

Connexer Ltd.

Packaging Software for Debian

Roberto C. Sánchez
Ohio Linuxfest
Saturday, September 29, 2007

Overview

- Slides available online:
 - <http://people.connexer.com/~roberto>
- Reasons to package software for Debian
- Packaging resources
- Components of a *debian/* directory
- Assembling the toolbox
- Packaging the software

Packaging resources

- Debian New Maintainers' Guide
- Debian policy manual (and sub-policies)
- #debian-mentors on OFTC IRC
- debian-mentors@lists.debian.org
- <http://www.debian.org/doc/devel-manuals>
- Google
- Hire me :-)

Reasons to package for Debian

- Altruism/Contribute
- Build a better OS
- Learn something
- Help your company sell/market product
- Distribute software you wrote yourself
- It is (can be) fun
- It is (can be) easy
- It is a challenge
- You can target a larger audience
- It will help you on the road to being a Debian Developer

debian/ directory components

- *README.Debian* – notes to users
- *changelog** – changes and bug closures
- *compat** – debhelper compatibility level
- *control** – package descriptions/priorities
- *copyright** – license and copyright info
- *cron.d* – if the package needs a cron job
- *defaults* – package default settings

debian/ directory components

- *dirs** – directories which must be created
- *docs* – documentation files
- *examples* – sample code/scripts to install
- *init.d* – initialization script
- info/man pages – if not included upstream
- *menu* – place binaries in Debian menu
- *patches/* - location of dpatch patches

debian/ directory components

- *{pre,post} {inst,rm}* – maintainer scripts
- *rules** – makefile for building package(s)
- *shlibs* – library version information
- *watch* – automate process of updating to new upstream releases
- **.install* – for creating multiple binary packages (e.g., shared and static libs)
- Anything with an asterisk (*) is required

Assembling the toolbox

- “Debian Package Customization HOWTO”
 - Describes how to modify existing packages
 - Explains how to setup the necessary tools
 - online: <http://people.connexer.com/~roberto>
- Necessary packages: dh-make, piuparts, devscripts, debhelper, dpkg-dev, sudo, fakeroot, pbuilder, gnupg, lintian, linda
- Other: local mirror or fast 'net connection
- Find some software to package

About the example

- Name: tuition-ledger
- Developer: me
- License: GPL
- Status: unreleased as of yet
- Purpose: database-driven application to track tuition obligations and payments for a small Christian school

Packaging the software

- Obtain the source tarball
- Debianize the source using `dh_make`
- Remove unnecessary files from *debian/*
- Rename to remove `.ex` extensions
- Edit remaining files
- Add any new needed files
- Build the package

Obtain source and Debianize

```
roberto@miami: ~/tuition-ledger/tuition-ledger-0.0.1
File Edit View Terminal Tabs Help
tuition-ledger-0.0.1/missing
tuition-ledger-0.0.1/NEWS
tuition-ledger-0.0.1/ChangeLog
roberto@miami:~/tuition-ledger$ cd tuition-ledger-0.0.1/
roberto@miami:~/tuition-ledger/tuition-ledger-0.0.1$ dh_make -c gpl -s -f ../tui
tion-ledger-0.0.1.tar.gz
Maintainer name : Roberto C. Sanchez
Email-Address   : roberto@connexer.com
Date            : Sat, 29 Sep 2007 00:43:06 -0400
Package Name    : tuition-ledger
Version         : 0.0.1
License         : gpl
Type of Package : Single
Hit <enter> to confirm:
Done. Please edit the files in the debian/ subdirectory now. tuition-ledger
uses a configure script, so you probably don't have to edit the Makefiles.
roberto@miami:~/tuition-ledger/tuition-ledger-0.0.1$ ls debian/
README.Debian  dirs                manpage.1.ex      preinst.ex
changelog      docs                manpage.sgml.ex  prerm.ex
compat         emacsen-install.ex  manpage.xml.ex    rules
control        emacsen-remove.ex  menu.ex           tuition-ledger-default.ex
copyright      emacsen-startup.ex  postinst.ex       tuition-ledger.doc-base.EX
cron.d.ex      init.d.ex           postrm.ex         watch.ex
roberto@miami:~/tuition-ledger/tuition-ledger-0.0.1$
```

Edit remaining files

- Edit *changelog* with ``debchange``; close ITP (for first release) or note changes
- Place author, copyright and license information in the *copyright* file
- Review all files in upstream tarball and note differently licensed files in *copyright*
- Update list of directories to create in *dirs*
- Create menu entry in *menu* file (check menu sub-policy)

