– LPI 101 –

Use Red Hat Package Management [5]

(Linux Professional Institute Certification)

a


```
   .~.      by: Andrew Eager
   /V\          andy@linuxivr.com
  // \\
  @._.@
```



```
$Id: gl1.102.6.slides.tex,v 1.4 2003/05/30 05:09:04 waratah Exp $
```

Use Red Hat Package Management

Objective

Candidates should be able to perform package management under Linux distributions that use RPMs for package distribution. This objective includes being able to install, re-install, upgrade, and remove packages, as well as obtain status and version information on packages. This objective also includes obtaining package information such as version, status, dependencies, integrity, and signatures. Candidates should be able to determine what files a package provides, as well as find which package a specific file comes from.

Use Red Hat Package Management

Key files, terms, and utilities

```
/etc/rpmrc
/usr/lib/rpm/*
```

Use Red Hat Package Management

Resources of interest

:

RPM - RedHat Package Manager

- RPM works with RedHat, SuSE & Mandrake (among others) and can do the following:

RPM - RedHat Package Manager

- RPM works with RedHat, SuSE & Mandrake (among others) and can do the following:
  - Build an RPM package

RPM - RedHat Package Manager

- RPM works with RedHat, SuSE & Mandrake (among others) and can do the following:
  - Build an RPM package
  - Install an RPM package

RPM - RedHat Package Manager

- RPM works with RedHat, SuSE & Mandrake (among others) and can do the following:
  - Build an RPM package
  - Install an RPM package
  - Update an already installed RPM package

RPM - RedHat Package Manager

- RPM works with RedHat, SuSE & Mandrake (among others) and can do the following:
  - Build an RPM package
  - Install an RPM package
  - Update an already installed RPM package
  - Query an RPM package

RPM - RedHat Package Manager

- RPM works with RedHat, SuSE & Mandrake (among others) and can do the following:
  - Build an RPM package
  - Install an RPM package
  - Update an already installed RPM package
  - Query an RPM package
  - Erase an RPM package

RPM - RedHat Package Manager

- RPM works with RedHat, SuSE & Mandrake (among others) and can do the following:
  - Build an RPM package
  - Install an RPM package
  - Update an already installed RPM package
  - Query an RPM package
  - Erase an RPM package
  - Verify an RPM package.

RPM Packages

- RPM package files consist of a single compressed file much like a tarball. Package files can be sourced from:
  - Local media (hard disk, cdrom etc)
  - An ftp site
  - An http site

RPM Packages

- RPM package files consist of a single compressed file much like a tarball. Package files can be sourced from:
  - Local media (hard disk, cdrom etc)
  - An ftp site
  - An http site
- Packages sourced from local media are specified using just their filename. For example:
  - acroread-4.05-1.i686.rpm

RPM Packages

- RPM package files consist of a single compressed file much like a tarball. Package files can be sourced from:
  - Local media (hard disk, cdrom etc)
  - An ftp site
  - An http site
- Packages sourced from local media are specified using just their filename. For example:
  - acroread-4.05-1.i686.rpm
- Packages sourced from ftp or http sites are specified using the following syntax:
  - ftp://USER:PASSWORD@HOST:PORT/path/to/package.rpm

RPM Filenames

RPM Filenames use a standard naming scheme:

`package`-version-patch.arch.rpm

package - The name of the package

version - The version number

patch - patch number of this package

arch - The architecture this package is for (i386, i586, i686, alpha, sparc)

Example:

`kernel`-2.4.9-21.i686.rpm

RPM Filenames

RPM Filenames use a standard naming scheme:

`package-version-patch.arch.rpm`

package - The name of the package

version - The version number

patch - patch number of this package

arch - The architecture this package is for (i386, i586, i686, alpha, sparc)

Example:

`kernel-2.4.9-21.i686.rpm`

RPM Filenames

RPM Filenames use a standard naming scheme:

