cmake(1)

Synopsis

```
cmake [<options>] (<path-to-source> | <path-to-existing-build>)
cmake [(-D <var>=<value>)...] -P <cmake-script-file>
cmake --build <dir> [<options>] [-- <build-tool-options>...]
cmake -E <command> [<options>...]
cmake --find-package <options>...
```

Description

The “cmake” executable is the CMake command-line interface. It may be used to configure projects in scripts. Project configuration settings may be specified on the command line with the -D option.

CMake is a cross-platform build system generator. Projects specify their build process with platform-independent CMake listfiles included in each directory of a source tree with the name CMakeLists.txt. Users build a project by using CMake to generate a build system for a native tool on their platform.

Options

`-C <initial-cache>`

Pre-load a script to populate the cache.

When cmake is first run in an empty build tree, it creates a CMakeCache.txt file and populates it with customizable settings for the project. This option may be used to specify a file from which to load cache entries before the first pass through the project’s cmake listfiles. The loaded entries take priority over the project’s default values. The given file should be a CMake script containing SET commands that use the CACHE option, not a cache-format file.

`-D <var>:<type>=<value>, -D <var>=<value>`

Create a cmake cache entry.

When cmake is first run in an empty build tree, it creates a CMakeCache.txt file and populates it with customizable settings for the project. This option may be used to specify a setting that takes priority over the project’s default value. The option may be repeated for as many cache entries as desired.

If the `:<type>` portion is given it must be one of the types specified by the `set()` command documentation for its `CACHE` signature. If the `:<type>` portion is omitted the entry will be created with no type if it does not exist with a type already. If a command in the project sets the type to `PATH` or `FILEPATH` then the `<value>` will be converted to an absolute path.

This option may also be given as a single argument: `-D<var>:<type>=<value>` or `-D<var>=<value>`.

`-U <globbing_expr>`

Remove matching entries from CMake cache.

This option may be used to remove one or more variables from the CMakeCache.txt file, globbing expressions using * and ? are supported. The option may be repeated for as many cache entries as desired.

Use with care, you can make your CMakeCache.txt non-working.

`-G <generator-name>`

Specify a build system generator.

CMake may support multiple native build systems on certain platforms. A generator is responsible for generating a particular build system. Possible generator names are specified in the `cmake-generators(7)` manual.

`-T <toolset-name>`

Specify toolset name if supported by generator.

Some CMake generators support a toolset name to be given to the native build system to choose a compiler. This is supported only on specific generators:

```
Visual Studio >= 10
Xcode >= 3.0
```

See native build system documentation for allowed toolset names.

`-A <platform-name>`

Specify platform name if supported by generator.

Some CMake generators support a platform name to be given to the native build system to choose a compiler or SDK. This is supported only on specific generators:

```
Visual Studio >= 8
```

See native build system documentation for allowed platform names.

`-Wno-dev`

Suppress developer warnings.

Suppress warnings that are meant for the author of the CMakeLists.txt files. By default this will also turn off deprecation warnings.

`-Wdev`

Enable developer warnings.

Enable warnings that are meant for the author of the CMakeLists.txt files. By default this will also turn on deprecation warnings.

`-Werror=dev`

Make developer warnings errors.

Make warnings that are meant for the author of the CMakeLists.txt files errors. By default this will also turn on deprecated warnings as errors.

`-Wno-error=dev`

Make developer warnings not errors.

Make warnings that are meant for the author of the CMakeLists.txt files not errors. By default this will also turn off deprecated warnings as errors.

`-Wdeprecated`

Enable deprecated functionality warnings.

Enable warnings for usage of deprecated functionality, that are meant for the author of the CMakeLists.txt files.

`-Wno-deprecated`

Suppress deprecated functionality warnings.

Suppress warnings for usage of deprecated functionality, that are meant for the author of the CMakeLists.txt files.

`-Werror=deprecated`

Make deprecated macro and function warnings errors.

Make warnings for usage of deprecated macros and functions, that are meant for the author of the CMakeLists.txt files, errors.

`-Wno-error=deprecated`

Make deprecated macro and function warnings not errors.

Make warnings for usage of deprecated macros and functions, that are meant for the author of the CMakeLists.txt files, not errors.

