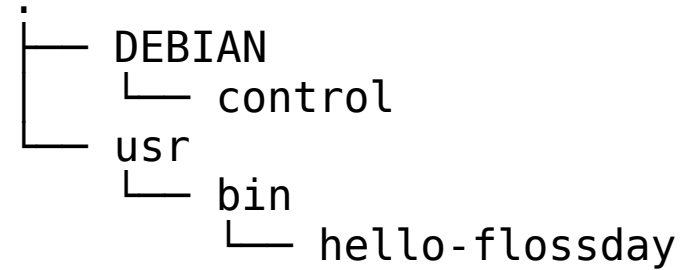
Other files:

- compat
- watch
- dh_install* targets
- *.dirs, *.docs, *.manpages, …
- maintainer scripts
- *.postinst, *.prerm, ….
- source/format
- patches/ – if you need to modify the upstream sources

https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial

Anatomy of a Debian package

Source

```
.
└── hello-flossday
    ├── debian
    │   ├── changelog
    │   ├── compat
    │   ├── control
    │   └── rules
    └── hello.py
```

→

Binary

```
.
├── DEBIAN
│   └── control
└── usr
    └── bin
        └── hello-flossday
```

Hands on — part 1

Building a simple package

git clone https://github.com/tiagovaz/vanier-flossday-2024.git

(demo + explanation)

Hands on — part 2

Building a (less) simple package

(demo + explanation)

Hands on — part 3

Modifying an existant Debian package

(demo + explanation)

Resources

https://www.debian.org/doc/manuals/debmake-doc/

https://wiki.debian.org/Packaging

https://feeding.cloud.geek.nz/posts/whats-in-a-debian-directory/

https://wiki.debian.org/BuildingTutorial

https://wiki.abrahamraji.in/simple-packaging-tutorial.html

https://wiki.debian.org/SimplePackagingTutorial

https://wiki.debian.org/HowToPackageForDebian

https://github.com/FooBarWidget/debian-packaging-for-the-modern-developer

https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial

```
 _____
|    < Questions? >     |
 -----------------------
        \    ^__^
         \   (Oo)_____
           (__)\         )\/\
               ||----w |
               ||     ||
```