

Packaging: RPM

Chen Gui

`gui.chen@intel.com`

May 09, 2013

- The lead — to identify an RPM package file (0xedabedb)
- The signature — to verify an RPM package file
- The header — to define an RPM package file
- The archive — to comprise an RPM package file

Handling RPM File

- Query RPM file

Example

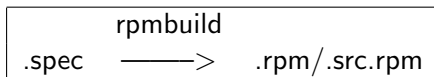
```
$ rpm -qp -qf '%{VCS}' bash.rpm
```

- Unpack RPM file

Example

```
$ rpm2cpio bash.rpm |cpio -idv
```

RPM Packaging



- Comments
Human-readable notes ignored by RPM
- Tags
Define data
- Scripts
Contain commands to be executed at specific times
- Macros
A method of executing multiple commands easily
- The %files list
A list of files to be included in the package
- Directives
Used in the %files list to direct RPM to handle certain files in a specific way
- Conditionals
Permit operating system- or architecture-specific preprocessing of the spec file

- N-V-R-E — Name/Version/Release/Epoch
- Description — %description (rpm -qi)
- Summary — restricted to one line
- License — define the copyright terms
- Source and Patch
 - Source/Source1/Source2
 - Patch/Patch1/Patch2

- Provides
list virtual package names that this package provides.
- Requires
A comma-separate list of packages that are required when the program is installed
- Conflicts
state what other packages cannot be installed simultaneously to this one.
- Obsoletes
remove another named package(s) when this package is installed.

Scriptlet

- `%prep`
Script commands to "prepare" the program (e.g. to uncompress it) so that it will be ready for building.
- `%build`
Script commands to "build" the program (e.g. to compile it) and get it ready for installing.
- `%install`
Script commands to "install" the program.
- `%clean`
Instructions to clean out the build root.
- `%pre` and `%post`
before (`%pre`) or after (`%post`) a package is installed
- `%preun` and `%postun` before (`%preun`) or after (`%postun`) a package is uninstalled

Option	Usage
-ba	Build all, both a binary and source RPM
-bb	Build a binary RPM
-bc	Build (compile) the program but do not make the full RPM, by executing the build commands through the %build section and stopping
-bp	Prepare for building a binary RPM, by executing the build commands through the %prep section and stopping
-bi	Execute the build commands through the %install section and stop
-bl	Check the listing of files for the RPM
-bs	Build a source RPM only

- BuildRequires

A comma-separated list of packages required for building (compiling) the program.

- Macros

Example

```
$ rpmbuild -ba --define '_topdir /mnt' bash.spec
```

- Subpackages

%package (-n)? subpackage-name

- rpmlint

config file: rpmlintrc

- Maximum RPM
<http://www.rpm.org/max-rpm/index.html>
- RPM Guide
http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/index.html
- Packagers Guide
http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/Packagers_Guide/index.html
- Packaging Guidelines
<http://fedoraproject.org/wiki/Packaging:Guidelines>