



## ***Linux Ubuntu 26.04 Manual Pages on command 'cpack.1'***

***\$ man cpack.1***

CPACK(1)

CMake

CPACK(1)

NAME

cpack - CPack Command-Line Reference

SYNOPSIS

cpack [<options>]

DESCRIPTION

The `cpack` executable is the CMake packaging program. It generates installers and source packages in a variety of formats.

For each installer or package format, `cpack` has a specific backend, called "generator". A generator is responsible for generating the required inputs and invoking the specific package creation tools. These installer or package generators are not to be confused with the makefile generators of the `cmake` [command](#).

All supported generators are specified in the `cpack-generators` [manual](#). The command `cpack --help` prints a list of generators supported for the target platform. Which of them are to be used can be selected through the `CPACK_GENERATOR` [variable](#): `CPACK_GENERATOR` variable or through the command-line option `-G`.

The `cpack` program is steered by a configuration file written in the CMake language [manual](#): `cmake-language`. Unless chosen differently through the command-line option `--config`, the file `CPackConfig.cmake` in the current directory is used.

In the standard CMake workflow, the file `CPackConfig.cmake` is generated by the `cmake` [executable](#), provided the `CPack` [module](#) is included by the project's `CMakeLists.txt` file.

OPTIONS

-G <generators>

<generators> is a semicolon-separated list <#cmake-language-lists> of generator names. cpack will iterate through this list and produce package(s) in that generator's format according to the details provided in the CPackConfig.cmake configuration file. If this option is not given, the CPACK\_GENERATOR <#variable:CPACK\_GENERATOR> variable determines the default set of generators that will be used.

-C <configurations>

Specify the project configuration(s) to be packaged (e.g. Debug, Release, etc.), where <configurations> is a semicolon-separated list <#cmake-language-lists>. When the CMake project uses a multi-configuration generator such as Xcode <#generator:Xcode> or Visual Studio <#visual-studio-generators>, this option is needed to tell cpack which built executables to include in the package. The user is responsible for ensuring that the configuration(s) listed have already been built before invoking cpack.

-D <var>=<value>

Set a CPack variable. This will override any value set for <var> in the input file read by cpack.

--config <configFile>

Specify the configuration file read by cpack to provide the packaging details. By default, CPackConfig.cmake in the current directory will be used.

-V, --verbose

Run cpack with verbose output. This can be used to show more details from the package generation tools and is suitable for project developers.

--debug

Run cpack with debug output. This option is intended mainly for the developers of cpack itself and is not normally needed by project developers.

--trace

Put the underlying cmake scripts in trace mode.

--trace-expand

Put the underlying cmake scripts in expanded trace mode.

-P <packageName>

Override/define the value of the CPACK\_PACKAGE\_NAME <#variable:CPACK\_PACKAGE\_NAME> variable used for packaging. Any value set for this variable in the CPackCon?

fig.cmake file will then be ignored.

-R <packageVersion>

Override/define the value of the CPACK\_PACKAGE\_VERSION <#variable:CPACK\_PACKAGE\_VERSION> variable used for packaging. It will override a value set in the CPackConfig.cmake file or one automatically computed from CPACK\_PACKAGE\_VERSION\_MAJOR <#variable:CPACK\_PACKAGE\_VERSION\_MAJOR>, CPACK\_PACKAGE\_VERSION\_MINOR <#variable:CPACK\_PACKAGE\_VERSION\_MINOR> and CPACK\_PACKAGE\_VERSION\_PATCH <#variable:CPACK\_PACKAGE\_VERSION\_PATCH>.

-B <packageDirectory>

Override/define CPACK\_PACKAGE\_DIRECTORY <#variable:CPACK\_PACKAGE\_DIRECTORY>, which controls the directory where CPack will perform its packaging work. The resultant package(s) will be created at this location by default and a \_CPack\_Packages subdirectory will also be created below this directory to use as a working area during package creation.

--vendor <vendorName>

Override/define CPACK\_PACKAGE\_VENDOR <#variable:CPACK\_PACKAGE\_VENDOR>.

--preset <presetName>

Use a preset from cmake-presets(7) <#manual:cmake-presets(7)>.

