INSTALLING AND UPDATING SOFTWARE PACAKGES

Objectives

- Register a system to your Red Hat account and assign it entitlements for software updates and support services using Red Hat Subscription Management.
- Find, install and update software packages using the **yum** command.
- Enable and disable use of Red Hat or third-party Yum repositories by a server.

REGISTERING SYSTEMS FOR RED HAT SUPPORT

1. RED HAT SUBSCRIPTION MANAGEMENT

Red Hat Subscription Management provides tools that can be used to entitle machines to product subscriptions, allowing administrators to get updates to software packages and track information about support contracts and subscriptions used by the systems. Standard tools such as PackageKit and **yum** can obtain software packages and updates through a content distribution network provided by Red Hat.

There are four basic tasks performed with Red Hat Subscription Management tools:

- **Register** a system to associate that system to a Red Hat account. This allows Subscription Manager to uniquely inventory the system. When no longer in use, a system may be unregistered.
- **Subscribe** a system to entitle it to updates for selected Red Hat products. Subscriptions have specific levels of support, expiration dates, and default repositories. The tools can be used to either auto-attach or select a specific entitlement. As needs change, subscriptions may be removed.
- **Enable** repositories to provide software packages. Multiple repositories are enabled by default with each subscription, but other repositories such as updates or source code can be enabled or disabled as needed.
- **Review** and track entitlements that are available or consumed. Subscription information can be viewed locally on a specific system or, for an account, in either the Red Hat Customer Portal Subscriptions page or the Subscription Asset Manager (SAM)

2. REGISTRATION FROM THE COMMAND LINE

Use the subscription-manager(8) to register a system without using a graphical environment. The subscription-manager command can automatically attach a system to the best-matched compatible subscriptions for the system.

- Register a system to a Red Hat account:

  [user@host ~]$ subscription-manager register --username=yourusername --password=yourpassword

- View available subscriptions:

  [user@host ~]$ subscription-manager list --available | less

- Auto-attach a subscription:

  [user@host ~]$ subscription-manager attach --auto
- Alternatively, attach a subscription from a specific pool from the list of available subscriptions:

  [user@host ~]$ subscription-manager attach --pool=poolID
- View consumed subscriptions:

  [user@host ~]$ subscription-manager list --consumed
- Unregister a system:

  [user@host ~]$ subscription-manager unregister

RPM SOFTWARE PACKAGES

1. SOFTWARE PACAKGES AND RPM

The RPM Package Manager, originally developed by Red Hat, provides a standard way to package software for distribution. Managing software in the form of RPM packages is much simpler than working with software that has simply been extracted into a file system from an archive. It lets administrators track which files were installed by the software package and which ones need to be removed if it is uninstalled, and check to ensure that supporting packages are present when it is installed. Information about installed packages is stored in a local RPM database on each system. All software provided by Red Hat for Red Hat Enterprise Linux is provided as an RPM package.

RPM package files names consist of four elements (plus the **.rpm** suffix): name-version-release.architecture:

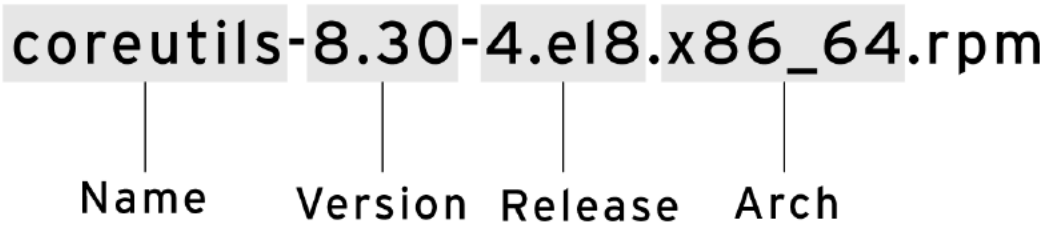

Figure 14.3: RPM file name elements

- NAME is one or more words describing the contents (coreutils).
- VERSION is the version number of the original software (8.30).
- RELEASE is the release number of the package based on that version, and is set by the packager, who might not be the original software developer (4.el8).
- ARCH is the processor architecture the package was compiled to run on. **noarch** indicates that this package's contents are not architecture-specific (as opposed to **x86_64** for 64-bit, **aarch64** for 64-bit ARM, and so on).