`-E <command> [<options>...]`

See Command-Line Tool Mode.

`-L[A][H]`

List non-advanced cached variables.

List cache variables will run CMake and list all the variables from the CMake cache that are not marked as INTERNAL or ADVANCED. This will effectively display current CMake settings, which can then be changed with -D option. Changing some of the variables may result in more variables being created. If A is specified, then it will display also advanced variables. If H is specified, it will also display help for each variable.

`--build <dir>`

Build a CMake-generated project binary tree.

This abstracts a native build tool’s command-line interface with the following options:

```
<dir>          = Project binary directory to be built.
--target <tgt> = Build <tgt> instead of default targets.
--config <cfg> = For multi-configuration tools, choose <cfg>.
--clean-first  = Build target 'clean' first, then build.
                 (To clean only, use --target 'clean'.)
--use-stderr   = Ignored.  Behavior is default in CMake >= 3.0.
--             = Pass remaining options to the native tool.
```

Run cmake –build with no options for quick help.

`-N`

View mode only.

Only load the cache. Do not actually run configure and generate steps.

`-P <file>`

Process script mode.

Process the given cmake file as a script written in the CMake language. No configure or generate step is performed and the cache is not modified. If variables are defined using -D, this must be done before the -P argument.

`--find-package`

Run in pkg-config like mode.

Search a package using find_package() and print the resulting flags to stdout. This can be used to use cmake instead of pkg-config to find installed libraries in plain Makefile-based projects or in autoconf-based projects (via share/aclocal/cmake.m4).

`--graphviz=[file]`

Generate graphviz of dependencies, see CMakeGraphVizOptions.cmake for more.

Generate a graphviz input file that will contain all the library and executable dependencies in the project. See the documentation for CMakeGraphVizOptions.cmake for more details.

`--system-information [file]`

Dump information about this system.

Dump a wide range of information about the current system. If run from the top of a binary tree for a CMake project it will dump additional information such as the cache, log files etc.

`--debug-trycompile`

Do not delete the try_compile build tree. Only useful on one try_compile at a time.

Do not delete the files and directories created for try_compile calls. This is useful in debugging failed try_compiles. It may however change the results of the try-compiles as old junk from a previous try-compile may cause a different test to either pass or fail incorrectly. This option is best used for one try-compile at a time, and only when debugging.

`--debug-output`

Put cmake in a debug mode.

Print extra information during the cmake run like stack traces with message(send_error) calls.

`--trace`

Put cmake in trace mode.

Print a trace of all calls made and from where.

`--trace-expand`

Put cmake in trace mode.

Like `--trace`, but with variables expanded.

`--warn-uninitialized`

Warn about uninitialized values.

Print a warning when an uninitialized variable is used.

`--warn-unused-vars`

Warn about unused variables.

Find variables that are declared or set, but not used.

`--no-warn-unused-cli`

Don’t warn about command line options.

Don’t find variables that are declared on the command line, but not used.

`--check-system-vars`

Find problems with variable usage in system files.

Normally, unused and uninitialized variables are searched for only in CMAKE_SOURCE_DIR and CMAKE_BINARY_DIR. This flag tells CMake to warn about other files as well.

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

Print usage information and exit.

Usage describes the basic command line interface and its options.

`--version,-version,/V [<f>]`

Show program name/version banner and exit.

If a file is specified, the version is written into it. The help is printed to a named <f>ile if given.

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

Print all help manuals and exit.

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

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

Print one help manual and exit.

The specified manual is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

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 help is printed to a named <f>ile if given.

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

Print help for one command and exit.

The **`cmake-commands(7)`** manual entry for `<cmd>` is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

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 help is printed to a named <f>ile if given.

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

Print cmake-commands manual and exit.

The **`cmake-commands(7)`** manual is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

Print help for one module and exit.

The **`cmake-modules(7)`** manual entry for `<mod>` is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

List modules with help available and exit.

The list contains all modules for which help may be obtained by using the `--help-module` option followed by a module name. The help is printed to a named <f>ile if given.

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

Print cmake-modules manual and exit.

The `cmake-modules(7)` manual is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

Print help for one policy and exit.

The `cmake-policies(7)` manual entry for `<cmp>` is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

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 help is printed to a named <f>ile if given.

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

Print cmake-policies manual and exit.

