



Rocky Enterprise Linux 9.2 Manual Pages on command 'rpmbuild.8'

C:\>man rpmbuild.8

RPMBUILD(8) System Manager's Manual RPMBUILD(8)

NAME

rpmbuild - Build RPM Package(s)

SYNOPSIS

BUILDING PACKAGES:

rpmbuild {-ba|-bb|-bp|-bc|-bi|-bl|-bs} [rpmbuild-options] SPECFILE ...

rpmbuild {-ra|-rb|-rp|-rc|-ri|-rl|-rs} [rpmbuild-options] SOURCEPACKAGE ...

rpmbuild {-ta|-tb|-tp|-tc|-ti|-tl|-ts} [rpmbuild-options] TARBALL ...

rpmbuild {--rebuild|--recompile} SOURCEPKG ...

MISCELLANEOUS:

rpmbuild --showrc

rpmbuild-options

[--buildroot DIRECTORY] [--clean] [--nobuild]

[--rmsource] [--rmspec] [--short-circuit] [--build-in-place]

[--noprep] [--noclean] [--nocheck]

[--rpmfcdebug]

[--target PLATFORM]

[--with OPTION] [--without OPTION]

DESCRIPTION

rpmbuild is used to build both binary and source software packages. A package con?

sists of an archive of files and meta-data used to install and erase the archive

files. The meta-data includes helper scripts, file attributes, and descriptive in?

formation about the package. Packages come in two varieties: binary packages, used to encapsulate software to be installed, and source packages, containing the source code and recipe necessary to produce binary packages.

One of the following basic modes must be selected: Build Package, Build Package from Tarball, Recompile Package, Show Configuration.

GENERAL OPTIONS

These options can be used in all the different modes.

-?, --help

Print a longer usage message than normal.

--version

Print a single line containing the version number of rpm being used.

--quiet

Print as little as possible - normally only error messages will be displayed.

-v Print verbose information - normally routine progress messages will be displayed.

-vv Print lots of ugly debugging information.

--rpmfcdebug

Enables to debug dependencies generation.

--rcfile FILELIST

Each of the files in the colon separated FILELIST is read sequentially by rpm for configuration information. Only the first file in the list must exist, and tildes will be expanded to the value of \$HOME. The default FILELIST is /usr/lib/rpm/rpmrc:/usr/lib/rpm/redhat/rpmrc:/etc/rpmrc:~/rpmrc.

--pipe CMD

Pipes the output of rpm to the command CMD.

--dbpath DIRECTORY

Use the database in DIRECTORY rather than the default path /var/lib/rpm

--root DIRECTORY

Use the file system tree rooted at DIRECTORY for all operations. Note that this means the database within DIRECTORY will be used for dependency checks and any scriptlet(s) (e.g. %post if installing, or %prep if building, a

package) will be run after a chroot(2) to DIRECTORY.

-D, --define='MACRO EXPR'

Defines MACRO with value EXPR.

BUILD OPTIONS

The general form of an rpm build command is

```
rpmbuild -bSTAGE|-rSTAGE|-tSTAGE [ rpmbuild-options  
] FILE ...
```

The argument used is -b if a spec file is being used to build the package, -r if a source package is to be rebuild and -t if rpmbuild should look inside of a (possibly compressed) tar file for the spec file to use. After the first argument, the next character (STAGE) specifies the stages of building and packaging to be done and is one of:

- ba Build binary and source packages (after doing the %prep, %build, and %install stages).
- bb Build a binary package (after doing the %prep, %build, and %install stages).
- bp Executes the "%prep" stage from the spec file. Normally this involves unpacking the sources and applying any patches.
- bc Do the "%build" stage from the spec file (after doing the %prep stage). This generally involves the equivalent of a "make".
- bi Do the "%install" stage from the spec file (after doing the %prep and %build stages). This generally involves the equivalent of a "make install".
- bl Do a "list check". The "%files" section from the spec file is expanded, and checks are made to verify that each file exists.
- bs Build just the source package.

The following options may also be used:

--buildroot DIRECTORY

When building a package, override the BuildRoot tag with directory DIRECTORY.

--clean

Remove the build tree after the packages are made.

--nobuild

Do not execute any build stages. Useful for testing out spec files.

--noprep

Do not execute %prep build stage even if present in spec.

--noclean

Do not execute %clean build stage even if present in spec.

--nocheck

Do not execute %check build stage even if present in spec.

--nodebuginfo

Do not generate debuginfo packages..

--nodeps

Do not verify build dependencies.

--rmsource

Remove the sources after the build (may also be used standalone, e.g. "rpm? build --rmsource foo.spec").

--rmspec

Remove the spec file after the build (may also be used standalone, eg. "rpm? build --rmspec foo.spec").

--short-circuit

Skip straight to specified stage (i.e., skip all stages leading up to the specified stage). Only valid with -bc, -bi, and -bb. Useful for local testing only. Packages built this way will be marked with an unsatisfiable dependency to prevent their accidental use.

--build-in-place

Build from locally checked out sources. Sets _builddir to current working directory. Skips handling of -n and untar in the %setup and the deletion of the buildSubdir.

--target PLATFORM

When building the package, interpret PLATFORM as arch-vendor-os and set the macros %_target, %_target_cpu, and %_target_os accordingly.

--with OPTION

Enable configure OPTION for build.

--without OPTION

Disable configure OPTION for build.

REBUILD AND RECOMPILE OPTIONS

There are two other ways to invoke building with rpm:

`rpmbuild --rebuild|--recompile SOURCEPKG ...`

When invoked this way, `rpmbuild` installs the named source package, and does a `prep`, `compile` and `install`. In addition, `--rebuild` builds a new binary package. When the build has completed, the build directory is removed (as in `--clean`) and the sources and spec file for the package are removed.

These options are no longer superseded by the `-r*` options which allow more fine control over what stages of the build to run.

SHOWRC

The command

```
rpmbuild --showrc
```

shows the values `rpmbuild` will use for all of the options currently set in `rpmrc` and macros configuration file(s).

FILES

rpmrc Configuration

`/usr/lib/rpm/rpmrc`

`/usr/lib/rpm/redhat/rpmrc`

`/etc/rpmrc`

`~/.rpmrc`

Macro Configuration

`/usr/lib/rpm/macros`

`/usr/lib/rpm/redhat/macros`

`/etc/rpm/macros`

`~/.rpmmacros`

Temporary

`/var/tmp/rpm*`

SEE ALSO

`gendiff(1)`,

`popt(3)`,

`rpm(8)`,

`rpm2cpio(8)`,

`rpmkeys(8)`

`rpmspec(8)`,

`rpmsign(8)`,

rpmbuild --help - as rpm supports customizing the options via popt aliases it's im?
possible to guarantee that what's described in the manual matches what's available.

<http://www.rpm.org/> <URL:<http://www.rpm.org/>>

AUTHORS

Marc Ewing <marc@redhat.com>

Jeff Johnson <jbj@redhat.com>

Erik Troan <ewt@redhat.com>

Red Hat, Inc.

09 June 2002

RPMBUILD(8)