

Usage: rpm [OPTION...]

Query options (with -q or --query):

```
-c, --configfiles      list all configuration files
-d, --docfiles         list all documentation files
--dump                dump basic file information
-l, --list             list files in package
--queryformat=QUERYFORMAT use the following query format
-s, --state            display the states of the listed files
-a, --all              query/verify all packages
-f, --file             query/verify package(s) owning file
-g, --group            query/verify package(s) in group
-p, --package         query/verify a package file
-W, --ftswalk         query/verify package(s) from TOP file tree
walk
--sourcepkgid        query/verify package(s) with source
package identifier
--pkgid               query/verify package(s) with package
identifier
--hdrid              query/verify package(s) with header
identifier
--fileid             query/verify package(s) with file
identifier
--specfile           query a spec file
--triggeredby        query the package(s) triggered by the
package
--whatrequires       query/verify the package(s) which require
a dependency
--whatprovides       query/verify the package(s) which provide
a dependency
--whatconflicts      query/verify the package(s) which conflict
with a dependency
--whatobsoletes      query/verify the package(s) which obsolete
a dependency
--nomanifest         do not process non-package files as
manifests
```

Verify options (with -V or --verify):

```
--nofdigests         don't verify file digests
--nofiles            don't verify files in package
--nodeps             don't verify package dependencies
--noscript           don't execute verify script(s)
-a, --all            query/verify all packages
-f, --file           query/verify package(s) owning file
-g, --group          query/verify package(s) in group
-p, --package        query/verify a package file
-W, --ftswalk        query/verify package(s) from TOP file tree
walk
--sourcepkgid        query/verify package(s) with source
package identifier
--pkgid               query/verify package(s) with package
identifier
--hdrid              query/verify package(s) with header
identifier
--fileid             query/verify package(s) with file
identifier
--specfile           query a spec file
--triggeredby        query the package(s) triggered by the
package
--whatrequires       query/verify the package(s) which require
a dependency
--whatprovides       query/verify the package(s) which provide
a dependency
--whatconflicts      query/verify the package(s) which conflict
with a dependency
--whatobsoletes      query/verify the package(s) which obsolete
a dependency
--nomanifest         do not process non-package files as
```

manifests

Dependency check/order options:

```
--[no]aid          Add packages to resolve dependencies
--[no]suggest      Do not suggest missing dependency
                  resolution(s)
```

File tree walk options (with --ftswalk):

```
--comfollow       FTS_COMFOLLOW: follow command line symlinks
--logical         FTS_LOGICAL: logical walk
--nochdir         FTS_NOCHDIR: don't change directories
--nostat         FTS_NOSTAT: don't get stat info
--physical       FTS_PHYSICAL: physical walk
--seedot         FTS_SEEDOT: return dot and dot-dot
--xdev           FTS_XDEV: don't cross devices
--whiteout       FTS_WHITEOUT: return whiteout information
```

Signature options:

```
--addsign         sign package(s) (identical to --resign)
-K, --checksig   verify package signature(s)
--delsign        delete package signatures
--import         import an armored public key
--resign         sign package(s) (identical to --addsign)
--nodigest       don't verify package digest(s)
--nosignature    don't verify package signature(s)
```

Database options:

```
--rebuilddb      rebuild database inverted lists from
                  installed package headers
```

Build options with [<specfile> | <tarball> | <source package>]:

```
-bp              build through %prep (unpack sources and
                  apply patches) from <specfile>
-bc              build through %build (%prep, then compile)
                  from <specfile>
-bi              build through %install (%prep, %build,
                  then install) from <specfile>
-bl             verify %files section from <specfile>
-ba             build source and binary packages from
                  <specfile>
-bb             build binary package only from <specfile>
-bs             build source package only from <specfile>
-bt             track versions of sources from <specfile>
-bf             fetch missing source and patch files
-tp             build through %prep (unpack sources and
                  apply patches) from <tarball>
-tc             build through %build (%prep, then compile)
                  from <tarball>
-ti             build through %install (%prep, %build,
                  then install) from <tarball>
-ta             build source and binary packages from
                  <tarball>
-tb             build binary package only from <tarball>
-ts             build source package only from <tarball>
--rebuild       build binary package from <source package>
--recompile     build through %install (%prep, %build,
                  then install) from <source package>
--clean         remove build tree when done
--nobuild       do not execute any stages of the build
--nodeps       do not verify build dependencies
--rmsource     remove sources when done
--rmspec       remove specfile when done
--short-circuit skip straight to specified stage (only for
                  c,i)
--target=CPU-VENDOR-OS  override target platform
```

