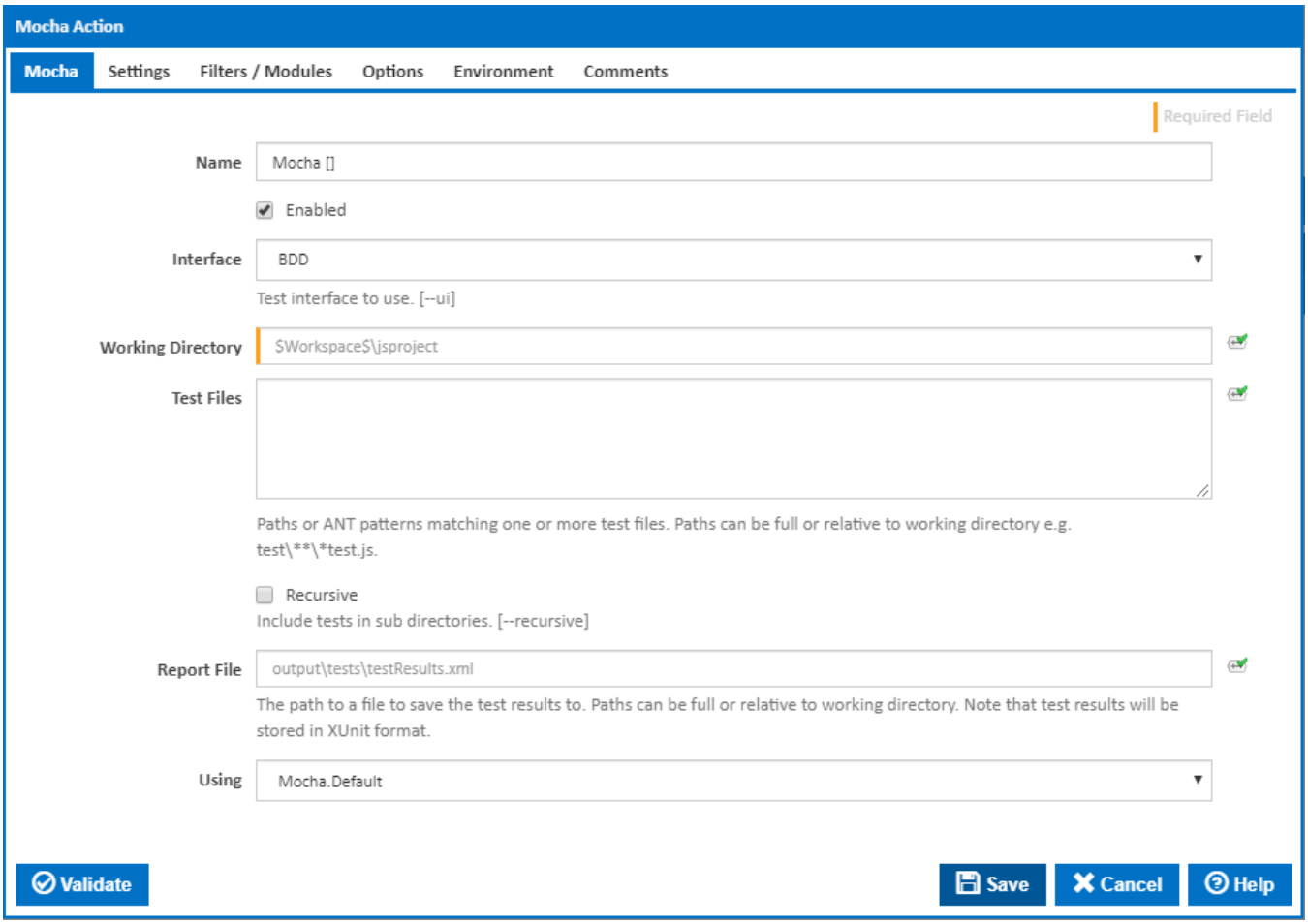
Mocha Action

The Mocha action in Continua is a wrapper around the mocha command line. If you're having trouble using the Mocha action, please refer to the Command Line Reference.

