

RHCE BOOT CAMP

Package Management



redhat.®

CERTIFIED
E N G I N E E R

RPM

- Redhat Package Manager
- RPM's provide full software packaging features: pre-install scripts, post-install scripts, dependencies, meta information, and an installed software database to name a few.
- The RPM system maintains a database of all installed software on a machine - this is useful for tracking and updating reasons, as well as dependency verification and software management.

RPM

- rpm: The Redhat Package Manager tool. Provides interface to RPM system, performing queries, installs, upgrades, uninstalls and general database maintenance operations.
 - -i option: install the given package
 - -q option: query the database
 - -e option: erase the given package from the system

RPM QUERIES

- Below are just a few examples of the types of queries you can run against the RPM database.
 - **rpm -qa** Queries for the names of all installed rpms.
 - **rpm -qi** Queries the rpm database for package information.
 - **rpm -qf** Determines which rpm a file is associated with.
 - **rpm -ql** Queries the rpm database to determine which files are associated with a particular rpm.
- With any of these commands, you can add the **-p** option to run the command against a package before it is installed.

RPM INSTALLATION VERIFICATION

- In addition to storing information about where a package is installed, rpm also stores permissions, file sizes, md5sums, and ownership information. This information can be easily referenced to see if anything has been changed.
 - **rpm -Va** Verifies all installed packages.
 - **rpm -Vi <package>** Verifies given package.
- Rackspace Best Practice Example
 - `rpm -Va | grep ^..5`

RPM VERIFY OUTPUT

- **S** File Size differs
- **M** Mode differs (includes permissions and file type)
- **5** MD5 sum differs
- **D** Device major/minor number mismatch
- **L** readLink(2) path mismatch
- **U** User ownership differs
- **G** Group ownership differs
- **T** mTime differs
- **C** SELinux Context differs

EXTRACT RPM CONTENTS

- Use this technique to make a clean working copy of the files and directories that would be installed with a package.
 - `cd /temp/dir`
 - `rpm2cpio /path/to/package | cpio -i -d -m`
- This would allow you to:
 - Replace one corrupted file without un-installing and then re-installing a package
 - Compare original configuration files versus modified files in the running system to quickly locate changed lines, for example with the 'diff' utility

YUM

- yum: Yellowdog Updater Modified
 - Supports package installation over the network through repositories.
 - RPM backend
 - Simple interface

REPOSITORIES

- Repositories of packages must be listed in files in the `/etc/yum.repos.d` directory with names ending in `.repo` and having a format like:
 - `[label-for-repo]`
 - `name = descriptive text`
 - `baseurl = protocol://path/to/directory/of/packages`
- Access to the Red Hat Network, including any Satellite Servers, is implemented through a plugin to the yum tool itself and not as a repository definition in the above format.

LAB

1. Connect to <http://server1.example.com> and read the information there.
2. Download the OpenOffice archive from `server1` and choose an appropriate location to extract all its RPMs
3. Install the `createrepo` package and use it to turn your collection of OpenOffice packages into a yum repository
4. Add that repository to your local yum configuration
5. Using yum, install the “openoffice.org3-writer” package, and/or any others from your new repository


```
slideshow.end();
```