`package-version-patch.arch.rpm`

package - The name of the package

version - The version number

patch - patch number of this package

arch - The architecture this package is for (i386, i586, i686, alpha, sparc)

Example:

`kernel-2.4.9-21.i686.rpm`

RPM Filenames

RPM Filenames use a standard naming scheme:

`package-version-patch.arch.rpm`

package - The name of the package

version - The version number

patch - patch number of this package

arch - The architecture this package is for (i386, i586, i686, alpha, sparc)

Example:

`kernel-2.4.9-21.i686.rpm`

RPM Operating Modes

RPM operations are split up into 4 major modes:

RPM Operating Modes

RPM operations are split up into 4 major modes:

- Querying & Verifying

RPM Operating Modes

RPM operations are split up into 4 major modes:

- Querying & Verifying
- Installing, Upgrading or Removing

RPM Operating Modes

RPM operations are split up into 4 major modes:

- Querying & Verifying
- Installing, Upgrading or Removing
- Building Packages

RPM Operating Modes

RPM operations are split up into 4 major modes:

- Querying & Verifying
- Installing, Upgrading or Removing
- Building Packages
- RPM database administration functions

Verifying package Integrity

Having downloaded an rpm from the Internet, the very first thing you want to do is verify its integrity.

You do this with the `-K` or `--checksig` option to rpm:

```
# rpm -K kernel-2.4.9-31.i586.rpm
kernel-2.4.9-31.i586.rpm: md5 gpg OK
```

NOTE: Some packages use PGP to check integrity while others use GnuPG.

Installing, Upgrading & Removing

Understanding the RPM terminology in relation to Installing, upgrading & removing rpm packages is essential:

Installing, Upgrading & Removing

Understanding the RPM terminology in relation to Installing, upgrading & removing rpm packages is essential:

- Install - Install a package. Good for Kernels

Installing, Upgrading & Removing

Understanding the RPM terminology in relation to Installing, upgrading & removing rpm packages is essential:

- Install - Install a package. Good for Kernels
- Upgrade - Upgrade a package if it’s installed, otherwise install the package

Installing, Upgrading & Removing

Understanding the RPM terminology in relation to Installing, upgrading & removing rpm packages is essential:

- Install - Install a package. Good for Kernels
- Upgrade - Upgrade a package if it’s installed, otherwise install the package
- Freshen - Upgrade a package only if it’s already installed.

Installing, Upgrading & Removing

Understanding the RPM terminology in relation to Installing, upgrading & removing rpm packages is essential:

- Install - Install a package. Good for Kernels
- Upgrade - Upgrade a package if it’s installed, otherwise install the package
- Freshen - Upgrade a package only if it’s already installed.
- Erase - Remove a package.

Installing, Upgrading & Removing - Options

The table below summarises the various options used for package installation, freshening and removal.

Mode	Short option	Long option
Install	-i	`--install`
Upgrade	-U	`--upgrade`
Freshen	-F	`--freshen`
Erase	-e	`--erase`

Using RPM on the command line

Generally you use rpm in one of the following modes:

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`
- `rpm -U [install-options] package_file ...`

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`
- `rpm -U [install-options] package_file ...`
- `rpm -F [install-options] package_file ...`

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`
- `rpm -U [install-options] package_file ...`
- `rpm -F [install-options] package_file ...`
- `rpm -e [erase-options] package_name ...`

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`
- `rpm -U [install-options] package_file ...`
- `rpm -F [install-options] package_file ...`
- `rpm -e [erase-options] package_name ...`

Commonly used options are:

`-v` Verbose

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`
- `rpm -U [install-options] package_file ...`
- `rpm -F [install-options] package_file ...`
- `rpm -e [erase-options] package_name ...`

Commonly used options are:

`-v` Verbose

`-h` print progress hash marks

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`
- `rpm -U [install-options] package_file ...`
- `rpm -F [install-options] package_file ...`
- `rpm -e [erase-options] package_name ...`

Commonly used options are:

