

Software Distribution and Package Management

Pieter Lexis

February 15, 2016

whoami

Pieter Lexis

- OS3 graduate
- SysAdmin for a long time
- DNS + DNSSEC Nerd
- Developer/SysAdmin at PowerDNS

POWERDNS 

Package managers

Who here ...

- Has used a package manager?
- Knows why packages and package managers exist?
- Can name me a package manager?
- Has ever created a package?

A package management system

- Automates installing, configuring, upgrading of software
- Does this in a consistent manner
- Ensures all requirements (dependencies) are in place
- Guarantees installations are repeatable and deterministic
- Looks like magic in the beginning *grin*

Installers versus package managers

Package manager

- Part of the OS
- Omnipotent
- One package format
- Installs dependencies

Installer

- For one software package
- No OS integration
- Multiple package formats
- Dependencies bundled

Concepts

Repository

A repository is
a location where packages (and the list of packages) are stored.

- Web-based
- In a VCS
- On the local filesystem

Package

A package is

an archive containing installable files, as well as metadata and helper scripts related to these files.

Metadata can be:

- Package maintainer information
- URL to the “upstream” software
- Dependency list

Helper script examples:

- Post-installation script
- Upgrade script

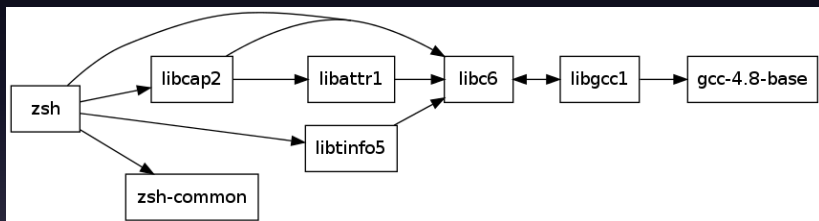
Dependencies

A dependency is

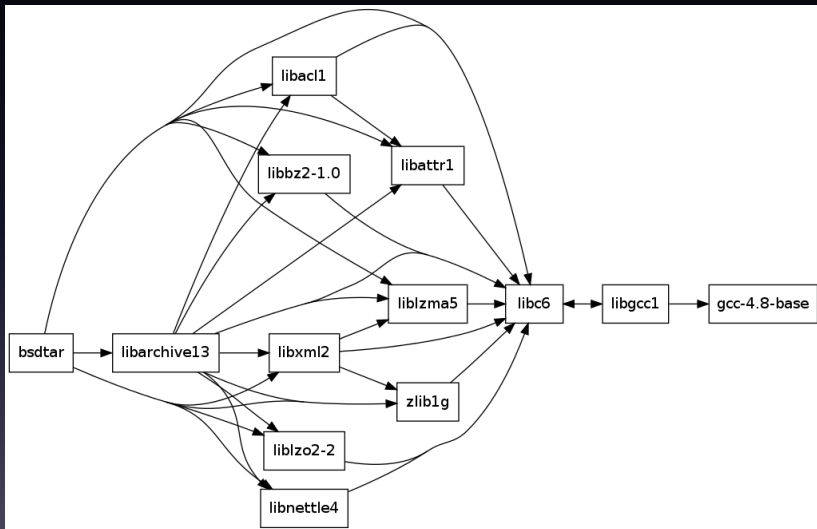
A package required for another package to function.

- Runtime dependency
 - Shared library
 - Interpreter
- Build-time dependency
 - Compiler
 - Documentation generators

Dependencies – zsh



Dependencies – bsdtar



*NIX Package Management

Debian

Debian ecosystem – Tools

apt

- Library for repository interaction
- Enforces package policies
- Handles the dependency management
- Downloads and verifies packages

dpkg

- Low-level tool for local package management
- Handles actual package installations (and errors!)
- Keeps the state of all packages

Debian ecosystem – deb format

- Unix ar archive with 3 files
 - debian-binary
 - control.tar{, .gz, .xz}
 - data.tar{, .gz, .bz2, .lzma, .xz}
- Separate source package format
- Repositories publish signed lists with checksums

Debian ecosystem – Versioning

- Format: [epoch:]upstream_version[-local_revision]
- Higher is always newer
- Special rules for pre-releases