The Mocha action is used to unit test JavaScript running on node.js

Mocha



Name

A friendly name for this action (will be displayed in the actions workflow area).

Enabled

Determines if this action will be run within the relevant stage.

Interface

The domain specific language or interface you are using in your tests. Choose either the BDD, TDD, Exports, QUnit and Require-style interface. [--ui]

Working Directory

Where to run the Mocha command line. Mocha will look for tests in a 'test' subfolder under the working directory.

Test Files

Paths or ANT patterns matching one or more test files.

Recursive

Tick this to look for text files in subdirectories of the 'test' folder. [--recursive]

Report File

Where to save the test results. Path can be full or relative to working directory. Note that test results will be stored in XUnit format.

Using

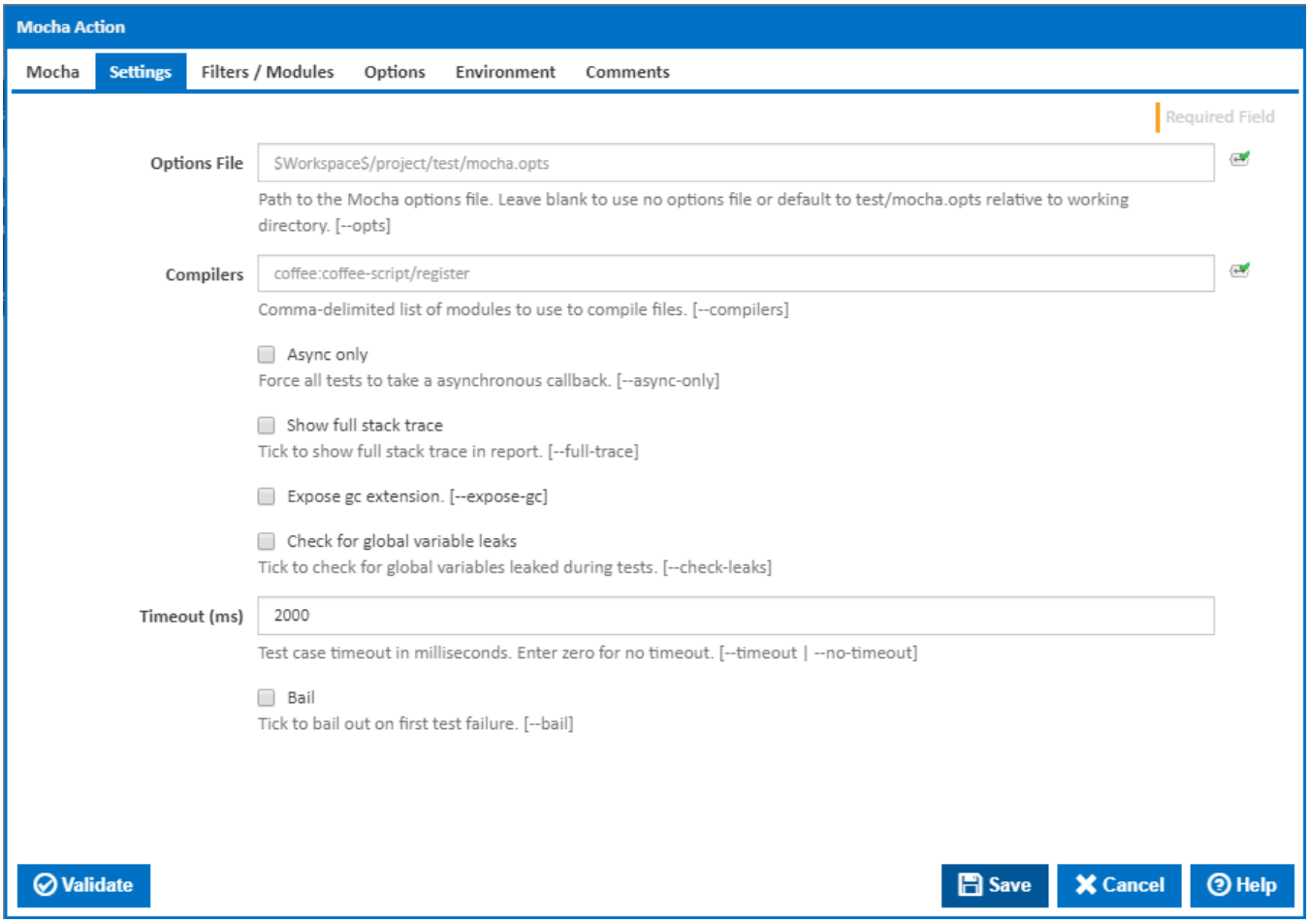
The Using drop down is populated with any property collector whose namespace matches the pattern defined by the Mocha action. The pattern for this action is `^Mocha.*`

