

Getting Stuff Packaged for Debian

Roberto C. Sánchez

Connexer, Ltd.

<http://www.connexer.com>

roberto@connexer.com

May 1, 2011

Presented at Linuxfest Northwest

Bellingham Technical College, Bellingham, WA

What Is This All About?

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

So, you have just discovered (or perhaps even written) a cool program. You think, “it would be neat if I could quickly and easily make this available to thousands of users.” There are lots ways, but getting the software packaged for Debian (or packaging it yourself), is a great way!

Overview

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

- Objective
- A Little Bit About the Presenter
- Why Bother?
- Licensing and Redistribution
- Structuring Software to Facilitate Packaging
- The Debian Release Cycle
- New Upstream Releases
- Recruiting Help
- Documentation
- Pitching In to Help
- Beyond Debian
- Summary

Objective

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

- What this presentation is:
 - Tips/pointers/ideas on facilitating the packaging of software for Debian
- What this presentation is not:
 - A technical HOWTO on the mechanics of packaging for Debian (see an earlier [presentation](#) I gave or the [Debian New Maintainer Guide](#) for technical details)
- When this presentation is over:
 - You should want to get some software packaged for Debian

A Little Bit About the Presenter

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

- Using Debian since 2002
- Contributing to Debian since 2003
- Entered Debian New Maintainer queue on June 20, 2005
- Started *Connexer, Ltd.* in January, 2006
- Became official Debian Developer on March 13, 2007
- Currently maintain 39 [Debian packages](#) and sponsor uploads for several prospective Debian Developers

Why Bother?

- **Simplicity**
 - Packaging is straightforward in the vast majority of cases
 - Newer `debhelper` versions handle most tasks
 - Documentation is plentiful, and lots of help is available
- **Exposure**
 - Debian has many users (hundreds of thousands, millions?)
 - Debian has [100+ derivatives](#), each with its own user base
 - Potentially attract new contributors to your project
- **Portability**
 - Debian officially supports 9 hardware architectures
 - Built for additional non-released architectures
- **Ease of installation/configuration/upgrade**
 - Installation as easy as `'apt-get install <package>'`
 - Configuration managed with `debconf`, etc.
 - Upgrade as easy as `'apt-get upgrade <package>'`
- **Improve the software ecosystem**

Licensing and Redistribution

- Make sure to use a [Debian Free Software Guidelines](#) (DFSG) compatible license
 - Examples: GPL, BSD, Artistic
 - Note: The [Open Source Definition](#) is a generic version of the DFSG
- Make certain that all components of the software are properly licensed
 - Make sure appropriate license and copyright info follow components taken from other projects
 - Don't mix components under incompatible licenses
- Don't assume that distribution medium implies license or redistributability (e.g., making available via CPAN alone does not apply the "same terms as Perl itself" license)
- Don't get upset with people who want to package your software for Debian (this doesn't happen often, but it does happen)

Structuring Software to Facilitate Packaging

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

- DO NOT embed entire libraries or code copies
- Use standard build infrastructure:
 - C/C++ - autotools `./configure; make; make install`
 - Java - ant
 - Perl - `Module::Build`
- Make the application **FHS** compliant
- Facilitate installation to destination outside of prefix:
 - `./configure --prefix=/usr`
 - `make all`
 - `make install DESTDIR=/tmp/foo-1.2.3/debian/tmp`
 - `/usr/bin/bar` should be found at `/tmp/foo-1.2.3/debian/tmp/usr/bin/bar`
- Make sure to properly **manage SONAME for libraries**
- If your software checks for updates, allow disabling feature
- Lots more (just ask on *debian-mentors* or other lists)

The Debian Release Cycle

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

- The Debian Project makes a stable release every 18-24 months
 - *Squeeze*, the current *stable*, was released on Feb 6th
- Once *stable* is released, only security fixes are possible
- New packages enter through *unstable*
- Following a transition period with no release critical (RC) bugs, the package propagates to *testing*
- Prior to the *stable* stable, a freeze of *testing* occurs, where propagation from *unstable* is curtailed unless it specifically fixes bugs

New Upstream Releases

- Keep copyright and license information up to date
- Release with standard tarballs (e.g., `hello-world-1.2.3.tar.gz`)
- Make releases easy to find for [Debian External Health Status](#) (DEHS)
 - A listable directory on a HTTP or FTP server is best
 - A project hosting service (e.g., CPAN, SourceForge, Berlios, GitHub, Google Code) is also OK
- Don't ship a `debian/` directory in your release
- Don't ship VCS directories when it doesn't make sense (e.g., `CVS/` and `.svn/` not OK, but `.git/` may be OK)
- Make security updates available for previous stable releases
- Remember that the release will not directly reach current users of *stable*

Recruiting Help

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

- File a [Request For Package](#) (RFP) bug
- Check with an already formed [Debian Team](#)
 - Teams formed around specific applications/frameworks
 - Debian Kernel Team
 - Debian GNOME Team
 - Teams formed around specific languages/interpreters
 - Debian Java Packaging Team
 - Debian Perl Group
 - Debian Python Applications Packaging Team
 - Teams formed around general categories of software
 - Debian Games Team
 - Debian GIS Team
 - Debian Scientific Computing Team
 - Debian Crosswire Packaging Team
- Make an announcement on your project website
- Hire someone to do the work for you

Documentation

- Provide good quality installation/upgrade documentation:
 - Deviations from or extensions to your language's build conventions (e.g., Perl, Ruby, OCaml, etc.)
 - Additional manual actions (e.g., database setup or configuration)
- Include manual pages for executables, config files, etc.
- Include detailed change logs and release notes

Pitching In to Help

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

- On the Debian side
 - Search [Work Needing and Prospective Packages](#) (WNPP) for RFP bugs, or stale Intent To Package (ITP) bugs
 - Join one of the aforementioned teams
 - Help triage bugs as they are reported, and help solve them once the software gets packaged
 - Package the software yourself, and get it uploaded by a sponsor (via [mentors.debian.net](#))
 - Take the plunge and [become an official Debian Developer](#)
- On the upstream side
 - Involve the maintainer of the Debian package(s) of your software in major structure changes
 - Acknowledge security issues and release fixes in a timely manner
 - Accept patches from the Debian maintainer

Beyond Debian

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

- See if your favorite derivative will automatically pick up the new package at some point
- Request a [Sync to Ubuntu](#)

Summary

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

There are lots of reasons to get software packaged for Debian, and lots of ways to do it yourself or get help (whether you are an upstream developer or an interested user). Have at it!

Questions?

Getting Stuff
Packaged for
Debian

Roberto C.
Sánchez

Questions?