Edit remaining files

- Create *watch* file entry
 - Follow examples included in *watch.ex*
 - Test with ``uscan --verbose --report`` from top-level source directory
 - This is used in the future to check for new releases
- Edit maintainer scripts as necessary

The *control* file

- Consult Policy 2.4 and 2.5 for section and priority information
- Add short and long descriptions
- Add build-time dependencies (any libraries or programs needed to compile and/or build the software)
- Add non-library dependencies, recommendations and suggestions
- Specify arch: *any*, *all*, or a list

The *control* file

```
control + (~ / tuition-ledger / tuition-ledger-0.0.1 / debian) - VIM
File Edit View Terminal Tabs Help
Source: tuition-ledger
Section: misc
Priority: extra
Maintainer: Roberto C. Sanchez <roberto@connexer.com>
Build-Depends: debhelper (>= 5), autotools-dev, libpqxx-dev, libwxgtk2.6-dev
Standards-Version: 3.7.2

Package: tuition-ledger
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}
Suggests: postgresql-8.1
Description: help track tuition payments for a small Christian school
 This is a wxWidgets-based application that runs against a database backend
 in order to track tuition obligations and payments for a small Christian
 school.

~
~
~
~
~
~
~
~
~
~
~

11,24 All
```

The *rules* file

- The remaining discussion assumes C/C++ software that uses standard GNU tools
- Read comments in the *rules* file and fill in appropriate code or tweak what is there
 - the proper ``configure`` invocation
 - the proper ``make`` invocation
 - the proper ``make install`` invocation
 - the proper ``make distclean`` invocation
- Delete unnecessary commands

The *rules* file

```
dh_testdir
$(MAKE)
touch $@

clean:
dh_testdir
dh_testroot
rm -f build-stamp
-$(MAKE) distclean
ifneq "$(wildcard /usr/share/misc/config.sub)" ""
cp -f /usr/share/misc/config.sub config.sub
endif
ifneq "$(wildcard /usr/share/misc/config.guess)" ""
cp -f /usr/share/misc/config.guess config.guess
endif
dh_clean

install: build
dh_testdir
dh_testroot
dh_clean -k
dh_installdirs
$(MAKE) DESTDIR=$(CURDIR)/debian/tuition-ledger install
```

Build the package

```
roberto@miami: ~/tuition-ledger/tuition-ledger-0.0.1
File Edit View Terminal Tabs Help
-> unmounting dev/pts filesystem
-> unmounting proc filesystem
Current time: Sat Sep 29 01:59:16 EDT 2007
pbuilder-time-stamp: 1191045556
-> cleaning the build env
    -> removing directory /var/lib/chroot/pbuilder-sid/build//20977 and its subd
irectories
  signfile /var/lib/chroot/pbuilder-sid/results/tuition-ledger_0.0.1-1.dsc B2B97B
B1

You need a passphrase to unlock the secret key for
user: "Roberto C. Sanchez <roberto@connexer.com>"
1024-bit DSA key, ID A7C89464, created 2006-07-15 (main key ID B2B97BB1)

  signfile /var/lib/chroot/pbuilder-sid/results/tuition-ledger_0.0.1-1_i386.chang
es B2B97BB1

You need a passphrase to unlock the secret key for
user: "Roberto C. Sanchez <roberto@connexer.com>"
1024-bit DSA key, ID A7C89464, created 2006-07-15 (main key ID B2B97BB1)

Successfully signed dsc and changes files
roberto@miami:~/tuition-ledger/tuition-ledger-0.0.1$
```

Check the results

- Hopefully the build succeeded
- Make sure *.diff.gz* is not bloated (~10k)
- Run ``piuparts`` to check install/remove
- Run ``lintian -viI <changes_file>``
- Run ``linda -vi <changes_file>``
- Run ``dpkg --info <deb_file(s)>``
- Run ``dpkg --contents <deb_file(s)>``

Install the package

- Use ``dpkg -i <deb_file(s)>`` to install
- Alternatively, place .debs in local repo
 - “*Setting up your own APT repository with upload support*” at debian-administration.org

Beyond the basics

- If you need to patch the upstream code, consider using dpatch
- Fix all the lintian and linda warnings
- Override bogus warnings
- Build more complex packages
- Use debconf to handle configuration data

Summary

- Building Debian packages can be fun and profitable
- Start small and work your way up
 - Don't try to package gcc or postgres at first
 - Avoid the more exotic options at first
- Make a habit of creating clean, policy-compliant packages from the start

Questions?