Building RPMS
HEP SYSMAN, RAL

Steve Traylen

s.m.traylen@rl.ac.uk

2nd July 2004

Overview

- About RPM.
- A simple package.
- SPEC file
- System variables.
- Patching an RPM.
- RPM dependencies.
- Simple RPM of a script.
- Pre and post install Scripts.
- New features at RPM.
- Conclusions.

About RPM

- RPM = Redhat Package Manager.
- 8 of the top 10 distributions use rpms (exceptions slackware, debian and gentoo).
- RPM ported to Solaris, Irix, Digital, AIX. (Restriction is Berkeley db)
- Current version is version 4.2.
- RPM v1 released Oct 1996.

A simple package.

- Indent is installed like this.

-
  ```
  $ tar xzf indent.2.2.6-tar.gz
  $ cd indent-2.2.6
  $ ./configure
  $ make
  # make install
  ```

Building an RPM.

- Redhat sets up default locations in /usr/src/redhat.
- Required files
  /usr/src/redhat/SPECS/indent.spec
  /usr/src/redhat/SOURCES/indent-2.2.6.tar.gz
- Build command (-ba = build all).
  `# rpmbuild –ba indent.spec` (In rpm > 4.1 , rpm -> rpmbuild !!!)
- Creates
  /usr/src/redhat/RPMS/i386/indent-2.2.6-1.i386.rpm
  /usr/src/redhat/SRPMS/indent-2.2.6-1.i386.rpm
- The src rpm contains the tar balls and the spec file.

Spec File

- Spec files describe everything.
- They contain:
  - Preamble: Information about the software and final package.
  - %description: Description of the software.
  - %prep stage: Unpacks the software.
  - %build stage: Builds the software.
  - %install stage: Installs the software.
  - %files stage: List of files to be contained in a package.

Spec Preamble and Description

- Summary: GNU indent
  Name: indent
  Version: 2.2.6
  Release: 2
  Source0: %{name}-%{version}.tar.gz
  License: GPL
  Group: Development/Tools
  BuildRoot: %{_tmppath}/%{name}-root
  %description
  The GNU indent program reformats C code to any of a variety of formatting standards, or you can define your own.
- Variables, come from the user in the spec file, eg %{name} or the system eg %{_tmppath}.

%prep stage

- Starts with a %prep in the spec file.
- Want to untar indent-2.2.6.tar.gz into /usr/src/redhat/BUILD/.

  ```
  $ cd /usr/src/redhat/BUILD/indent-2.2.6
  ```

  `$ chown –R root:root` and `chmod –R a+rX,g-w,o-w`
- Within the SPEC file use this or the shorthand version with the %setup macro.

  ```
  %prep
  %setup
  ```
- Alternative %setups
  - %setup –n indent   used to name the directory to call in to.
  - %setup –d indent   mkdir and change into directory before untarring.

%build stage

- Starts with a %build in the SPEC file.
- Already in the correct directory.
- Build to be located on the system.
- Spec file requires

```
%build
./configure –prefix=%{_prefix}
make
```

- There are macros for common ones, eg %configure

%install stage

- Want to install in our special rpm build root located at
  ```
  BuildRoot: %{_tmppath}/%{name}-root
  ```
- This is a different root to what we configured for.
- Spec file contains
  ```
  %build
  rm -rf %{buildroot}
  make DESTDIR=${buildroot} install
  ```
- Software must support this fakeroot installation, if not just copy it.

%files stage

- Is a list of files in the final rpm and is maintained manually.
- It is relative to the rpm build root.
- In the spec file.

```
%files
%defattr(644,root,root)
%attr (755,root,root) %{_bindir}/indent
%{_mandir}/man1/indent.1
%doc README INSTALL AUTHORS
```

- Doc files without a path will be grabbed from the rpm build root and put in /usr/share/doc/indent-2.2.6

Final indent SPEC file.

```
Summary: GNU indent
Name: indent
Version: 2.2.6
Release: 2
Source0: %{name}-%{version}.tar.gz
License: GPL
Group: Development/Tools
BuildRoot: %{_tmppath}/%{name}-root
%description
The GNU indent program reformats C code to any of a variety of formatting standards, or
you can define your own.
%prep
%setup
%build
./configure –prefix=%{_prefix}
make
%install
rm -rf %{buildroot}
make DESTDIR=%{buildroot} install
%files
%defattr(644,root,root)
%attr (755,root,root) %{_bindir}/indent
%{_mandir}/man1/indent.1
%doc README INSTALL AUTHORS
```

System Variables

- Defined in
  /usr/lib/rpm/macros
  /usr/lib/rpm/i686-redhat-linux/macros
  ~/.rpmacros
- Display them all with
  $ rpm –showrc
- Eg
  RPM_PACKAGE_RELEASE="%{release}“
  %{?buildroot:RPM_BUILD_ROOT="%{buildroot}“
  export RPM_BUILD_ROOT}
  -14: _prefix   /usr
  -14: _rpmdir   %{_topdir}/RPMS

Overriding System Values

- This is done in ~/.rpmacros.
- A good example.
  - %_topdir   /home/csf/traylens/redhat
  - Redefines /usr/src/redhat allowing development of packages not as root.
- Another example for ~root/.rpmacros
  - %_repackage_all_erasures 1

Patching an RPM

- Install the src rpm

  $ rpm –Uvh openssh-3.4p2-1.src.rpm
- Uncompress the source ready for building.

