RPM RPM package tool

START HERE

`rpm -qa`
List installed packages

`rpm -qlp <file.rpm>`
Look inside a package file

`rpm -qi <pkg>`
Package details

`sudo rpm -Uvh <file.rpm>`
Install or upgrade a file

`rpm -ql <pkg>`
Files a package installed

`rpm -V <pkg>`
Check files against the db

`rpm -qf <path>`
Which package owns this file

`sudo rpm -e <pkg>`
Remove a package

.rpm file rpm -qp reads it → `rpm -U` → **RPM database** rpm -q reads it → `rpm -V` → **Files on disk** checked against db

CONCEPTS

rpm handles single package files and the local database. It does not download or resolve dependencies: use dnf for that.

Packages are named name-version-release.arch, e.g. tree-2.2.1-4.fc44.x86_64. Any leading part works as a query: tree or tree-2.2.1.

Add -p to any query to read a .rpm file instead of the installed database.

QUERY INSTALLED

`rpm -qa`
List all installed packages

`rpm -qa '<pattern>'`
List packages matching a glob
Quote the glob, e.g. 'kernel', so the shell leaves it alone.*

`rpm -q <pkg>`
Is it installed, and which version

`rpm -qi <pkg>`
Show package details

`rpm -ql <pkg>`
List files of a package

`rpm -qf <path>`
Which package owns a file

`rpm -qc <pkg>`
List its config files

`rpm -qd <pkg>`
List its documentation files

`rpm -qa --last`
Packages by install time, newest first

`rpm -q --changelog <pkg>`
Show the package changelog

`rpm -q --scripts <pkg>`
Show install and remove scriptlets

`rpm -q --dump <pkg>`
Files with size, mtime, digest, mode

`rpm -q --dupes`
Find packages installed twice

`rpm -q --json <pkg>`
Dump the header as JSON

DEPENDENCIES

`rpm -qR <pkg>`
What a package requires

`rpm -q --provides <pkg>`
What a package provides

DEPENDENCIES (cont.)

`rpm -q --whatprovides <cap>`
Which package provides a capability
Works for files and sonames too, e.g. 'libc.so.6()(64bit)'.

`rpm -q --whatrequires <cap>`
Which installed packages need it
Only direct requirements on that exact capability, not reverse dependencies in general.

`rpm -q --whatrecommends <cap>`
Which packages recommend it

`rpm -q --conflicts <pkg>`
What a package conflicts with

`rpm -q --obsoletes <pkg>`
What a package replaces

PACKAGE FILES

`rpm -qip <file.rpm>`
Details of a package file

`rpm -qlp <file.rpm>`
List files in a package file

`rpm -qRp <file.rpm>`
Requirements of a package file

`rpm -qp --scripts <file.rpm>`
Scriptlets it would run
Check before installing a package from an untrusted source.

`rpm2cpio <file.rpm> | cpio -idmv`
Extract files without installing
Unpacks into the current directory, keeping the paths.

`rpm2cpio <file.rpm> | cpio -t`
List the payload

`rpm2archive <file.rpm> | tar -tz`
Convert payload to tar.gz
Default pax format has no 4 GB per-file limit, unlike cpio.

INSTALL, UPGRADE, REMOVE

`sudo rpm -Uvh <file.rpm>`
Install or upgrade, with progress
-v verbose, -h hash marks. Removes the old version.

`sudo rpm -ivh <file.rpm>`
Install, keep other versions
Use -i for kernels so the old one stays bootable.

`sudo rpm -Fvh <file.rpm>`
Upgrade only if already installed

`sudo rpm -e <pkg>`
Remove a package

INSTALL, UPGRADE, REMOVE (cont.)

`sudo rpm -e --allmatches <pkg>`
Remove every installed version

`sudo rpm -Uvh --test <file.rpm>`
Dry run: would it install?

`sudo rpm --reinstall <file.rpm>`
Reinstall the same version

`sudo rpm -Uvh --oldpackage <file.rpm>`
Downgrade to an older version

`sudo rpm -Uvh --nodeps <file.rpm>`
Skip the dependency check
Leaves a broken system if the deps are really missing.

`sudo rpm -Uvh --replacefiles <file.rpm>`
Allow overwriting other packages' files

`sudo rpm -Uvh --noscripts <file.rpm>`
Do not run scriptlets

Prefer sudo dnf install ./<file.rpm>: it installs the local file and pulls in its dependencies.

VERIFY

`rpm -V <pkg>`
Check files against the database
No output means nothing changed.

`rpm -Va`
Verify every installed package
Slow; add --nodeps to skip dependency checks.

`rpm -Vf <path>`
Verify the package owning a file

`rpm -V --nomtime <pkg>`
Ignore timestamp changes

`sudo rpm --restore <pkg>`
Reset owner, mode, mtime and caps
Fixes metadata only; restore contents with rpm --reinstall.

rpm -V flags, e.g. S.5....T.

Char	Changed
`S`	size
`M`	mode or file type
`5`	digest (contents)
`D`	device number
`L`	symlink target
`U / G`	owner / group
`T`	mtime
`P`	capabilities
`. / ?`	passed / could not test
`c d g l r`	config, doc, ghost, license, readme file

SIGNATURES AND KEYS

`sudo rpmkeys --import <key.asc>`
Trust a signing key
rpm --import still works as an alias.

`rpmkeys -K <file.rpm>`
Check digests and signature
Prints 'digests signatures OK'; NOKEY means the key is not imported.

`rpmkeys -Kv <file.rpm>`
Show each signature and digest

`rpmkeys --list`
List imported keys

`sudo rpmkeys --delete <keyid>`
Remove an imported key

`rpm -qa 'gpg-pubkey*'`
Keys as pseudo-packages (rpmdb keyring)

QUERY FORMAT

`rpm -q --qf '%{NAME} %{VERSION}\n' <pkg>`
Print chosen tags
--qf is short for --queryformat. Add \n yourself.

`rpm -q --qf '[%{FILENAMES}\n]' <pkg>`
Array tags: one line per entry

`rpm --querytags`
List all tag names

DATABASE AND CONFIG

`sudo rpmdb --rebuilddb`
Rebuild the package database
Try this when queries hang or report a corrupt db; rpm --rebuilddb is an alias.

`rpm -E '%{_dbpath}'`
Expand a macro

RECIPES

Which package owns this command

```
rpm -qf "$(command -v <cmd>)"
# not installed yet? ask dnf
dnf provides '*/bin/<cmd>'
```

Biggest installed packages

```
rpm -qa --qf
'%{SIZE}\t%{NAME}\n' | sort -n
| tail
```

Look inside a download before installing

```
rpmkeys -K <file.rpm>
rpm -qip <file.rpm>
rpm -qp --scripts <file.rpm>
mkdir peek && cd peek &&
rpm2cpio ../<file.rpm> | cpio -
idmv
```

Find and repair changed files

```
rpm -Va --nodeps --nomtime
rpm -qf <path>
sudo rpm --reinstall <file.rpm>
```