NAME

ctest - CTest Command-Line Reference

SYNOPSIS

```
ctest [<options>]
```

DESCRIPTION

The "ctest" executable is the CMake test driver program. CMake-generated build trees created for projects that use the ENABLE_TESTING and ADD_TEST commands have testing support. This program will run the tests and report results.

OPTIONS

-C <cfg>, --build-config <cfg>

Choose configuration to test.

Some CMake-generated build trees can have multiple build configurations in the same tree. This option can be used to specify which one should be tested. Example configurations are "Debug" and "Release".

-V,--verbose

Enable verbose output from tests.

Test output is normally suppressed and only summary information is displayed. This option will show all test output.

-VV,--extra-verbose

Enable more verbose output from tests.

Test output is normally suppressed and only summary information is displayed. This option will show even more test output.

--debug

Displaying more verbose internals of CTest.

This feature will result in a large number of output that is mostly useful for debugging dashboard problems.

--output-on-failure

Output anything outputted by the test program if the test should fail. This option can also be enabled by setting the environment variable CTEST_OUTPUT_ON_FAILURE

-F

Enable failover.

This option allows ctest to resume a test set execution that was previously interrupted. If no interruption occurred, the -F option will have no effect.

-j <jobs>, --parallel <jobs>

Run the tests in parallel using thegiven number of jobs.

This option tells ctest to run the tests in parallel using given number of jobs. This option can also be set by setting the environment variable CTEST_PARALLEL_LEVEL.

-Q,--quiet

Make ctest quiet.

This option will suppress all the output. The output log file will still be generated if the --output-log is specified. Options such as --verbose, --extra-verbose, and --debug are ignored if --quiet is specified.

-O <file>, --output-log <file>

Output to log file

This option tells ctest to write all its output to a log file.

-N,--show-only
Disable actual execution of tests.

This option tells ctest to list the tests that would be run but not actually run them. Useful in conjunction with the -R and -E options.

-L <regex>, --label-regex <regex>
Run tests with labels matching regular expression.

This option tells ctest to run only the tests whose labels match the given regular expression.

-R <regex>, --tests-regex <regex>
Run tests matching regular expression.

This option tells ctest to run only the tests whose names match the given regular expression.

-E <regex>, --exclude-regex <regex>
Exclude tests matching regular expression.

This option tells ctest to NOT run the tests whose names match the given regular expression.

-LE <regex>, --label-exclude <regex>
Exclude tests with labels matching regular expression.

This option tells ctest to NOT run the tests whose labels match the given regular expression.

-D <dashboard>, --dashboard <dashboard>
Execute dashboard test

This option tells ctest to act as a Dart client and perform a dashboard test. All tests are <Mode><Test>, where Mode can be Experimental, Nightly, and Continuous, and Test can be Start, Update, Configure, Build, Test, Coverage, and Submit.

-D <var>:<type>=<value>
Define a variable for script mode

Pass in variable values on the command line. Use in conjunction with -S to pass variable values to a dashboard script. Parsing -D arguments as variable values is only attempted if the value following -D does not match any of the known dashboard types.

-M <model>, --test-model <model>
Sets the model for a dashboard

This option tells ctest to act as a Dart client where the TestModel can be Experimental, Nightly, and Continuous. Combining -M and -T is similar to -D

-T <action>, --test-action <action>
Sets the dashboard action to perform

This option tells ctest to act as a Dart client and perform some action such as start, build, test etc. Combining -M and -T is similar to -D

--track <track>
Specify the track to submit dashboard to

Submit dashboard to specified track instead of default one. By default, the dashboard is submitted to Nightly, Experimental, or Continuous track, but by specifying this option, the track can be arbitrary.

-S <script>, --script <script>
Execute a dashboard for a configuration

This option tells ctest to load in a configuration script which sets a number of parameters such as the binary and source directories. Then ctest will do what is required to create and run a dashboard. This option basically sets up a dashboard and then runs ctest -D with the appropriate options.

-SP <script>, --script-new-process <script>

Execute a dashboard for a configuration

This option does the same operations as -S but it will do them in a separate process. This is primarily useful in cases where the script may modify the environment and you do not want the modified environment to impact other -S scripts.

-A <file>, --add-notes <file>

Add a notes file with submission

This option tells ctest to include a notes file when submitting dashboard.

-I [Start,End,Stride,test#,test#|Test file], --tests-information

Run a specific number of tests by number.

This option causes ctest to run tests starting at number Start, ending at number End, and incrementing by Stride. Any additional numbers after Stride are considered individual test numbers. Start, End,or stride can be empty. Optionally a file can be given that contains the same syntax as the command line.

-U, --union

Take the Union of -I and -R

When both -R and -I are specified by default the intersection of tests are run. By specifying -U the union of tests is run instead.

