

Packaging for Syadmins

David Nalley

david@gnsa.us

david.nalley@fedoraproject.org

New presentation available at:

<http://ke4qqq.fedorapeople.org/M11.pdf>

The non-code portions of this presentation are licensed under a Creative Commons, Attribution, Share Alike 3.0, unported license.



define: package

OS-specific or distro-specific installable container
for software

Why should sysadmins care?

SO WHAT?

Why should sysadmins care?

Automation – packaging makes decisions ahead of time

Why should sysadmins care?

Automation – packaging makes decisions ahead of time

Anyone still doing:

```
tar -zxvf foo.tar.gz  
cd foo  
./configure  
make  
make install
```

Why should sysadmins care

Repeatability

Using packages means that you will get the same behavior and files in the same location on subsequent installs

Why should sysadmins care?

Documentation:

Packages are documentation – provided you keep the original work. Building a package is far better than maintaining installation instructions.

Why should sysadmins care?

Easier customization

Need to deploy a package with different compiler options?

Need to patch the source for some cool new option?

Why should sysadmins care?

Security:

No compilers on hosts

Running your own package repository means you can ensure that every host with a given package will get updates.

Packaging for F/LOSS

Sysadmins should package F/LOSS for:
fame
peer review
experience

Packaging is really a snapshot of being a sysadmin

Automating
Configuration
Documenting

Doing things right the first time

Packaging we'll cover

RPM
DEB
Solaris

Examples

Learning by reading a book/website takes days
or weeks

Hopefully learning by seeing examples from
someone else will be faster

Longum iter est per praecepta, breve et efficax
per exempla!

RPM

Red Hat Package Manager

Used by RHEL, SLES, Fedora, and a few others

Resources

Maximum RPM: <http://www.rpm.org/max-rpm/>

Build Service at openSuSE:

http://en.opensuse.org/Build_Service

IBM DeveloperWorks – good generic docs

<http://docs.fedoraproject.org/drafts/rpm-guide-en/>

http://www.vim.org/scripts/script.php?script_id=98

Setting up your environment

Fedora: `yum install fedora-packager`

SLES/openSuSE: `yast -i osc`

RHEL: `up2date install rpmbuild`

Setting up your environment

`/usr/lib/rpm/macros`

```
mkdir -p ~/rpmbuild/SRPMS ~rpmbuild/RPMS \  
~/rpmbuild/SPECS ~rpmbuild/SOURCE \  
~/rpmbuild/BUILD ~rpmbuild/BUILDROOT
```

spec files – heart of rpm building

spec files are the instruction manual for the build process.

Tools

rpmbuild

rpmlint

mock

koji

Novell's Open Build Service

Contents of a spec file

headers – information about the package

scripts – the real work

files – identifying ownership and permissions

changelog – who changed what when

spec file framework

```
Name:  
Version:  
Release:          1%{?dist}  
Summary:  
  
Group:  
License:  
URL:  
Source0:  
BuildRoot:        %(mktemp -ud %[_tmppath]/%{name}-%{version}-%{release}-  
XXXXXXXX)  
  
BuildRequires:  
Requires:  
  
%description
```

spec file framework

```
%prep
%setup -q

%build
%configure
make %{?_smp_mflags}

%install
rm -rf $RPM_BUILD_ROOT
make install DESTDIR=$RPM_BUILD_ROOT

%clean
rm -rf $RPM_BUILD_ROOT

%files
%defattr(-,root,root,-)
%doc

%changelog
```