If you create a property collector for this action, make sure you select the **Path Finder PlugIn** type and give it a name that will match the pattern above in blue. Example names listed here, search the table's Plugin column for "**Mocha**".

For more in-depth explanations on property collectors see Property Collectors.

Alternatively, you can select the **Custom** option from the Using drop down list and specify a path in the resulting input field that will be displayed. Please read Why it's a good idea to use a property collector before using this option.

Settings



Options File

Path to the Mocha options file. Leave blank to use no options file or default to test/mocha.opts relative to working directory. [--opts]

Compilers

A Comma-delimited list of compiler modules to use to compile files. [--compilers]

Async only

Tick this to force all tests to take a asynchronous callback. [--async-only]

Show full stack trace

Tick to show full stack trace in the report file. [--full-trace]

Expose gc extension

Tick to expose the JavaScript garbage collector to your tests so you can, for example, force garbage collection.\ [--expose-gc]

Check for global variable leaks

Tick to check for any global variable leaks during tests. [--check-leaks]

Accepted Global Variables

Visible only if the '**Check for global variable leaks**' checkbox is ticked.

Comma-delimited list of accepted global variable names. You can use * as a wildcard. Leave empty to check all globals. Use '*' to ignore all globals. [--globals]

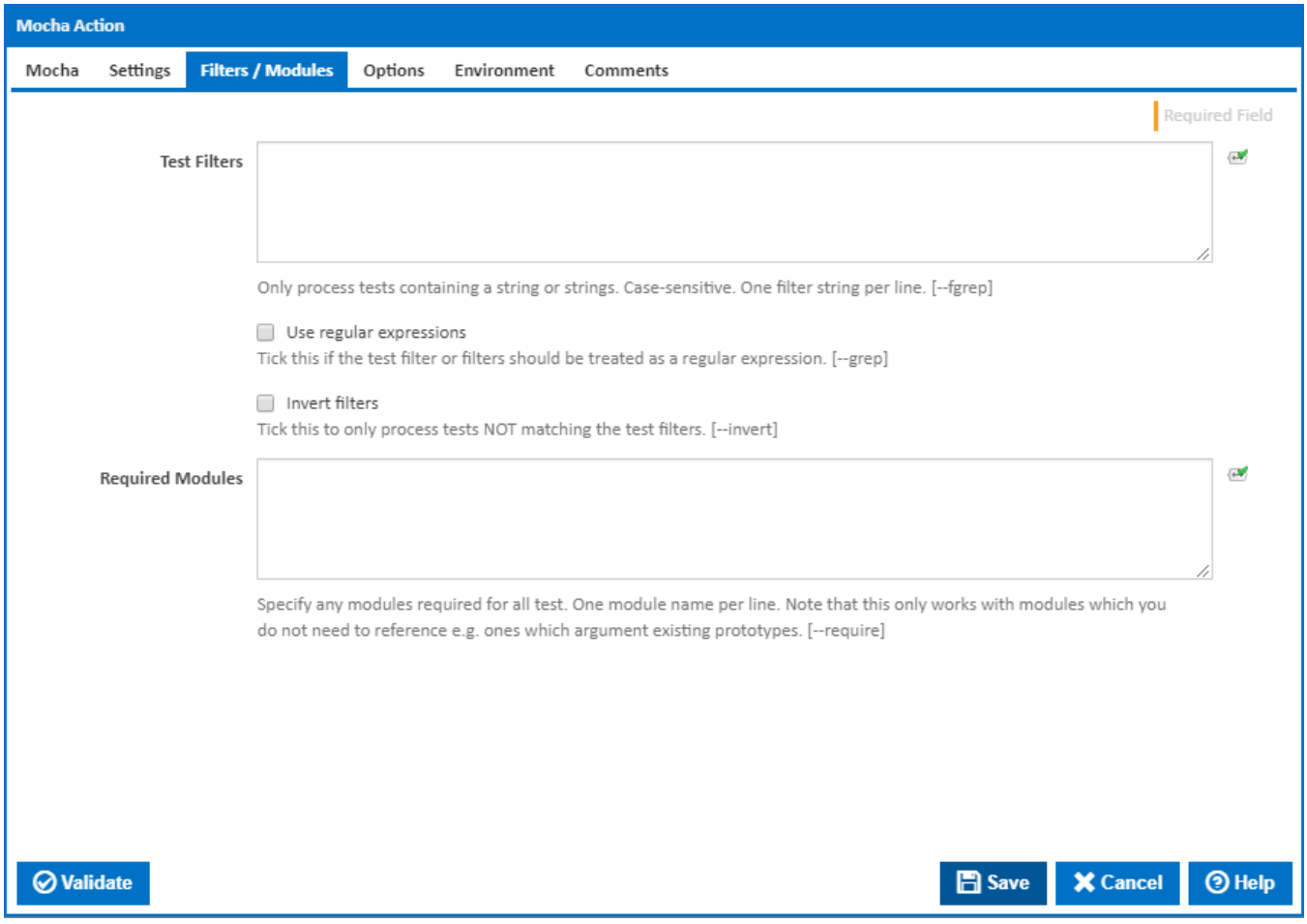
Timeout (ms)

Specify the time to wait for a test to finish. If the test does not finish before the timeout the the test will fail. Enter zero for no timeout. [–timeout | --no-timeout]

Bail

Tick to exit action if one test fails without running any further tests. [--bail]

Filters / Modules



Test Filters

Use this to filter tests to those containing a specific string or one of a list of strings. The match is case-sensitive with one filter string per line. [--fgrep]

Use regular expressions

Tick this to treat each test filter string as a regular expression. Overrides the command line argument --fgrep to --grep

