



Linux Ubuntu 26.04 Manual Pages on command 'pip-wheel.1'

\$ man pip-wheel.1

PIP3-WHEEL(1) pip PIP3-WHEEL(1)

NAME

pip3-wheel - description of pip3 wheel command

DESCRIPTION

Build Wheel archives for your requirements and dependencies.

Wheel is a built-package format, and offers the advantage of not recompiling your software during every install. For more details, see the wheel docs: ?

<https://wheel.readthedocs.io/en/latest/>

'pip wheel' uses the build system interface as described here: ?

<https://pip.pypa.io/en/stable/reference/build-system/>

USAGE

python -m pip wheel [options] <requirement specifier> ...

python -m pip wheel [options] -r <requirements file> ...

python -m pip wheel [options] [-e] <vcs project url> ...

python -m pip wheel [options] [-e] <local project path> ...

python -m pip wheel [options] <archive url/path> ...

OPTIONS

-w, --wheel-dir <dir>

Build wheels into <dir>, where the default is the current working directory.

(environment variable: PIP_WHEEL_DIR)

--no-binary <format_control>

Do not use binary packages. Can be supplied multiple times, and each time adds to the existing value. Accepts either ":all:" to disable all binary packages, ":none:" to

empty the set (notice the colons), or one or more package names with commas between them (no colons). Note that some packages are tricky to compile and may fail to install when this option is used on them.

(environment variable: PIP_NO_BINARY)

`--only-binary <format_control>`

Do not use source packages. Can be supplied multiple times, and each time adds to the existing value. Accepts either `":all:"` to disable all source packages, `":none:"` to empty the set, or one or more package names with commas between them. Packages without binary distributions will fail to install when this option is used on them.

(environment variable: PIP_ONLY_BINARY)

`--prefer-binary`

Prefer binary packages over source packages, even if the source packages are newer.

(environment variable: PIP_PREFER_BINARY)

`--no-build-isolation`

Disable isolation when building a modern source distribution. Build dependencies specified by PEP 518 must be already installed if this option is used.

(environment variable: PIP_NO_BUILD_ISOLATION)

`--use-pep517`

Use PEP 517 for building source distributions (use `--no-use-pep517` to force legacy behaviour).

(environment variable: PIP_USE_PEP517)

`--check-build-dependencies`

Check the build dependencies when PEP517 is used.

(environment variable: PIP_CHECK_BUILD_DEPENDENCIES)

`-c, --constraint <file>`

Constrain versions using the given constraints file. This option can be used multiple times.

(environment variable: PIP_CONSTRAINT)

`-e, --editable <path/url>`

Install a project in editable mode (i.e. setuptools "develop mode") from a local project path or a VCS url.

(environment variable: PIP_EDITABLE)

`-r, --requirement <file>`

Install from the given requirements file. This option can be used multiple times.

(environment variable: PIP_REQUIREMENT)

`--src <dir>`

Directory to check out editable projects into. The default in a virtualenv is "`<venv path>/src`". The default for global installs is "`<current dir>/src`".

(environment variable: PIP_SRC, PIP_SOURCE, PIP_SOURCE_DIR, PIP_SOURCE_DIRECTORY)

`--ignore-requires-python`

Ignore the Requires-Python information.

(environment variable: PIP_IGNORE_REQUIRES_PYTHON)

`--no-deps`

Don't install package dependencies.

(environment variable: PIP_NO_DEPS, PIP_NO_DEPENDENCIES)

`--progress-bar <progress_bar>`

Specify whether the progress bar should be used [on, off, raw] (default: on)

(environment variable: PIP_PROGRESS_BAR)

`--no-verify`

Don't verify if built wheel is valid.

(environment variable: PIP_NO_VERIFY)

`-C, --config-settings <settings>`

Configuration settings to be passed to the PEP 517 build backend. Settings take the form KEY=VALUE. Use multiple `--config-settings` options to pass multiple keys to the backend.

(environment variable: PIP_CONFIG_SETTINGS)

`--build-option <options>`

Extra arguments to be supplied to 'setup.py bdist_wheel'.

(environment variable: PIP_BUILD_OPTION)

`--global-option <options>`

Extra global options to be supplied to the setup.py call before the install or bdist_wheel command.

(environment variable: PIP_GLOBAL_OPTION)

`--pre` Include pre-release and development versions. By default, pip only finds stable versions.

(environment variable: PIP_PRE)

--require-hashes

Require a hash to check each requirement against, for repeatable installs. This option is implied when any package in a requirements file has a --hash option.

(environment variable: PIP_REQUIRE_HASHES)

--group <[path:]group>

Install a named dependency-group from a "pyproject.toml" file. If a path is given, the name of the file must be "pyproject.toml". Defaults to using "pyproject.toml" in the current directory.

(environment variable: PIP_GROUP)

--no-clean

Don't clean up build directories.

(environment variable: PIP_NO_CLEAN)

AUTHOR

pip developers

COPYRIGHT

The pip developers

25.1

Sep 22, 2025

PIP3-WHEEL(1)