Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Debian Packaging (Part 1)

Linux System Administration Meeting #2

Thursday, 24th July 2014 15:00 – 17:00

BFH – Room HALL 217 (Hallerstrasse, Bern)

Daniel Baumann <daniel.baumann@bfh.ch>

IT System Engineer, Infrastructure Team

Overview

Introduction

- BFH News

Debian Packaging, Part 1

- Theory
- Practice („by hand“)

Aperitif

Next meeting: Debian Packaging, Part 2 (September 2014)

- Overview of an Automated Package Infrastructure
- Practice („automagic“)

BFH News

BFH News

https://git.bfh.ch

- Fully user automated now

https://lists.bfh.ch

- Use our mailinglists, specifically bfh-linux-users@lists.bfh.ch

- **Linux User Meeting on 8th of August**

  Please come to next meeting, it's about OpenSSH and you should bring all your friends and collegues...

- **Desktop „Brain Storming“**

  Gathering Ideas/Needs for the BFH Linux Standard Client

Debian Packaging Theory

Debian Packages

- Reliable and homogenous way of installing, upgrading, removing, distributing Software (or associated data)
- Can be built by anyone, not just Debian people
- BFH ITS has it too
- Eventhough it solves all problems other operating systems and distributions are suffering from...
- ...debian packaging is an art mastered over the years. :)

Debian Source Packages (1/2)

State of the Art: Source Package Format 3.0 (quilt) with XZ

- One Upstream Tarball: **foo_1.2.3.orig.tar.xz** ...
- ...or multiple Upstream Tarballs: **foo_1.2.3.orig.tar.xz** and **foo_1.2.3.orig-{bar,baz}.tar.xz**
- Debian Packaging: **foo_1.2.3-1.debian.tar.xz**
- Debian Source Control: **foo_1.2.3-1.dsc** (GPG signed)

Alternatively: Source Package Format 3.0 (native) with XZ

- Upstream and Debian Packaging Tarball: **foo_1.2.3.tar.xz**
- Debian Source Control: **foo_1.2.3-1.dsc** (GPG signed)

Common to both native and non-native Formats

- Allowed legacy compression types for all tarballs: gzip, bzip2

Debian Source Packages (2/2)

Legacy: Source Package Format 1.0 „non-native“

- Upstream Tarball (non-native): **foo_1.2.3.orig.tar.gz**
- Debian Packaging: **foo_1.2.3-1.diff.gz**
- Debian Source Control: **foo_1.2.3-1.dsc** (GPG signed)

Legacy: Source Package Format 1.0 „native“

- Upstream and Debian Packaging Tarball: **foo_1.2.3.tar.gz**
- Debian Source Control: **foo_1.2.3-1.dsc** (GPG signed)

Common to both native and non-native Formats

- Allowed legacy compression types for the upstream tarball only: gzip, bzip2

Debian Binary Packages (1/2)

All Source Packages produce any of the following Debian packages

- Architecture Dependent Binary Package: **foo_1.2.3-1_amd64.deb**
- Architecture Independent Binary Packages: **foo_1.2.3-1_all.deb**
- (exception/special case: udebs for debian-installer)

Internal Structure

- Multimember ar archive (use 'ar x foo_1.2.3-1_amd64.deb' to unpack)

- **debian-binary:** Debian Package Version String
- **data.tar.xz:** contains the „upstream program files“
- **control.tar.gz:** contains the meta data and the maintainer scripts

- data.tar.xz can be compressed with different compression types, control.tar.gz is always gzip.

Debian Binary Packages (2/2)

Maintainer Scripts

- Per package specific
- Can do anything and are executed as root
- Can be in any language (theoretically; almost always shell, rarly python and perl...)