Only the package name is required for installing packages from repositories. If multiple versions exist, the package with the higher version number is installed. If multiple releases of a single version exist, the package with the higher release number is installed.

Each RPM package is a special archive made up of three components:

- The files installed by the package.

- Information about the package (metadata), such as the name, version, release, and arch; a summary and description of the package; whether it requires other packages to be installed; licensing; a package change log; and other details.
- Scripts that may run when this package is installed, updated, or removed, or are triggered when other packages are installed, updated, or removed.

Updating Software with RPM Packages

Red Hat generates a complete RPM package to update software. An administrator installing that package gets only the most recent version of the package. To update software, RPM removes the older version of the package and installs the new version. Updates usually retain configuration files, but the packager of the new version defines the exact behavior.

2. EXAMING RPM PACKAGES

The **rpm** utility is a low-level tool that can get information about the contents of package files and installed packages. By default, it gets information from the local database of installed packages. However, you can use the -p option to specify that you want to get information about a downloaded package file. You might want to do this in order to inspect the contents of the package file before installing it.

The general form of a query is:

- **rpm -q [select-options] [query-options]**

RPM queries: General information about installed packages

- rpm -qa: List all installed packages
- rpm -qf FILENAME: Find out what package provides FILENAME

user@host ~]$ rpm -qf /etc/yum.repos.d

redhat-release-8.0-0.39.el8.x86_64

RPM queries: Information about specific packages

- rpm -q: List what version of the package is currently installed

[user@host ~]$ rpm -q yum

yum-4.0.9.2-4.el8.noarch

- rpm -qi: Get detailed information about the package
- rpm -ql: List the files installed by the package

[user@host ~]$ rpm -ql yum

/etc/yum.conf

/etc/yum/pluginconf.d

/etc/yum/protected.d

/etc/yum/vars

/usr/bin/yum

/usr/share/man/man1/yum-aliases.1.gz

/usr/share/man/man5/yum.conf.5.gz

/usr/share/man/man8/yum-shell.8.gz

/usr/share/man/man8/yum.8.gz

3. INSTALLING RPM PACKAGES

The **rpm** command can also be used to install an RPM package that you have downloaded to your local directory.

[root@host ~]# rpm -ivh wonderwidgets-1.0-4.x86_64.rpm

Verifying... ## [100%]

Preparing... ## [100%]

Updating / installing...

1:wonderwidgets-1.0-4 ## [100%]

[root@host ~]#

SUMMARY OF RPM QUERY COMMANDS

Installed packages can be queried directly with the **rpm** command. Add the **-p** option to query a package file before installation.

COMMAND	TASK
rpm -qa	List all RPM packages currently installed
rpm -q NAME	Display the version of NAME installed on the system
rpm -qi NAME	Display detailed information about a package
rpm -ql NAME	List all files included in a package
rpm -qc NAME	List configuration files included in a package
rpm -qd NAME	List documentation files included in a package
rpm -q --changelog NAME	Show a short summary of the reason for a new package release
rpm -q --scripts NAME	Display the shell scripts run on package installation, upgrade, or removal

INSTALLING AND UPDATING SOFTWARE PACKAGES WITH YUM

1. MANAGING SOFTWARE PACAKGES WITH YUM

The low-level **rpm** command can be used to install packages, but it is not designed to work with package repositories or resolve dependencies from multiple sources automatically.

Yum is designed to be a better system for managing RPM-based software installation and updates. The **yum** command allows you to install, update, remove, and get information about software packages and their dependencies. You can get a history of transactions performed and work with multiple Red Hat and third-party software repositories.

