

Packaging training

3. rpm

rpm

RPM are used in CentOS, Suse, Fedora...

```
rpm -ivh mypackage.rpm  
yum install ...
```

Where to start

Tutorial at:

http://fedoraproject.org/wiki/How_to_create_an_RPM_package#Preparing_your_system

<http://wiki.centos.org/HowTos/SetupRpmBuildEnvironment>

http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html-single/RPM_Guide/index.html

Dependencies

```
$sudo yum install rpm-build
```

Env setup

In the home dir, create the structure:

```
$mkdir -p ~/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
```

```
# Create a .rpmmacros to specify build dir location
```

```
$echo '%_topdir %(echo $HOME/rpmbuild)' > ~/.rpmmacros
```

Get upstream code

```
$cd rpmbuild/SOURCES
```

```
$git clone https://github.com/osallou/training-packaging-1.git training-1.0
```

```
$tar cvfz training-1.0.tar.gz training-1.0
```

rpmbuild dirs

SOURCES: contains upstream code archive and patches

SPEC: spec file used to build the package

RPMS: binary package

SRPMS: source package

The SPEC file

Let's create a training.spec

File specifies how to build the package, which files should be in package etc...

<http://rpm5.org/docs/api/specfile.html>

step by step: source desc

Name: training

Version: 1.0

Release: 1%{?dist}

Summary: bla bla

Group: Applications/Productivity

License: GPLv2+

URL: <https://xxx>

Url does not need to be valid, only basename of Source will be used to determine the name of the archive to search for in SOURCES, in our case it will search for training-1.0.tar.gz

Source: <https://github.com/osallou/training-packaging-2/archive/training-%{version}.tar.gz>

#BuildArch: noarch for architecture independent packages

Autoreq: 0

#BuildRequires: gcc, <= add packages required for building

%description

bla bla

step by step: packages

Now we define the packages to create, all %XX attributes are followed by the package name, because we have here a # multiple binary package. No need to do so for a single binary package.

We do not specify training-devel but devel, all packages will be named from the package name defined before

%package devel

Summary: Headers and libs required for development

Group: Development/Libraries

Requires: training-bin

%description devel

Contains headers and libraries required for development

In this example we will package binary AND shared library together

%package bin

Group: Applications/Productivity

Summary: bla

%description bin

bla bla

step by step: build

%prep

Execute setup macro, we specify here the name of the #
directory once .tar.gz is uncompressed

%setup -q -n %{name}-%{version}

%build

Execute what is needed to compile the software

%configure

make

%install

Several macro/variable are available, here we refer to the build
directory

#[https://fedoraproject.org/wiki/Packaging:RPMMacros?](https://fedoraproject.org/wiki/Packaging:RPMMacros?rd=Packaging/RPMMacros)
rd=Packaging/RPMMacros

rm -rf \$RPM_BUILD_ROOT

Recreate rpm build root

%__install -d -m 0755 \$RPM_BUILD_ROOT

make DESTDIR=\$RPM_BUILD_ROOT install

remove unwanted files

rm -f \$RPM_BUILD_ROOT/%{_libdir}/*.la

step by step: define files

Post installation instructions (optional)

%post bin

ldconfig

Post uninstallation instructions

(optional)

%postun bin

ldconfig

Dispatch files in packages

All files in RPM_BUILD_ROOT must be specified, else they must be removed first

%files devel

%defattr(-,root,root)

/usr/include/squizz

%{_libdir}/*.a

%{_libdir}/*.so

%files bin

%defattr(-,root,root)

%{_bindir}/squizz

%doc /usr/share/doc/squizz/alifmt.html

%doc /usr/share/doc/squizz/seqfmt.html

/usr/share/doc/squizz

/usr/share/man/man1/squizz.1.gz

/usr/share/man/man5/alifmt.5.gz

/usr/share/man/man5/seqfmt.5.gz

%{_libdir}/*.so.*

%defattr : default files/dir ownership, per file modification is possible with “%attr(0644, root, root) /etc/myfile.conf”

%{_libdir}: see available macros, this one is the library directory set automatically in configure step by the system (usr/lib)