- Run before unpacking on the rootfs: preinst
- Run after unpacking on the rootfs: postinst

- Run before removal from the rootfs: prerm
- Run after removal from the rootfs: postrm

Debian Archive (1/2)

Structure

- Archive areas: **main contrib non-free**
- Pool: one pool per archive area, contains source and binary packages in own directory per package, e.g. **pool/main/f/foo/**.
- Distributions by Suites: **stable, testing, unstable, experimental**
- Distributions by Codename: **squeeze, wheezy, jessie, sid**
- Distributions by Version (stable only): **6.0.10, 7.6**
- Package indices refering to the packages in pool are localted in dists, e.g. **dists/stable/main/binary-amd64/Packages.xz**

Instances

- Debian Archives are mirrored around the world, redirectors (http.debian.net; prefered) and splitviews (cdn.debian.net; old-ish) are available.

Debian Archive (2/2)

Package Uploading

- Uploading to the archive master (ftp.upload.debian.org)
- Contents of the upload in the changes file, e.g. **foo_1.2.3-1.changes** (GPG signed)
- Needs to have **sources attached** if upstream tarball(s) not present in the archive
- Uploads uploading all packages (binary and source) are so called **sourceful** (usually from the maintainer)
- Uploads of binary packages (regardless if arch-dep or arch-indep) are so called **binary uploads** (usually from the buildd or porter)

Package Life Cycle

- Normally uploads go to unstable, migrate to testing and end up eventually in a stable release when Debian releases a new stable release of the distribution
- Shortcut: backports

Debian Versions

Versioning

- ${package_version} = ${upstream_version}-${debian_revision}
- Upstream versions can have dashes but no underscores,

  e.g. foo_**1.2.3-4**-5_all.deb
- Debian revision is, when reading backwards, the string up to the first dash,

  e.g foo_1.2.3-4-**5**_all.deb

Upgrade Path

- Strict versioning scheme
- Simplified: stable ≤ testing ≤ unstable ≤ experimental
- Additional repos in between: proposed-updates (+deb7u1), security (+deb7u1), backports (~bpo70+1)
- Special: + (always greater), ~ (always lower), NMUs (-0), and binNUMs (+b1)

  e.g. 1.2.3-1 < 1.2.3-2 < 1.2.3-2+foo < 1.2.3-2+foobar < 1.2.3-3~foo < 1.2.3-3~foobar < 1.2.3-3
- Workaround when having messed up: epoche (not shown in filenames though),

  e.g. 2.0-1 < 1:1.0-1 < 2:0.9-1

Testing Migration

Automatic testing migration

- If aged appropriatly to specified urgency (low=10d, medium=5d, high=1d, critical=asap)
- If built on as many architectures successfully as the package already in testing has
- If no more RC bugs than the package already in testing has
- All its depends are satisfied in testing
- Not blocked by a current migration

Automatic testing removals

- Unfixed RC bugs for x days in testing
- If depends get removed (not just usage or security bugs, but also uninstallability, FTBFS, etc.)

Debian Packaging Practice

Summary

chroot

- debootstrap sid sid http://ftp.ch.debian.org/debian
- Create an unprivileged user
- Install build-essential, fakeroot, and debian-keyring

Sources

- Download if in archive: apt-get source foo
- Download manually: dget http://ftp.ch.debian.org/debian/ ... /foo_1.2.3-1.dsc
- Unpack sources (if needed) with: dpkg-source -x foo_1.2.3-1.dsc

Binaries

- Build package with: dpkg-buildpackage
- Check upload with: lintian -i foo_1.2.3-1.changes
- Upload your package to the archive you're allowed to with dput

Thank You for Your Attention.

❤ Source Code is freely available

Debian Packaging: Part 2 (September 2014)

Infrastructure Setup

- Buildd (sbuild, wanna-build)
- Archive Master (reprepro)
- Archive Push Mirroring („ftpsync“)
- Usefull Notifications (mail, irc)
- Where Cryptography is involved (archive keys, keyrings)
- ...

Workflow

- Maintaining Packages with Git
- Building Packages completely out of a Git repository
- ...

Debian Packaging: Links

Debian

- Debian Policy: https://www.debian.org/doc/debian-policy/
- Debian Developer Reference: https://www.debian.org/doc/manuals/developers-reference/
- Debian New Maintainers' Guide: https://www.debian.org/doc/manuals/maint-guide/ (use with caution!)

...

Upstream related

- Filesystem Hierarchy Standard: http://www.pathname.com/fhs/

...