Ordering example

$0.5 < 0.10 < 0.99 < 1 < 1.0\sim rc1 < 1.0 < 1.0+nmu1 < 1.1$

Overview

- Repositories contain lists and packages
 - Separated by release and architecture
 - Lists are signed
- `apt-get update` retrieves and verifies the lists
- When installing a package
 - Calculates the dependencies needed
 - Downloads the requested packages and their dependencies
 - Verifies the downloaded packages
 - Calls `dpkg` to install the packages
 - `dpkg` unpacks and configures the packages¹

¹By calling the helperscripts

Red Hat

Red Hat ecosystem – Tools

Frontends

- Differ per distro
 - yum – Red Hat, CentOS, Fedora
 - Zypper – openSUSE
 - urpmi – Mandriva
- Repository and dependency management

rpm

- Low-level tool for interaction with packages
- Gateway to the local package database

Red Hat ecosystem – rpm format

- Binary format
 - Lead – Identifying the file as RPM
 - Signature – To verify integrity
 - Header – Package metadata
 - File Archive – in `cpio` format (optionally compressed)
- Source packages (SRPM) have the same format

Red Hat ecosystem – Versioning

Format: `version-release[.distributor].architecture`

Ordering example

`1.4-0.1.a < 1.4-0.2.a < 1.4-0.4.b < 1.4-0.5.b < 1.4-1`

FreeBSD

Ports

- Categorized collection of small Makefiles
- Makefile calls functions of the pkg system
- Install-time == build-time
- Installation:
 - Resolve dependencies
 - Download sources and patches
 - Verify sources and patches
 - Build and install
 - Register in the local database

pkg-ng

- Repository package list is a signed SQLite database
- One tool (`pkg`) to rule them all

Package Format

- Compressed Tar archive with:
- Metadata file (+METADATA)
 - YAML
 - Scripts are included *in* this file
- All files in the filesystem tree

Other systems

OS-level Systems

- Gentoo eBuild
- NetBSD pkgsrc
- Arch Linux pacman
- Homebrew for Mac

Honorable mention

All the various app-stores

Other-Other Systems

- OS vendors can be slow/hesitant to include software
- Application-based package-managers
- *Usually* used to install on a user level

Other-Other Systems – Examples

- Bundler + RubyGems
- CPAN
- Cabal
- Composer
- Maven
- PyPi
- npm

Building Packages

The Big Picture

- 1 Get and unpack upstream sources
- 2 Apply distribution specific patches
- 3 `./configure && make && make DESTDIR=../build install`
- 4 Run distribution packaging scripts
- 5 ???
- 6 Profit

The Debian Way

Different packaging Systems

- cdb_s – Common Debian Build System
- dh
- debhelper – Wrappers for dh

Source package

Consists of 3 files:

- `package_${version}-${debianversion}.dsc`
- `package_${version}.orig.tar.gz`
- `package_${version}-${debianversion}.debian.tar.xf`

The Debian Way – debian directory

- `control` – Package metadata
- `rules` – Makefile that calls all `dh_*` functions
- `patches/` – Distribution specific patches
- Files that hint debhelper what directories and files to install

The Red Hat Way

- Source RPM contains a `.spec` file
- This plain text file contains *everything*:
 - Metadata
 - Build commands
 - Maintainer scripts
 - Changelog
- Most steps are called as macros

When you don't care

Enter fpm

The F*cking Package Manager

“I want a simple way to create packages without all the bullshit. In my own infrastructure, I have no interest in Debian policy and RedHat packaging guidelines - I have interest in my group's own style culture and have a very strong interest in getting work done.”

– Jordan Sissel

Enter fpm

- Can create packages in many formats
 - deb
 - rpm
 - OS X
 - Solaris
- Extremely easy to use
- Best used in combination with configuration management

Thank you

Questions?

Links

- <https://wiki.debian.org/IntroDebianPackaging>
- <https://joeyh.name/talks/debhelper/debhelper-slides.odp>
- <https://www.debian.org/doc/manuals/maint-guide/>
- <https://wiki.freebsd.org/pkgng>
- <http://rpm.org/max-rpm/s1-rpm-file-format-rpm-file-format.html>
- <http://fedoraproject.org/wiki/Packaging:NamingGuidelines>
- <http://www.slideshare.net/AdamGonnerman/advanced-apt>