Finding Software with Yum

- **yum help** displays usage information.
- **yum list** displays installed and available packages.

  [user@host]$ yum list 'http*'

  Lists all available packages of http
- **yum search KEYWORD** lists packages by keywords found in the name and summary fields only.

  To search for packages that have "web server" in their name, summary, and description fields, use search all:

  [user@host ~]$ yum search all 'web server'
- **yum info PACKAGENAME** returns detailed information about a package, including disk space needed for installation.

  To get information on the Apache HTTP Server:

  [user@host ~]$ yum info httpd
- **yum provides PATHNAME** displays packages that match the path name specified

  To find packages that provide the /var/www/html directory, use:

  [user@host ~]$ yum provides /var/www/html

2. Installing and removing software with yum

- **yum install PACKAGENAME** obtains and installs a software package, including any dependencies.

  [user@host ~]$ yum install httpd

Dependencies resolved.

==========

Package Arch Version Repository Size

==========

Installing:

httpd x86_64 2.4.37-7.module... rhel8-appstream 1.4 M

Installing dependencies:

apr x86_64 1.6.3-8.el8 rhel8-appstream 125 k

apr-util x86_64 1.6.1-6.el8 rhel8-appstream 105 k

...output omitted...

Transaction Summary

- **yum update PACKAGENAME** obtains and installs a newer version of the specified package, including any dependencies. Generally the process tries to preserve configuration files in place, but in some cases, they may be renamed if the packager thinks the old one will not work after the update. With no PACKAGENAME specified, it installs all relevant updates.

  [user@host ~]$ sudo yum update

  Since a new kernel can only be tested by booting to that kernel, the package is specifically designed so that multiple versions may be installed at once. If the new kernel fails to boot, the old kernel is still available. Using yum update kernel will actually install the new kernel. The configuration files hold a list of packages to always install even if the administrator requests an update.
- **yum remove PACKAGENAME** removes an installed software package, including any supported packages.

  [user@host ~]$ sudo yum remove httpd

Installing and removing groups of software with yum

- **yum** also has the concept of groups, which are collections of related software installed together for a particular purpose. In Red Hat Enterprise Linux 8, there are two kinds of groups. Regular groups are collections of packages. Environment groups are collections of regular groups. The packages or groups provided by a group may be mandatory (they

must be installed if the group is installed), default (normally installed if the group is installed), or optional (not installed when the group is installed, unless specifically requested).

Like yum list, the yum group list command shows the names of installed and available groups.

[user@host ~]$ yum group list

Available Environment Groups:

Server with GUI

Minimal Install

Server

...output omitted...

Available Groups:

Container Management

.NET Core Development

- **yum group info** displays information about a group. It includes a list of mandatory, default, and optional package names.

  [user@host ~]$ yum group info "RPM Development Tools"

  Group: RPM Development Tools

  Description: These tools include core development tools such rpmbuild.

  Mandatory Packages:

  redhat-rpm-config

  rpm-build

  Default Packages:

  rpmdevtools

  Optional Packages:

  rpmlint

- **yum group install** installs a group that installs its mandatory and default packages and the packages they depend on.

  [user@host ~]$ sudo yum group install "RPM Development Tools"

  ...output omitted...

  Installing Groups:

  RPM Development Tools

Transaction Summary

==========

Install 64 Packages

Total download size: 21 M

Installed size: 62 M

Is this ok [y/N]: y

...output omitted...

SUMMARY OF YUM COMMANDS

Packages can be located, installed, updated, and removed by name or by package groups.

TASK:	COMMAND:
List installed and available packages by name	`yum list [NAME-PATTERN]`
List installed and available groups	`yum group list`
Search for a package by keyword	`yum search KEYWORD`
Show details of a package	`yum info PACKAGENAME`
Install a package	`yum install PACKAGENAME`
Install a package group	`yum group install GROUPNAME`
Update all packages	`yum update`
Remove a package	`yum remove PACKAGENAME`
Display transaction history	`yum history`