Install/Upgrade/Erase options:

```
--allfiles          install all files, even configurations
                    which might otherwise be skipped
--excludedocs      do not install documentation
--justdb           update the database, but do not modify the
                    filesystem
--nofdigests       don't verify file digests
--noscripts        do not execute package scriptlet(s)
--notriggers       do not execute any scriptlet(s) triggered
                    by this package
--[no]repackage    save erased package files by repackaging
--test            don't install, but tell if it would work
                    or not
--allmatches       remove all packages which match <package>
                    (normally an error is generated if
                    <package> specified multiple packages)
--badreloc         relocate files in non-relocatable package
-e, --erase=<package>+
--excludepath=<path>
                    skip files with leading component <path>
--fileconflicts    detect file conflicts between packages
-F, --freshen=<packagefile>+
                    upgrade package(s) if already installed
-h, --hash         print hash marks as package installs (good
                    with -v)
--ignorearch       don't verify package architecture
--ignoreos         don't verify package operating system
--ignoresize       don't check disk space before installing
-i, --install      install package(s)
--nodeps           do not verify package dependencies
--noorder          do not reorder package installation to
                    satisfy dependencies
--oldpackage       upgrade to an old version of the package
                    (--force on upgrades does this
                    automatically)
--percent          print percentages as package installs
--prefix=<dir>     relocate the package to <dir>, if
                    relocatable
--relocate=<old>=<new>
                    relocate files from path <old> to <new>
--replacefiles     ignore file conflicts between packages
--replacepkgs     reinstall if the package is already present
-U, --upgrade=<packagefile>+
                    upgrade package(s)
```

Common options:

```
-D, --define='MACRO EXPR'    Define MACRO with value EXPR
--undefine='MACRO'          Undefine MACRO
-E, --eval='EXPR'           Print macro expansion of EXPR
--macros=<FILE:...>         Read <FILE:...> instead of default file(s)
--rpm lua=<FILE:...>        Read <FILE:...> instead of default RPM Lua
                             file(s)
--rpm popt=<FILE:...>       Read <FILE:...> instead of default POPT
                             file(s)
--target=CPU-VENDOR-OS     Specify target platform
--nodigest                 Don't verify package digest(s)
--nosignature              Don't verify package signature(s)
--querytags                Display known query tags
--showrc                   Display macro and configuration values
-r, --root=ROOT            Use ROOT as top level directory (default:
                             "/")
--quiet                     Provide less detailed output
-v, --verbose               Provide more detailed output
--version                  Print the version
--stats                     Display operation statistics
```

Options implemented via popt alias/exec:

```
--scripts                list install/erase scriptlets from
                             package(s)
--scripts:yaml            list install/erase scriptlets (YAML markup)
--danglinglink            display link needed, but not provided, by
```

```

packages
--orphandirs      display directories needed, but not
                  provided, by packages
--setperms        set permissions of files in a package
--setugids        set user/group ownership of files in a
                  package
--setfscontext    set selinux extended attributes file
                  context of files in a package
--conflicts       list capabilities this package conflicts
                  with
--obsoletes       list other packages removed by installing
                  this package
--provides        list capabilities that this package
                  provides
--requires        list capabilities required by package(s)
--suggests        list enhancements suggested by package(s)
--enhances        list enhancements suggested by package(s)
--info            list descriptive information from
                  package(s)
--changelog       list change logs for this package
--xml             list metadata in XML
--yaml           list metadata in YAML
--json           list metadata in JSON
--whatneeds       list packages that depend on args
--needswhat       list prerequisite packages of args
--deb:Packages    display Debian Packages file
--deb:Sources     display Debian Sources file
--deb:control     display binary debian/control file
--deb:preinst     display debian/preinst script
--deb:postinst    display debian/postinst script
--deb:prerm       display debian/prerm script
--deb:postrm      display debian/postrm script
--deb:md5sums     display debian/md5sums manifest
--yum:primary.xml display yum primary.xml markup
--yum:filelists.xml display yum filelists.xml markup
--yum:other.xml   display yum other.xml markup
--yum:primary.sqlite display yum primary.sqlite import
--yum:filelists.sqlite display yum filelists.sqlite import
--yum:other.sqlite display yum other.sqlite import
--wnh:primary.yaml display wnh primary.yaml markup
--wnh:filelists.yaml display wnh filelists.yaml markup
--wnh:other.yaml  display wnh other.yaml markup
--wnh:primary.mongo display wnh primary.mongo markup
--wnh:filelists.mongo display wnh filelists.mongo markup
--wnh:other.mongo display wnh other.mongo markup
--wdj:srpm        display wdj binary->source Cassandra markup
--wdj:srpm.mongo  display wdj binary->source Mongo markup
--nix:meta        display nix meta markup
--xiu:stdenv      display xiu stdenv markup
--triggers        list trigger scriptlets from package(s)
--last            list package(s) by install time, most
                  recent first
--dupes           list duplicated packages
--filesbypkg      list all files from each package
--fileclass       list file names with classes
--filecolor       list file names with colors
--filecontext     list file names with security context from
                  header
--fscontext       list file names with security context from
                  file system
--recontext       list file names with security context from
                  policy RE
--fileprovide     list file names with provides
--filerequire     list file names with requires
--vendorprovides  find package name that contains a provided
                  capability (needs rpmdb-vendor package
                  installed)
--vendorrequires  find package name that contains a required
                  capability (needs rpmdb-vendor package

```

installed)
--autorollback Failed transactions will be rolled back automatically
--buildpolicy=<policy> set buildroot <policy> (e.g. compress man pages)
--force force operation (short hand for --replacepkgs --replacefiles --oldpackage on installation)
--downgrade Allow an upgrade to replace a newer package with an older one.
--what-provides find package name that contains a provided capability
--what-requires find package name that contains a required capability
--norepackage Disable re-package of the files before erasing
--with=<option> enable configure <option> for build
--without=<option> disable configure <option> for build
--arch list package architecture

Help options:

?, --help Show this help message
--usage Display brief usage message