Building the spec file

```
%{!?python_sitelib: %define python_sitelib %(%{__python} -c "from
distutils.sysconfig import get_python_lib; print get_python_lib()")}
Name:                zapplet
Version:             0.1
Release:             4%{dist}
Summary:            Zenoss Tray Applet
Group:              Applications/Communications
License:            GPLv2+
URL:                http://zapplet.sourceforge.net
Source0:            http://downloads.sourceforge.net/zapplet/zapplet-%
{version}.tar.gz
Source1:            http://ke4qqq.fedorapeople.org/zapplet/README.fedora
BuildRoot:          %{_tmppath}/%{name}-%{version}-%{release}-root-%(
{__id_u} -n)
BuildRequires:     python python-setuptools-devel desktop-file-utils
Requires:          pygtk2
BuildArch:          noarch

%description
Zapplet is a tray applet for monitoring aspects of Zenoss from the
desktop.
```

Building the spec file

```
%prep
%setup -q -n zapplet-%{version}
cp %{SOURCE1} ./README.fedora
%build
CFLAGS="%{optflags}" %{__python} setup.py build

%install
rm -rf %{buildroot}
%{__python} setup.py install -O1 --skip-build --root %{buildroot}

desktop-file-install --add-category="Network" --dir=%{buildroot}%
_{datadir}/applications zapplet.desktop
rm -rf %{buildroot}%{_sysconfdir}/xdg/autostart/zapplet.desktop
```


Building the spec file

```
%clean
rm -rf %{buildroot}

%files
%defattr(-,root,root,-)
%{python_sitelib}/*
%{_bindir}/%{name}
%{_datadir}/applications/zapplet.desktop
%doc README.fedora
%changelog
* Mon Jan 19 2009 David Nalley <david@gnsa.us> 0.1-4
- Made changes suggested by ianweller to fix cvs issues
* Sun Dec 28 2008 David Nalley <david@gnsa.us> 0.1-3
- Made changes suggested by Brian Pepple
* Sat Dec 27 2008 David Nalley <david@gnsa.us> 0.1-2
- Making corrections suggested by Fabian Affolter
* Sun Dec 14 2008 David Nalley <david@gnsa.us> 0.1-1
- Initial packaging efforts
```

Building

```
rpmbuild -ba /path/to/zapplet.spec
```

rpmlint and other checks

```
rpmlint /path/to/zapplet.spec  
rpmlint /path/to/zapplet-0.1-4.src.rpm  
rpmlint /path/to/zapplet-0.1-4.noarch.rpm
```

And you are going to install and test it first, right?

Packaging for Debian

Resources

Debian New Maintainers' Guide -
<http://www.debian.org/doc/maint-guide/>
Debian Policy Manual -
<http://www.debian.org/doc/debian-policy/>

Setting up your environment

```
apt-get install dhmake
```

A directory to hold your builds

```
mkdir ~/debian
```

Debianization

Like brainwashing, only different.

```
dh_make -e email@server.com -f ../source.tar.gz
```


Debianization

Answer type of package you are creating – all the simple stuff should answer 's' to.

dh_make will create a copy of the source as 'source.orig.tar.gz'

Debianization

only run `dh_make` once in a directory.

First installation – file list gathering

To figure out a list of files the package owns and to create the first installation – install to a temp directory (or buildroot) `~/debian/packagename`

Might involve some editing of Makefile or if really 'hardcoded' you might end up editing source to match Debian Policy

control file

generated by 'debianization'

Contains information used by package
management tools

Copyright file

More important if you are releasing your
changes.

Format isn't specified by Debian Policy

changelog

```
foo (0.1.1-57) unstale; urgency=low
 * 57th try at debian package
 * changed source to avoid hardcoded paths
```