Configuration files: %config(noreplace) /etc/myfile.conf

step by step: changelog

The required list of modifications

%changelog

* Mon Feb 18 2013 Olivier Sallou <olivier.sallou@irisa.fr> - 1.0-1

- Fedora packaging

Let's build the package

```
$cd rpmbuild
```

```
$rpmbuild -ba SPEC/training.spec
```

```
.....
```

```
Checking for unpackaged file(s): /usr/lib/rpm/check-files /home/vagrant/rpmbuild/BUILDROOT/training-1.0-1.el6.x86_64
```

```
Wrote: /home/vagrant/rpmbuild/SRPMS/training-1.0-1.el6.src.rpm
```

```
Wrote: /home/vagrant/rpmbuild/RPMS/x86_64/training-devel-1.0-1.el6.x86_64.rpm
```

```
Wrote: /home/vagrant/rpmbuild/RPMS/x86_64/training-bin-1.0-1.el6.x86_64.rpm
```

```
Wrote: /home/vagrant/rpmbuild/RPMS/x86_64/training-debuginfo-1.0-1.el6.x86_64.rpm
```

SRPMS/training-1.0-1.el6.src.rpm: package for source code

RPMS/x86_64/training-devel-1.0-1.el6.x86_64.rpm, RPMS/x86_64/training-bin-1.0-1.el6.x86_64.rpm: our binary packages for x86_64 architecture

Inside package

Let's check the content

```
[vagrant@localhost rpmbuild]$ rpm -qlp /home/vagrant/rpmbuild/RPMS/x86_64/training-bin-1.0-1.el6.x86_64.rpm  
/usr/bin/squizz  
/usr/lib64/libbioali.so.o  
/usr/lib64/libbioali.so.o.o.o  
/usr/lib64/libbioseq.so.o  
/usr/lib64/libbioseq.so.o.o.o  
/usr/share/doc/squizz  
/usr/share/doc/squizz/alifmt.html  
/usr/share/doc/squizz/seqfmt.html  
/usr/share/man/man1/squizz.1.gz  
/usr/share/man/man5/alifmt.5.gz  
/usr/share/man/man5/seqfmt.5.gz
```

Check package: rpmlint

```
[vagrant@localhost rpmbuild]$ rpmlint /home/vagrant/rpmbuild/RPMS/x86_64/training-bin-1.0-1.el6.x86_64.rpm
training-bin.x86_64: I: enchant-dictionary-not-found en_US
training-bin.x86_64: W: invalid-url URL: https://projets.pasteur.fr/projects/show/mobyle HTTP Error 404: Not Found
training-bin.x86_64: E: binary-or-shlib-defines-rpath /usr/bin/squizz ['/usr/lib64']
training-bin.x86_64: W: shared-lib-calls-exit /usr/lib64/libbioali.so.0.0.0 exit@GLIBC_2.2.5
training-bin.x86_64: W: shared-lib-calls-exit /usr/lib64/libbioseq.so.0.0.0 exit@GLIBC_2.2.5
training-bin.x86_64: W: one-line-command-in-%post ldconfig
training-bin.x86_64: W: one-line-command-in-%postun ldconfig
1 packages and 0 specfiles checked; 1 errors, 5 warnings.
```


Patches

To create a patch:

```
$cd path_to_myfile.c  
$cp myfile.c myfile.c.orig  
# Edit myfile.c  
$cd back_to_source_root  
$diff -u path_to_myfile.c/myfile.c.orig path_to_myfile.c/myfile.c > path_to_rpmbuild/SOURCES/mypatch.patch
```

Using patches in SPEC file:

```
Patch0: mypatch.patch  
Patch1: otherpatch.path
```

```
%prep  
%setup -q -n %{name}-%{version}  
%patcho -p0  
....
```

Test install/removal

Always test your package!

```
$ sudo rpm -ivh /home/vagrant/rpmbuild/RPMS/x86_64/training-bin-1.0-1.el6.x86_64.rpm  
$ sudo yum remove training-bin
```

Solution

Full spec file, patches.. are available at
https://build.opensuse.org/package/view_file/science:mobyle/squizz/squizz.spec?expand=1