--rerun-failed

Run only the tests that failed previously

This option tells ctest to perform only the tests that failed during its previous run. When this option is specified, ctest ignores all other options intended to modify the list of tests to run (-L, -R, -E, -LE, -I, etc). In the event that CTest runs and no tests fail, subsequent calls to ctest with the --rerun-failed option will run the set of tests that most recently failed (if any).

--max-width <width>

Set the max width for a test name to output

Set the maximum width for each test name to show in the output. This allows the user to widen the output to avoid clipping the test name which can be very annoying.

--interactive-debug-mode [0|1]

Set the interactive mode to 0 or 1.

This option causes ctest to run tests in either an interactive mode or a non-interactive mode. On Windows this means that in non-interactive mode, all system debug pop up windows are blocked. In dashboard mode (Experimental, Nightly, Continuous), the default is non-interactive. When just running tests not for a dashboard the default is to allow popups and interactive debugging.

--no-label-summary

Disable timing summary information for labels.

This option tells ctest not to print summary information for each label associated with the tests run. If there are no labels on the tests, nothing extra is printed.

--build-and-test

Configure, build and run a test.

This option tells ctest to configure (i.e. run cmake on), build, and or execute a test. The configure and test steps are optional. The arguments to this command line are the source and binary directories. By default this will run CMake on the Source/Bin directories specified unless --build-nocmake is specified. The --build-generator option *must* be provided to use --build-and-test. If --test-command is specified then that will be run after the build is complete. Other options that affect this mode are --build-target --build-nocmake, --build-run-dir, --build-two-config, --build-exe-dir, --build-project,--build-noclean, --build-options

--build-target
Specify a specific target to build.

This option goes with the --build-and-test option, if left out the all target is built.

--build-nocmake
Run the build without running cmake first.

Skip the cmake step.

--build-run-dir
Specify directory to run programs from.

Directory where programs will be after it has been compiled.

--build-two-config
Run CMake twice

--build-exe-dir
Specify the directory for the executable.

--build-generator
Specify the generator to use.

--build-generator-toolset
Specify the generator-specific toolset.

--build-project
Specify the name of the project to build.

--build-makeprogram
Override the make program chosen by CTest with a given one.

--build-noclean
Skip the make clean step.

--build-config-sample
A sample executable to use to determine the configuration

A sample executable to use to determine the configuration that should be used. e.g. Debug/Release/etc

--build-options
Add extra options to the build step.

This option must be the last option with the exception of --test-command

--test-command
The test to run with the --build-and-test option.

--test-timeout
The time limit in seconds, internal use only.

--tomorrow-tag
Nightly or experimental starts with next day tag.

This is useful if the build will not finish in one day.

--ctest-config
The configuration file used to initialize CTest state when submitting dashboards.

This option tells CTest to use different initialization file instead of CTestConfiguration.tcl. This way multiple initialization files can be used for example to submit to multiple dashboards.

--overwrite
Overwrite CTest configuration option.

By default ctest uses configuration options from configuration file. This option will overwrite the configuration option.

--extra-submit <file>[;<file>]
Submit extra files to the dashboard.

This option will submit extra files to the dashboard.

--force-new-ctest-process
Run child CTest instances as new processes

By default CTest will run child CTest instances within the same process. If this behavior is not desired, this argument will enforce new processes for child CTest processes.

--schedule-random
Use a random order for scheduling tests

This option will run the tests in a random order. It is commonly used to detect implicit dependencies in a test suite.

--submit-index
Submit individual dashboard tests with specific index

This option allows performing the same CTest action (such as test) multiple times and submit all stages to the same dashboard (Dart2 required). Each execution requires different index.

--timeout <seconds>
Set a global timeout on all tests.

This option will set a global timeout on all tests that do not already have a timeout set on them.

--stop-time <time>
Set a time at which all tests should stop running.

Set a real time of day at which all tests should timeout. Example: 7:00:00 -0400. Any time format understood by the curl date parser is accepted. Local time is assumed if no timezone is specified.

--http1.0
Submit using HTTP 1.0.

This option will force CTest to use HTTP 1.0 to submit files to the dashboard, instead of HTTP 1.1.

--no-compress-output
Do not compress test output when submitting.

This flag will turn off automatic compression of test output. Use this to maintain compatibility with an older version of CDash which doesn't support compressed test output.

--print-labels
Print all available test labels.

This option will not run any tests, it will simply print the list of all labels associated with the test set.

--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.

SEE ALSO

The following resources are available to get help using CMake:

Home Page

http://www.cmake.org

The primary starting point for learning about CMake.

Frequently Asked Questions

http://www.cmake.org/Wiki/CMake_FAQ

A Wiki is provided containing answers to frequently asked questions.

Online Documentation

http://www.cmake.org/HTML/Documentation.html

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

Mailing List

http://www.cmake.org/HTML/MailingLists.html

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 *http://www.cmake.org* before posting questions to the list.

COPYRIGHT