

NAME

dpkg-buildpackage - build binary or source packages from sources

SYNOPSIS

dpkg-buildpackage [*option...*]

DESCRIPTION

dpkg-buildpackage is a program that automates the process of building a Debian package. It consists of the following steps:

1. It prepares the build environment by setting various environment variables (see **ENVIRONMENT**), runs the **init** hook, and calls **dpkg-source --before-build** (unless **-T** or **--target** has been used).
2. It checks that the build-dependencies and build-conflicts are satisfied (unless **-d** is specified).
3. If a specific target has been selected with the **-T** or **--target** option, it calls that target and stops here. Otherwise it runs the **preclean** hook and calls **fakeroot debian/rules cleanto** clean the build-tree (unless **-nc** is specified).
4. It runs the **source** hook and calls **dpkg-source -b** to generate the source package (unless a binary-only build has been requested with **-b**, **-B** or **-A**).
5. It runs the **build** hook and calls **debian/rules build-target**, then runs the **binary** hook followed by **fakeroot debian/rules binary-target** (unless a source-only build has been requested with **-S**). Note that *build-target* and *binary-target* are either **build** and **binary** (default case, or if **-b** is specified), or **build-arch** and **binary-arch** (if **-B** or **-G** are specified), or **build-indep** and **binary-indep** (if **-A** or **-g** are specified).
6. It runs the **changes** hook and calls **dpkg-genchanges** to generate a **.changes** file. Many **dpkg-buildpackage** options are forwarded to **dpkg-genchanges**.
7. It runs the **postclean** hook and if **-tc** is specified, it will call **fakeroot debian/rules clean** again.
8. It calls **dpkg-source --after-build**.
9. It runs the **check** hook and calls a package checker for the **.changes** file (if a command is specified in **DEB_CHECK_COMMAND** or with **--check-command**).
10.
It runs the **sign** hook and calls **gpg2** or **gpg** to sign the **.dsc** file (if any, unless **-us** is specified or on UNRELEASED builds), and the **.changes** file (unless **-uc** is specified or on UNRELEASED builds).
11.
It runs the **done** hook.

OPTIONS

- g** Specifies a build limited to source and architecture independent packages. Passed to **dpkg-genchanges**.
- G** Specifies a build limited to source and architecture specific packages. Passed to **dpkg-genchanges**.
- b** Specifies a binary-only build, no source files are to be built and/or distributed. Passed to **dpkg-genchanges**.
- B** Specifies a binary-only build, limited to architecture dependent packages. Passed to **dpkg-genchanges**.
- A** Specifies a binary-only build, limited to architecture independent packages. Passed to **dpkg-genchanges**.
- S** Specifies a source-only build, no binary packages need to be made. Passed to **dpkg-genchanges**.

- F** Specifies a normal full build, binary and source packages will be built. This is the same as the default case when no build option is specified.
- target=*target***
- target *target***
- T*target***
Calls **debian/rules *target*** after having setup the build environment and stops the package build process here. If **--as-root** is also given, then the command is executed as root (see **-r**). Note that official targets that are required to be run as root by the Debian policy do not need this option.
- as-root**
Only meaningful together with **--target**. Requires that the target be run with root rights.
- si**
- sa**
- sd**
- v*version***
- C*changes-description***
- m*maintainer-address***
- e*maintainer-address***
Passed unchanged to **dpkg-genchanges**. See its manual page.
- a, --host-arch *architecture***
Specify the Debian architecture we build for. The architecture of the machine we build on is determined automatically, and is also the default for the host machine.
- t, --host-type *gnu-system-type***
Specify the GNU system type we build for. It can be used in place of **--host-arch** or as a complement to override the default GNU system type of the host Debian architecture.
- target-arch *architecture***
Specify the Debian architecture the binaries built will build for. The default value is the host machine.
- target-type *gnu-system-type***
Specify the GNU system type the binaries built will build for. It can be used in place of **--target-arch** or as a complement to override the default GNU system type of the target Debian architecture.
- P*profile[,...]***
Specify the profile(s) we build, as a comma-separated list. The default behavior is to build for no specific profile. Also sets them (as a space separated list) as the **DEB_BUILD_PROFILES** environment variable which allows, for example, **debian/rules** files to use this information for conditional builds.
- j[*jobs*|**auto**]**
Number of jobs allowed to be run simultaneously, number of jobs matching the number of online processors if **auto** is specified, or unlimited number if *jobs* is not specified, equivalent to the **make(1)** option of the same name. Will add itself to the **MAKEFLAGS** environment variable, which should cause all subsequent make invocations to inherit the option. Also adds **parallel=*jobs*** or **parallel=** to the **DEB_BUILD_OPTIONS** environment variable which allows **debian/rules** files to use this information for their own purposes. The **parallel=*jobs*** or **parallel=** option in **DEB_BUILD_OPTIONS** environment variable will override the **-j** value if this option is given. Note that the **auto** value will get replaced by the actual number of currently active processors, and as such will not get propagated to any child process. If the number of online processors cannot be inferred then the code will fallback to using an unlimited number.

- D** Check build dependencies and conflicts; abort if unsatisfied. This is the default behavior.
- d** Do not check build dependencies and conflicts.
- nc** Do not clean the source tree (implies **-b** if nothing else has been selected among **-F**, **-g**, **-G**, **-B**, **-A** or **-S**).
- tc** Clean the source tree (using *gain-root-command* **debian/rules clean**) after the package has been built.

