

# Using the Red Hat Package Manager

29 January 2002 - v1.0.3

updated for RPM v4

Dag Wieërs

dag@wieers.com



## Overview

---

- Introduction to package management
  - Using RPM to install packages p15
  - Using RPM to remove packages p35
  - Using RPM to upgrade/freshen packages p43
  - Getting information about packages p51
  - Using RPM to verify installed packages p79
- RPM for developers
  - The basics of developing with RPM p119
  - A simple example spec file p125
  - Using RPM to build packages p139
  - Inside the Spec file p165

(We'll use Maximum RPM v1.1 as the reference manual)

- 2 -



## Why use packages ?

---

- Simplify system administration
- Install, upgrade and remove without problems
- Ability to add dependencies between software, libraries and tools
- Indexing and verifying installed systems
- Simplify software building and distribution

- 3 -



## RPM design goals

---

- Make it easy to get packages on and off the system
- Make it easy to verify a package was installed correctly
- Make it easy for the package builder
- Make it start with the original code
- Make it work on different computer architectures

- 4 -



## What is a package ?

- Package label (!= filename)
  - Software name logwatch
  - Software version 0.1.0
  - Package release dag.1
- Build date and build host
- Install date and total size of files installed
- Package license, vendor and packager
- Description, summary, group information and URL
- A signature to verify authenticity and integrity
- Changelog of the Spec file
- Per-file information
  - Location of file
  - File permissions and ownership
  - MD5 checksum

- 5 -



## Using RPM to install packages

- **rpm -i**
  - -h print "#" marks during install
  - -v display additional information
  - -vv display debugging information
  - --test
  
  - --nodeps do not check dependencies
  - --replacepks replace a package with a new copy
  - --replacefiles replace files owned by another pkg
  - --force ignore package and file conflicts
  - --noscripts do not execute install scripts
  - --prefix relocate package if possible
  
  - --root set alternate root
  - --dbpath use alternate RPM database

- 6 -



## Installation examples

```
[root@lune ~]# rpm -ihv /home/dag/autofs-3.1.5-5.i386.rpm
autofs      #####
```

```
[root@lune ~]# rpm -ivv ftp://ftp.easynet.be/...wertools/i386/iptraf*
D: counting packages to install
D: found 1 packages
D: looking for packages to download
D: logging into ftp.belnet.be as ftp, pw (username)
Retrieving ftp://ftp.easynet.be/...owertools/i386/iptraf-2.2.0-4.i386.rpm
D: ... as /var/tmp/rpm-xfer.ldArgM
...
```

```
[root@lune ~]# mount -t ext2 /dev/hdb2 /mnt/fw
[root@lune ~]# rpm -i --root /mnt/rh62 ipchains-1.3.9-17.i386.rpm
```

```
[root@lune ~]# rpm -i --prefix /mnt/rh62/usr ipchains-1.3.9-17.i386.rpm
```

- 7 -



## Using RPM to erase packages

- **rpm -e**
  - -vv display debugging information
  - --test perform erase tests only
  
  - --noscripts, --nodeps, --root, --dbpath
- **Configuration files will never be removed if changed !!**

- 8 -



## Using RPM to upgrade/freshen packages

- **rpm -U**
  - -h print "#" marks during upgrade
  - --test perform upgrade tests only
  - --oldpackage permits "upgrading" to an older pkg
  - --replacepks, --replacefiles, --force, --noscripts
- **Unchanged configuration files will be replaced**
- **Uses magic to determine when to replace**
- Can be used instead of rpm -i in all cases
- Use **rpm -F** to freshen files (i.e. upgrade only packages that are already installed)

- 9 -



## Getting information about packages

- **rpm -q**
  - -p query (not-installed) package file (parameter)
  - -f query package owning file (parameter)
  - -a query all installed packages
  - -i display summary package information
  - -l (-s) display list of files in package (w/ state)
  - -c display list of configuration files
  - -d display list of documentation files
  - --scripts display install and verify scripts
  - -v, -vv, --root, --dbpath