  $ rpmbuild –bp ~/redhat/SPECS/openssh.spec
- This has executed the %prep stages only of the SPEC FILE.
- We now have the untared.

  ~/redhat/BUILD/openssh-3.4p2

Modify the Source

- First change to ~/redhat/BUILD since all patching is done is this directory.
- Preserve the original

```
$ cp –r openssh-3.4p2 openssh-3.4p2.orig
```

- Modify files in openssh-3.4p2 with whatever.
- Create a patch file.

```
$ diff –uNr openssh-3.4p2.orig \
              openssh-3.4p2  \
              > ../SOURCE/my-openssh.patch
```

Add the Patch to SPEC File

- Add the patch to the preamble.

```
Release: 3
Source0: %{name}-%{version}.tar.gz
Patch0: my-openssh.patch
License: GPL
```

- Apply the patch in the %prep stage.

```
%prep
%setup
%patch0
%build
```

- Build the rpm

```
$ rpmbuild –ba ~/redhat/SPECS/openssh.spec
```

Adding Dependencies

- The preamble can have requirements and pre requisites added to it.
  Require: gcc > 2.3
  Require: smtpserver
  PreReq: chkconfig
- The preamble can have things that it provides.
  Provides: smtpserver
- Lots of dependencies are calculated automatically, eg libc, /bin/bash, /bin/perl. More and more with every new distribution.

CLRC

RPM example of a script.

- The heartbeat-0.2.tar.gz file contains
  heartbeat-0.2/README
  heartbeat-0.2/heartbeat
  heartbeat-0.2/heartbeat-cron
  heartbeat-0.2/heartbeat-logrotate
- A simple script to call sysreq and heartbeat at intervals from cron.
- A small logfile must also be rotated.

Heartbeat SPEC file.

- Version: 0.2
  Summary: Heartbeat periodically to sure.
  Name: heartbeat
  Release: 2
  Copyright: BSD
  Group: Utilities/System
  Source: %{name}-%{version}.tar.gz
  Vendor: RAL
  Requires: sysreq
  BuildRoot: %{_tmppath}/%{name}-%{version}-buildroot
  Packager: Steve Traylen s.m.traylen@rl.ac.uk
  %description
  Heartbeat to sure

Heartbeat SPEC File(2)

```
%prep
%setup
%build  (Nothing Needed Here)
%install
install -d %{buildroot}/var/log/heartbeat
install –D -m755  heartbeat \
     %{buildroot}/usr/sbin/heartbeat
install –D -m644  heartbeat-cron \
      %{buildroot}/etc/cron.d/heartbeat
install –D -m644  heartbeat-logrotate \
     %{buildroot}/etc/logrotate.d/heartbeat
```

Heartbeat SPEC File (3)

- %files

```
%defattr(-,root,root)
%attr(0755,root,root) /usr/sbin/heartbeat
%attr(0655,root,root) /var/log/heartbeat
%config  /etc/cron.d/heartbeat
%config  /etc/logrotate.d/heartbeat
%doc     README
%changelog
* Fri Aug 02 2002 Steve Traylen s.m.traylen@rl.ac.uk
Changed heartbeat period to 5 minutes.
```

Pre and Post Scripts

- Within the SPEC file pre and post scripts can be defined.
  %pre , Run before the package is installed.
  %post , Run after the package is installed.
  %preun, Run before the package is removed.
  %postun, Run after the package is removed.
- During an upgrade the new package is installed before the old package is removed!!
- For all scripts $1 is set to the number of that packages installed at that time.

Pre/Post Scripts Example

```
• %post
  if [ “$1” = 1 ] ;then                    # We must be installing.
       /sbin/chkconfig –add httpd
       /sbin/chkconfig httpd on
  else                                     # We must be upradin g.
       /sbin/chkconfig –reset httpd
       /sbin/chkconfig httpd condrestart
  fi
  %postun
  if [ “$1” = 0 ] ; then                   # We must be removing.
      /sbin/service httpd stop
     /sbin/chkconfig –delete httpd
  fi
```

New Features in RPM

- RPM now supports globbing now.
  - # rpm -qa | grep kernel-
    is now replaced with
    # rpm -q 'kernel-*' .
  - (Don't let the shell globing get there first).

New Features in RPM

- RPM now supports rollback though it is undocumented.

```
# rpm -e –repackage samba-client
# rpm -Uvh –repackage kernel-2.4.20.i386.rpm
```

- Packages are placed in /var/spool/repackage.
- Restore with

```
# rpm -Uvh --rollback '2 hours ago'
Rollback packages (+1/-0) to Wed Jun 30 12:07:27
2004 (0x40e29eef):
```

Conclusions

- To make an rpm is pretty easy once you have done one.
- The oddest bit is the magic contained within the the %setup and %patch macros.
- The best method is to grab a similar src rpm and modify the spec file to your needs.

Resources

- http://www.rpm.org/
- http://www-106.ibm.com/developerworks/library/l-rpm1/
- http://www-106.ibm.com/developerworks/library/l-rpm2/
- http://www-106.ibm.com/developerworks/library/l-rpm3/
- http://www.distrowatch.com/
- http://www.rpm.org/max-rpm/
- http://www.linuxjournal.com/article.php?sid=7034