--list-presets

List presets from cmake-presets(7) <#manual:cmake-presets(7)>.

-version [<file>], --version [<file>], /V [<file>]

Show program name/version banner and exit. The output is printed to a named <file> if given.

-h, -H, --help, -help, -usage, /?

Print usage information and exit.

Usage describes the basic command line interface and its options.

--help <keyword> [<file>]

Print help for one CMake keyword.

<keyword> can be a property, variable, command, policy, generator or module.

The relevant manual entry for <keyword> is printed in a human-readable text format.

The output is printed to a named <file> if given.

Changed in version 3.28: Prior to CMake 3.28, this option supported command names only.

`--help-full [<file>]`

Print all help manuals and exit.

All manuals are printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-manual <man> [<file>]`

Print one help manual and exit.

The specified manual is printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-manual-list [<file>]`

List help manuals available and exit.

The list contains all manuals for which help may be obtained by using the `--help-manual` option followed by a manual name. The output is printed to a named <file> if given.

`--help-command <cmd> [<file>]`

Print help for one command and exit.

The `cmake-commands(7)` `<#manual:cmake-commands(7)>` manual entry for <cmd> is printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-command-list [<file>]`

List commands with help available and exit.

The list contains all commands for which help may be obtained by using the `--help-command` option followed by a command name. The output is printed to a named <file> if given.

`--help-commands [<file>]`

Print `cmake-commands` manual and exit.

The `cmake-commands(7)` `<#manual:cmake-commands(7)>` manual is printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-module <mod> [<file>]`

Print help for one module and exit.

The `cmake-modules(7)` `<#manual:cmake-modules(7)>` manual entry for <mod> is printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-module-list [<file>]`

List modules with help available and exit.

The list contains all modules for which help may be obtained by using the `--help-module`

ule option followed by a module name. The output is printed to a named <file> if given.

`--help-modules [<file>]`

Print cmake-modules manual and exit.

The `cmake-modules(7)` `<#manual:cmake-modules(7)>` manual is printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-policy <cmp> [<file>]`

Print help for one policy and exit.

The `cmake-policies(7)` `<#manual:cmake-policies(7)>` manual entry for <cmp> is printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-policy-list [<file>]`

List policies with help available and exit.

The list contains all policies for which help may be obtained by using the `--help-policy` option followed by a policy name. The output is printed to a named <file> if given.

`--help-policies [<file>]`

Print cmake-policies manual and exit.

The `cmake-policies(7)` `<#manual:cmake-policies(7)>` manual is printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-property <prop> [<file>]`

Print help for one property and exit.

The `cmake-properties(7)` `<#manual:cmake-properties(7)>` manual entries for <prop> are printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-property-list [<file>]`

List properties with help available and exit.

The list contains all properties for which help may be obtained by using the `--help-property` option followed by a property name. The output is printed to a named <file> if given.

`--help-properties [<file>]`

Print cmake-properties manual and exit.

The `cmake-properties(7)` `<#manual:cmake-properties(7)>` manual is printed in a human-readable text format. The output is printed to a named <file> if given.

`--help-variable <var> [<file>]`

Print help for one variable and exit.

The `cmake-variables(7)` `<#manual:cmake-variables(7)>` manual entry for `<var>` is printed in a human-readable text format. The output is printed to a named `<file>` if given.

`--help-variable-list [<file>]`

List variables with help available and exit.

The list contains all variables for which help may be obtained by using the `--help-variable` option followed by a variable name. The output is printed to a named `<file>` if given.

`--help-variables [<file>]`

Print `cmake-variables` manual and exit.

The `cmake-variables(7)` `<#manual:cmake-variables(7)>` manual is printed in a human-readable text format. The output is printed to a named `<file>` if given.

## SEE ALSO

The following resources are available to get help using CMake:

### Home Page

`<https://cmake.org>`

The primary starting point for learning about CMake.

### Online Documentation and Community Resources

`<https://cmake.org/documentation>`

Links to available documentation and community resources may be found on this web page.

### Discourse Forum

`<https://discourse.cmake.org>`

The Discourse Forum hosts discussion and questions about CMake.

## Copyright

2000-2025 Kitware, Inc. and Contributors