-r*gain-root-command*

When **dpkg-buildpackage** needs to execute part of the build process as root, it prefixes the command it executes with *gain-root-command* if one has been specified. Otherwise, if none has been specified, **fakeroot** will be used by default, if the command is present. *gain-root-command* should start with the name of a program on the **PATH** and will get as arguments the name of the real command to run and the arguments it should take. *gain-root-command* can include parameters (they must be space-separated) but no shell metacharacters. *gain-root-command* might typically be **fakeroot**, **sudo**, **super** or **really**. **su** is not suitable, since it can only invoke the user's shell with **-c** instead of passing arguments individually to the command to be run.

-R*rules-file*

Building a Debian package usually involves invoking **debian/rules** as a command with several standard parameters. With this option it's possible to use another program invocation to build the package (it can include space separated parameters). Alternatively it can be used to execute the standard rules file with another make program (for example by using **/usr/local/bin/make -f debian/rules** as *rules-file*).

--check-command=*check-command*

Command used to check the **.changes** file itself and any artifact built referenced in the file. The command should take the **.changes** pathname as an argument. This command will usually be **lintian**.

--check-option=*opt*

Pass option *opt* to the *check-command* specified with **DEB_CHECK_COMMAND** or **--check-command**. Can be used multiple times.

--hook-*hook-name*=*hook-command*

Set the specified shell code *hook-command* as the hook *hook-name*, which will run at the times specified in the run steps. The hooks will always be executed even if the following action is not performed (except for the **binary** hook).

Note: Hooks can affect the build process, and cause build failures if their commands fail, so watch out for unintended consequences.

The current *hook-name* supported are:

init preclean source build binary changes postclean check sign done

The *hook-command* supports the following substitution format string, which will get applied to it before execution:

- %%** A single % character.
- %a** A boolean value (0 or 1), representing whether the following action is being performed.
- %p** The source package name.
- %v** The source package version.
- %s** The source package version (without the epoch).

%u The upstream version.

-psign-command

When **dpkg-buildpackage** needs to execute GPG to sign a source control (**.dsc**) file or a **.changes** file it will run *sign-command* (searching the **PATH** if necessary) instead of **gpg2** or **gpg**. *sign-command* will get all the arguments that **gpg2** or **gpg** would have gotten. *sign-command* should not contain spaces or any other shell metacharacters.

-kkey-id

Specify a key-ID to use when signing packages.

-us Do not sign the source package.

-uc Do not sign the **.changes** file.

--force-sign

Force the signing of the resulting files (since dpkg 1.17.0), regardless of **-us** or **-uc** or other internal heuristics.

-i[regex]

-I[pattern]

-s[nsAkurKUR]

-z, -Z Passed unchanged to **dpkg-source**. See its manual page.

--source-option=opt

Pass option *opt* to **dpkg-source**. Can be used multiple times.

--changes-option=opt

Pass option *opt* to **dpkg-genchanges**. Can be used multiple times.

--admindir=dir

--admindir dir

Change the location of the **dpkg** database. The default location is */var/lib/dpkg*.

-, --help

Show the usage message and exit.

--version

Show the version and exit.

ENVIRONMENT

DEB_CHECK_COMMAND

If set, it will be used as the command to check the **.changes** file. Overridden by the **--check-command** option.

DEB_SIGN_KEYID

If set, it will be used to sign the **.changes** and **.dsc** files. Overridden by the **-k** option.

DEB_BUILD_OPTIONS

If set, and containing **nocheck** the **DEB_CHECK_COMMAND** variable will be ignored.

DEB_BUILD_PROFILES

If set, it will be used as the active build profile(s) for the package being built. It is a space separated list of profile names. Overridden by the **-P** option.

Reliance on exported environment flags

Even if **dpkg-buildpackage** exports some variables, **debian/rules** should not rely on their presence and should instead use the respective interface to retrieve the needed values.

Variables set by dpkg-architecture

dpkg-architecture is called with the **-a** and **-t** parameters forwarded. Any variable that is output by its **-s** option is integrated in the build environment.

NOTES

Compiler flags are no longer exported

Between dpkg 1.14.17 and 1.16.1, **dpkg-buildpackage** exported compiler flags (**CFLAGS**, **CXXFLAGS**, **FFLAGS**, **CPPFLAGS** and **LDFLAGS**) with values as returned by **dpkg-buildflags**. This is no longer the case.

Default build targets

dpkg-buildpackage is using the **build-arch** and **build-indep** targets since dpkg 1.16.2. Those targets are thus mandatory. But to avoid breakages of existing packages, and ease the transition, it will fallback to using the **build** target if **make -f debian/rules -qnbuild-tar get** returns 2 as exit code.

BUGS

It should be possible to specify spaces and shell metacharacters and initial arguments for *gain-root-command* and *sign-command*.

SEE ALSO

[dpkg-source\(1\)](#), [dpkg-architecture\(1\)](#), [dpkg-buildflags\(1\)](#), [dpkg-genchanges\(1\)](#), [fakeroot\(1\)](#), [lintian\(1\)](#), [gpg2\(1\)](#), [gpg\(1\)](#).