rules file

build process instructions, the spec file of debs

rules file

```
1 #!/usr/bin/make -f
2 # -*- makefile -*-
3 # Sample debian/rules that uses debhelper.
4 # This file was originally written by Joey Hess and Craig Small.
5 # As a special exception, when this file is copied by dh-make into a
6 # dh-make output file, you may use that output file without restriction.
7 # This special exception was added by Craig Small in version 0.37 of dh-make.
8 # Uncomment this to turn on verbose mode.
9 #export DH_VERBOSE=1
10 configure: configure-stamp
11 configure-stamp:
12     dh_testdir
13     # Add here commands to configure the package.
14     touch configure-stamp
15 build: build-stamp
16 build-stamp: configure-stamp
17     dh_testdir
18     # Add here commands to compile the package.
19     $(MAKE)
20     #docbook-to-man debian/testpack.sgml > testpack.1
21     touch $@
22 clean:
23     dh_testdir
24     dh_testroot
25     rm -f build-stamp configure-stamp
26     # Add here commands to clean up after the build process.
27     $(MAKE) clean
28     dh_clean
```



```
29 install: build
30     dh_testdir
31     dh_testroot
32     dh_clean -k
33     dh_installdirs
34     # Add here commands to install the package into debian/testpack.
35     $(MAKE) DESTDIR=$(CURDIR)/debian/testpack install
36 # Build architecture-independent files here.
37 binary-indep: build install
38 # We have nothing to do by default.
39 # Build architecture-dependent files here.
```

```
40 binary-arch: build install
41     dh_testdir
42     dh_testroot
43     dh_installchangelogs
44     dh_installdocs
45     dh_installexamples
46 #    dh_install
47 #    dh_installmenu
48 #    dh_installdebconf
49 #    dh_installogrotate
50 #    dh_installemacsen
51 #    dh_installpam
52 #    dh_installmime
53 #    dh_python
54 #    dh_installinit
55 #    dh_installcron
56 #    dh_installinfo
57     dh_installman
58     dh_link
59     dh_strip
60     dh_compress
61     dh_fixperms
62 #    dh_perl
63 #    dh_makeshlibs
64     dh_installdeb
65     dh_shlibdeps
66     dh_gencontrol
67     dh_md5sums
68     dh_builddeb
69 binary: binary-indep binary-arch
70 .PHONY: build clean binary-indep binary-arch binary install configure
```

conffiles.ex

List used to stop installations/upgrades from munging configuration files

Building

`dpkg-buildpackage -rfakeroot`

Solaris Packaging

A bit more primitive, but also simpler

File list

Generate the file list by:

```
./configure -prefix=/tmp/dir  
make && make install
```

```
find /tmp/dir -print > /tmp/filelist
```

Clean the file list up

File has all of the leading /tmp/dir that need to be cleaned up

```
cat /tmp/filelist | sed s#/tmp/dir##g \  
> /tmp/clean.filelist
```

(or s/>\tmp\dir//g <--- Far less readable)

Creating Package Conf Files

```
cat /tmp/clean.filelist | pkgproto > /tmp/Prototype
```

```
d none /usr/local 0755 root root
d none /usr/local/bin 0755 root root
d none /usr/local/man 0755 root root
d none /usr/local/man/man1 0755 root root
d none /usr/local/lib 0755 root root
d none /usr/local/etc 0755 root root
d none /usr/local/info 0755 root root
d none /usr/local/share 0755 root root
f none /usr/local/bin/bison 0755 root root
f none /usr/local/man/man1/bison.1 0644 root root
```


Tweaking Package Conf Files

```
i pkginfo  
i checkinstall
```

There is also preinstall and postinstall

pkginfo files

```
PKG="zapplet"  
NAME="Zapplet 0.1"  
VERSION="0.1"  
ARCH="i86pc"  
CLASSES="none"  
CATEGORY="utility"  
VENDOR="Zenoss"  
PSTAMP="20090917"  
EMAIL="david@gnsa.us"  
ISTATES="S s 1 2 3"  
RSTATES="S s 1 2 3"  
BASEDIR="/"
```

pkgmk -o -r / -d /tmp -f Prototype

-o overwrite

-r root directory

-d where to build the package

-f name of file list to use

Tar

```
tar -cf - Zapplet | gzip -9 -c > Zapplet-0.1-1.x86.tar.gz
```

Questions and contacts reminders

david.nalley@fedoraproject.org
ke4qqq on irc.freenode.net