`-v` Verbose

`-h` print progress hash marks

`--force` Force RPM to overwrite existing packages or files

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`
- `rpm -U [install-options] package_file ...`
- `rpm -F [install-options] package_file ...`
- `rpm -e [erase-options] package_name ...`

Commonly used options are:

`-v` Verbose

`-h` print progress hash marks

`--force` Force RPM to overwrite existing packages or files

`--nodeps` Bypass dependency checking

Using RPM on the command line

Generally you use rpm in one of the following modes:

- `rpm -i [install options] package_file ...`
- `rpm -U [install-options] package_file ...`
- `rpm -F [install-options] package_file ...`
- `rpm -e [erase-options] package_name ...`

Commonly used options are:

`-v` Verbose

`-h` print progress hash marks

`--force` Force RPM to overwrite existing packages or files

`--nodeps` Bypass dependency checking

`--replacefiles` Overwrite files owned by other packages

Example - install option

Install kernel-2.4.18-4 without removing the existing kernel:

```
# rpm -q kernel
kernel-2.4.9-21
# rpm -ivh kernel-2.4.18-4*.rpm
Preparing...                ########################### [100%]
   1:kernel-2.4.18-4        ########################### [100%]
# rpm -q kernel
kernel-2.4.18-4
kernel-2.4.9-21
#
```

Example - Freshen option

Upgrade the existing kernel to kernel-2.4.18-4.

```
# rpm -q kernel
kernel-2.4.9-21
# rpm -Uvh kernel-2.4.18-4*.rpm
Preparing...                ########################### [100%]
   1:kernel-2.4.18-4        ########################### [100%]
# rpm -q kernel
kernel-2.4.18-4
#
```

Example - Upgrade option:

What the upgrade option does will depend on whether or not the package is currently installed. If installed, it will perform a freshen, otherwise it will perform an install:

Example - Upgrade option:

What the upgrade option does will depend on whether or not the package is currently installed. If installed, it will perform a freshen, otherwise it will perform an install:

```
# rpm -q gocr
package gocr is not installed
```

Example - Upgrade option:

What the upgrade option does will depend on whether or not the package is currently installed. If installed, it will perform a freshen, otherwise it will perform an install:

```
# rpm -q gocr
package gocr is not installed
# rpm -Uvh gocr-0.3.4-1.i386.rpm
Preparing...                ########################### [100%]
   1:gocr-0.3.4-1.rpm       ########################### [100%]
```

Example - Upgrade option:

What the upgrade option does will depend on whether or not the package is currently installed. If installed, it will perform a freshen, otherwise it will perform an install:

```
# rpm -q gocr
package gocr is not installed
# rpm -Uvh gocr-0.3.4-1.i386.rpm
Preparing...                ########################### [100%]
   1:gocr-0.3.4-1.rpm       ########################### [100%]

# rpm -q gocr
gocr-0.3.4-1
```

Example - Upgrade option:

What the upgrade option does will depend on whether or not the package is currently installed. If installed, it will perform a freshen, otherwise it will perform an install:

```
# rpm -q gocr
package gocr is not installed
# rpm -Uvh gocr-0.3.4-1.i386.rpm
Preparing...                ########################### [100%]
   1:gocr-0.3.4-1.rpm       ########################### [100%]

# rpm -q gocr
gocr-0.3.4-1
# rpm -Uvh gocr-0.3.6-1.i386.rpm
Preparing...                ########################### [100%]
   1:gocr-0.3.6-1.rpm       ########################### [100%]
```

Example - Upgrade option:

What the upgrade option does will depend on whether or not the package is currently installed. If installed, it will perform a freshen, otherwise it will perform an install:

```
# rpm -q gocr
package gocr is not installed
# rpm -Uvh gocr-0.3.4-1.i386.rpm
Preparing...                ########################### [100%]
   1:gocr-0.3.4-1.rpm       ########################### [100%]

# rpm -q gocr
gocr-0.3.4-1
# rpm -Uvh gocr-0.3.6-1.i386.rpm
Preparing...                ########################### [100%]
   1:gocr-0.3.6-1.rpm       ########################### [100%]
# rpm -q gocr
gocr-0.3.6-1
```