- 10 -



## Query examples

```
[dag@lune ~]$ rpm -qi logwrapper
Name       : logwrapper      Relocations: /usr
Version    : 0.1.0            Vendor: Dag Wieers <dag@wieers.com>
Release    : dag.1         Build Date: Thu 13 Dec 2001 09:06:44 PM CET
Install date: Fri 14 Dec 2001 ... Build Host: lune.dag.wieers.com
Group      : System/Tools   Source RPM: logwrapper-0.1.0-dag.4.src.rpm
Size       : 916           License: CPL
Packager   : Dag Wieers <dag@wieers.com>
URL        : http://dag.wieers.com/home-made/logwrapper/
Summary    : A wrapper script for logging all activity of a program
Description:
logwrapper is a tool that is able to log all activity of a program.
It will log the date, user and full command together with its output.
```

- 11 -



## Query examples

```
[dag@lune ~]$ rpm -qa
ld.so-1.9.5-13
vtun-2.3-1
diskcheck-3.1.1-2
...

[dag@lune ~]$ rpm -q logwatch
logwatch-0.1.0-1

[dag@lune ~]$ rpm -ql logwatch-0.1.0
/etc/logrotate.d/logwrapper
/etc/profile.d/logwrapper.sh
/usr/bin/logwrapper
/usr/share/doc/logwrapper-0.1.0
/usr/share/doc/logwrapper-0.1.0/README
/var/log/logwrapper
```

- 12 -



## Extra query options

- Extra options
  - --whatprovides query pkgs providing capability
  - --whatrequires query pkgs requiring capability
  
  - --provides display capabilities provided by pkg
  - --requires display capabilities required by pkg
  - --dump display all verifiable information
  
  - --queryformat
    - Queryformat tags:  
NAME, VERSION, RELEASE, SUMMARY, BUILDHOST,  
BUILDTIME, INSTALLTIME, SIZE, FILEMODES,  
FILENAMES, FILESIZES, REQUIRENAME, ...

Queryformat-tags are not case sensitive.



## Query examples

```
[root@lune ~]# rpm -qf `which logwrapper`  
logwrapper-0.1.0-1
```

```
[root@lune ~]# rpm -qf /lib/libss.so.2  
e2fsprogs-1.18-16
```

```
[root@lune ~]# rpm -qf --queryformat "%{summary}\n" /lib/libss.so.2  
Utilities for managing the second extended (ext2) file system.
```

```
[root@lune ~]# rpm -q --whatprovides webserver  
apache-1.3.14-2
```

```
[root@lune ~]# rpm -q --whatrequires apache  
piranha-gui-0.4.14-1  
mod_ssl-2.7.1-2
```



## Query examples

```
[root@lune ~]# rpm -qc --whatprovides webserver  
/etc/httpd/conf/access.conf  
/etc/httpd/conf/httpd.conf  
/etc/httpd/conf/srm.conf  
/etc/logrotate.d/apache  
/etc/rc.d/init.d/httpd  
/var/www/html/index.html
```

```
[root@lune ~]# rpm -q -f /lib/libc.so.6 --provides  
ldconfig  
ANSI_X3.110.so  
...  
libutil.so.1(GLIBC_2.0)  
glibc = 2.1.94-3
```



## Using RPM to verify packages

- rpm -V
  - -v display additional information
  - -vv display debugging information
  
  - -p verify against (not-installed) pkg (parameter)
  - -f verify package owning file (parameter)
  - -a verify all installed packages
  
  - --noscripts, --nodeps, --nofiles, --root, --dbpath
  -
- rpm -K
  - -v display additional information
  - -vv display debugging information
  
  - --nopgp do not verify PGP signatures



## Verify example (and mnemonics)

---

```
[dag@lune ~]$ rpm -V setup
```

```
S.5....T c /etc/host.conf
```

```
SM5....T c /etc/profile
```

```
S.?....T c /etc/securetty
```

- S -- file size differs
- M -- mode differs (includes permissions and file type)
- 5 -- MD5 checksum differs
- D -- device major/minor number mismatch
- L -- symbolic link mismatch
- U -- user ownership differs
- G -- group ownership differs
- T -- file modification time differs
- ? -- unreadable file
- 
- c -- configuration file