The `cmake-policies(7)` manual is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

Print help for one property and exit.

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

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

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 help is printed to a named <f>ile if given.

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

Print cmake-properties manual and exit.

The `cmake-properties(7)` manual is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

Print help for one variable and exit.

The `cmake-variables(7)` manual entry for `<var>` is printed in a human-readable text format. The help is printed to a named <f>ile if given.

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

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 help is printed to a named <f>ile if given.

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

Print cmake-variables manual and exit.

The `cmake-variables(7)` manual is printed in a human-readable text format. The help is printed to a named <f>ile if given.

Command-Line Tool Mode

CMake provides builtin command-line tools through the signature:

```
cmake -E <command> [<options>...]
```

Run `cmake -E` or `cmake -E help` for a summary of commands. Available commands are:

`chdir <dir> <cmd> [<arg>...]`

Change the current working directory and run a command.

`compare_files <file1> <file2>`

Check if `<file1>` is same as `<file2>`. If files are the same, then returns 0, if not itreturns 1.

`copy <file>... <destination>`

Copy files to `<destination>` (either file or directory). If multiple files are specified, the `<destination>` must be directory and it must exist.

`copy_directory <dir>... <destination>`

Copy directories to `<destination>` directory. If `<destination>` directory does not exist it will be created.

`copy_if_different <file>... <destination>`

Copy files to `<destination>` (either file or directory) if they have changed. If multiple files are specified, the `<destination>` must be directory and it must exist.

`echo [<string>...]`

Displays arguments as text.

`echo_append [<string>...]`

Displays arguments as text but no new line.

`env [--unset=NAME]... [NAME=VALUE]... COMMAND [ARG]...`

Run command in a modified environment.

`environment`

Display the current environment variables.

`make_directory <dir>...`

Create `<dir>` directories. If necessary, create parent directories too. If a directory already exists it will be silently ignored.

`md5sum <file>...`

Compute md5sum of files.

`remove [-f] <file>...`

Remove the file(s), use `-f` to force it. If a file does not exist it will be silently ignored.

`remove_directory <dir>`

Remove a directory and its contents. If a directory does not exist it will be silently ignored.

`rename <oldname> <newname>`

Rename a file or directory (on one volume).

`sleep <number>...`

Sleep for given number of seconds.

`tar [cxt][vf][zjJ] file.tar [<options>...] [--] [<file>...]`

Create or extract a tar or zip archive. Options are:

`--`

Stop interpreting options and treat all remaining arguments as file names even if they start in `-`.

`--files-from=<file>`

Read file names from the given file, one per line. Blank lines are ignored. Lines may not start in `-` except for `--add-file=<name>` to add files whose names start in `-`.

`--mtime=<date>`

Specify modification time recorded in tarball entries.

`--format=<format>`

Specify the format of the archive to be created. Supported formats are: `7zip`, `gnutar`, `pax`, `paxr` (restricted pax, default), and `zip`.

`time <command> [<args>...]`

Run command and return elapsed time.

`touch <file>`

Touch a file.

`touch_nocreate <file>`

Touch a file if it exists but do not create it. If a file does not exist it will be silently ignored.

UNIX-specific Command-Line Tools

The following `cmake -E` commands are available only on UNIX:

`create_symlink <old> <new>`

Create a symbolic link `<new>` naming `<old>`.

Windows-specific Command-Line Tools

The following `cmake -E` commands are available only on Windows:

`delete_regv <key>`

Delete Windows registry value.

`env_vs8_wince <sdkname>`

Displays a batch file which sets the environment for the provided Windows CE SDK installed in VS2005.

`env_vs9_wince <sdkname>`

Displays a batch file which sets the environment for the provided Windows CE SDK installed in VS2008.

`write_regv <key> <value>`

Write Windows registry value.

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.

Frequently Asked Questions

https://cmake.org/Wiki/CMake_FAQ

A Wiki is provided containing answers to frequently asked questions.

Online Documentation

https://cmake.org/documentation

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

Mailing List

https://cmake.org/mailing-lists

For help and discussion about using cmake, a mailing list is provided at cmake@cmake.org. The list is member-post-only but one may sign up on the CMake web page. Please first read the full documentation at https://cmake.org before posting questions to the list.