Example - Erase option

Remove the package gocr from the system

```
# rpm -q gocr
gocr-0.3.6-1
# rpm -e gocr
# rpm -q gocr
#
```

Querying Packages

RPM can be used to query a package (either installed or not).

```
rpm -q|--query [select-options] [query-options]
```

Querying Packages

RPM can be used to query a package (either installed or not).

```
rpm -q|--query [select-options] [query-options]
```

Select options : Choose what it is you want to query

- **`-a`** Query all installed packages.
- **`-f`** Query package owning FILE.
- **`-p`** Query an (uninstalled) package file

Querying Packages

RPM can be used to query a package (either installed or not).

```
rpm -q|--query [select-options] [query-options]
```

Select options : Choose what it is you want to query

- **`-a`** Query all installed packages.
- **`-f`** Query package owning FILE.
- **`-p`** Query an (uninstalled) package file

Query options : Choose what it is you want to see from the query:

- **`-i`** Show all information about the package
- **`-l`** Show what files are contained in the package
- **`-R`** List packages on which this package depends

Query - Example

Give a list of all packages with kern in their name:

```
$  rpm -qa | grep kern ↩
kernelcfg-0.5-5
glibc-kernheaders-2.4-7.14
kernel-2.4.9-21
kernel-source-2.4.18-4
```

Query - Example

Show a list of all files in kernel-2.4.9-21

```
$ rpm -ql kernel-2.4.9-21 ↩
/boot/System.map-2.4.9-21
/boot/module-info-2.4.9-21
/boot/vmlinuz-2.4.9-21
....
```

Query - Example

Show package which owns /bin/ls:

```
$ rpm -qf /bin/ls ↩
$ fileutils-4.1-10 ↩
```

Show complete information about the fileutils package:

```
$ rpm -qi fileutils ↩

Name        : fileutils                     Relocations: (not relocateable)
Version     : 4.1                                Vendor: Red Hat, Inc.
Release     : 10                             Build Date: Mon 25 Mar 2002 12:23:22 PM E
Install date: Fri 24 May 2002 02:18:08 PM EST      Build Host: daffy.perf.redhat.com
Group       : Applications/File             Source RPM: fileutils-4.1-10.src.rpm
Size        : 1679468                          License: GPL
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Summary     : The GNU versions of common file management utilities.
Description :
The fileutils package includes a number of GNU versions of common and
popular file management utilities. Fileutils includes the following
tools: chgrp (changes a file’s group ownership), chown (changes a
file’s ownership), chmod (changes a file’s permissions), cp (copies
files), dd (copies and converts files), df (shows a filesystem’s disk
usage), dir (gives a brief directory listing), dircolors (the setup
program for the color version of the ls command), du (shows disk
usage), install (copies files and sets permissions), ln (creates file
links), ls (lists directory contents), mkdir (creates directories),
mkfifo (creates FIFOs or named pipes), mknod (creates special files),
mv (renames files), rm (removes/deletes files), rmdir (removes empty
```

Verifying package files

This option to rpm is used to verify the files installed on the system with those from the rpm package file. This is not to be confused with the *integrity* of the package file.

The following table lists the characteristics verified:

5 - The MD5 checksum

S - The file size

L - Symbolic link

T - Modification time

D - Device major & minor number

U - User owner

G - Group owner

M - Permission and/or file type

Example - Verify package

Verify the setup package against the originally installed version.

```
$ rpm -V setup ↩
S.5....T c /etc/bashrc
S.5....T c /etc/csh.cshrc
S.5....T c /etc/csh.login
S.5....T c /etc/host.conf
S.5....T c /etc/printcap
S.5....T c /etc/profile
..?..... c /etc/securetty
.M...... c /etc/shadow
```

The End