Details on RPM spec Sections

coursera.org/learn/linux-tools-for-developers/supplement/CJAp9/details-on-rpm-spec-sections

The Header Section

The Header section is a list assignments, of the format:

Attribute: Value

An example showing the attributes usually used is provided below:

Summary: A great application!

Name: my_app

Version: 1.0

Release: 2

License: GPLv2

Group: Applications/Text

Source: ftp://ftp.myserver.com/pub/my_app/my_app-1.0.tgz

URL: https://www.myserver.com/my_app/index.html

Vendor: The Best Software Company

Packager: A genius <genius@myserver.com>

Patch0: my_app-1.0.patch0

Patch1: my_app-1.0.patch1

BuildRoot: /var/tmp/%{name}-buildroot

The **Name**, **Version**, and **Release** values are used to construct the names of the source and binary RPM files. Most other values are fairly self-explanatory, and can be chosen at will by the packager.

Note that the **License** field was called **Copyright** in earlier versions of RPM.

The description Section

The **description** section is completely free form. To see the description of an existing package, try **rpm -qi** package. An example description:

```
%description

This program enables the user to do everything that could

possibly be useful under Linux.
```

The prep Section

The **prep** section of the header must take the **tar** file of the source code, unpack it, and then apply any patch files needed. Often, this section is extremely short. An example **prep** section would be:

```
%prep

rm -rf $RPM_BUILD_DIR/my_app-1.0

tar zxvf $RPM_SOURCE_DIR/my_app-1.0.tar.gz

patch -p1 < my_app-1.0.patch1

patch -p2 < my_app-1.0.patch2
```

Modern versions of **rpmbuild** do not use such detailed **prep** sections. They can simply do:

```
%prep

%setup -q

%autosetup
```

The changelog Section

The **changelog** section lists a history of changes to the package. For example:

```
%changelog

* Tue Feb 29 2001 Mr Somebody <sbody@foo.com>

- important fix: cleaned up a bug that trashed the filesystem.


* Tue Feb 15 2001 Another Somebody <abody@foo.com>

- made minor changes to the initialization routine.
```

The build Section

The **build** section contains the commands needed to build your program. If your **prep** section did its work correctly, this should be one command:

```
%build

make
```

The install Section

The **install** section copies files to their final place in the filesystem. This is run after building the binaries, but not when the package is installed. When the binary is built, RPM will automatically package up the files that were put in place by the install section, then during installation, it will copy them back.

Usually, developers put an **install** target into makefiles that do this copying, which makes the install section of the **spec** file simple:

```
%install

make install
```

The files Section

The **files** section of the **spec** file lists all of the files that the **install** section copied into place. Files may be prefixed by an adjective that changes how RPM treats these files. The possible adjectives are:

Adjective	Description
%doc	Marks documentation from the source package to be included in a binary install. Multiple documents can be listed on the same line, or on separate lines. If no path for a documentation file is specified, then it will be put in a directory named **/usr/share/doc/package-version-build**.
%config	Mark configuration files. When you uninstall a package, any configuration files which have been changed will be saved with a **.rpmsave** extension.
%dir	By default, if you include a directory in the files section, the directory and all files it contains are put in as part of installation. Using **%dir** means that only the directory, and not the files that it contains, will be copied when the package is installed.

Adjective	Description
%attr	Set attributes for the file. An example would be **%attr(666,root,root)**, where the three fields are mode, owner, and group (just like the output of **ls -l**). Defaults are gotten by using a **-** in a place.
%defattr	The same **as %attr**, but you do not add a file to it. Instead, all files listed after the **%defattr** will automatically get the new attributes (unless they have their own **%attr** directive).

An example **files** section would be:

```
%files

%doc README

/usr/bin/my_app

%config /usr/bin/my_app_configure

/usr/man/man/my_app.1

%attr(600,-,root) /usr/lib/my_app/my_data
```