Invert filters

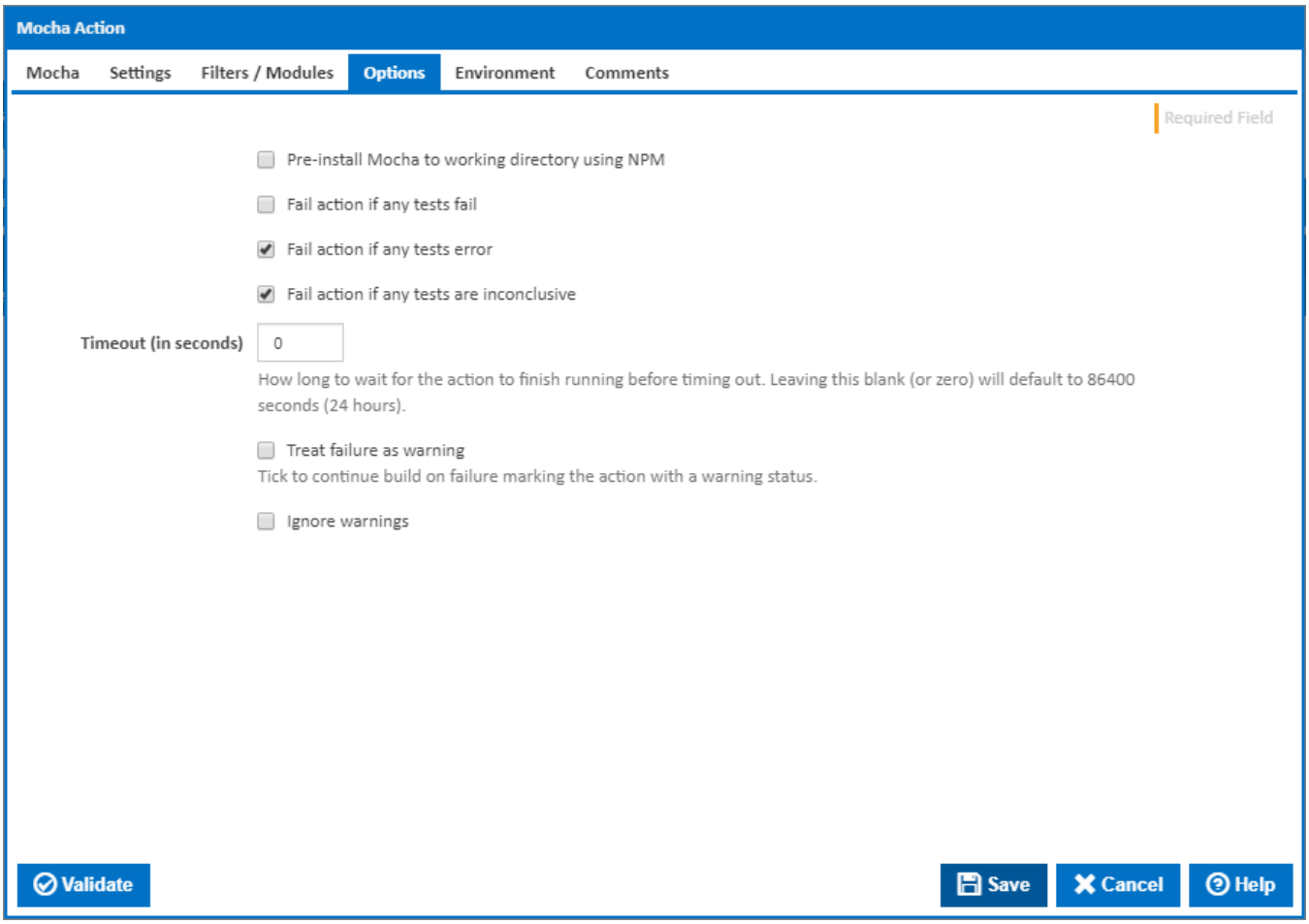
Tick this to us the filter strings to exclude tests. Only tests not matching any of the strings (or regular expressions) will be processed. [--invert]

Required Modules

Specify any modules required for all test. One module name per line. [--required]

Note: This only works with modules which you do not need to reference e.g. ones which argument existing prototypes.

Options



Pre-install Mocha to working directory using NPM

Mocha needs to be installed to the working directory before the tasks can be run. Tick this to automatically install Mocha before running the Mocha command line. Leave this cleared if you are installing Mocha with a separate preceding NPM action.

Run NPM Using

Visible only if the '**Pre-install Mocha to working directory using NPM**' checkbox is ticked.

The property collector pointing to the location of the NPM.exe file.

Fail action if any tests fail

Tick this to cause the build to fail if any tests fail.

Fail action if any tests error

Tick this to cause the build to fail if an error occurred while running any test.

Fail action if any tests are inconclusive

Tick this to cause the build to fail if any tests are inconclusive.

Timeout (in seconds)

How long to wait for the action to finish running before timing out. Leaving this blank (or zero) will default to 86400 seconds (24 hours).

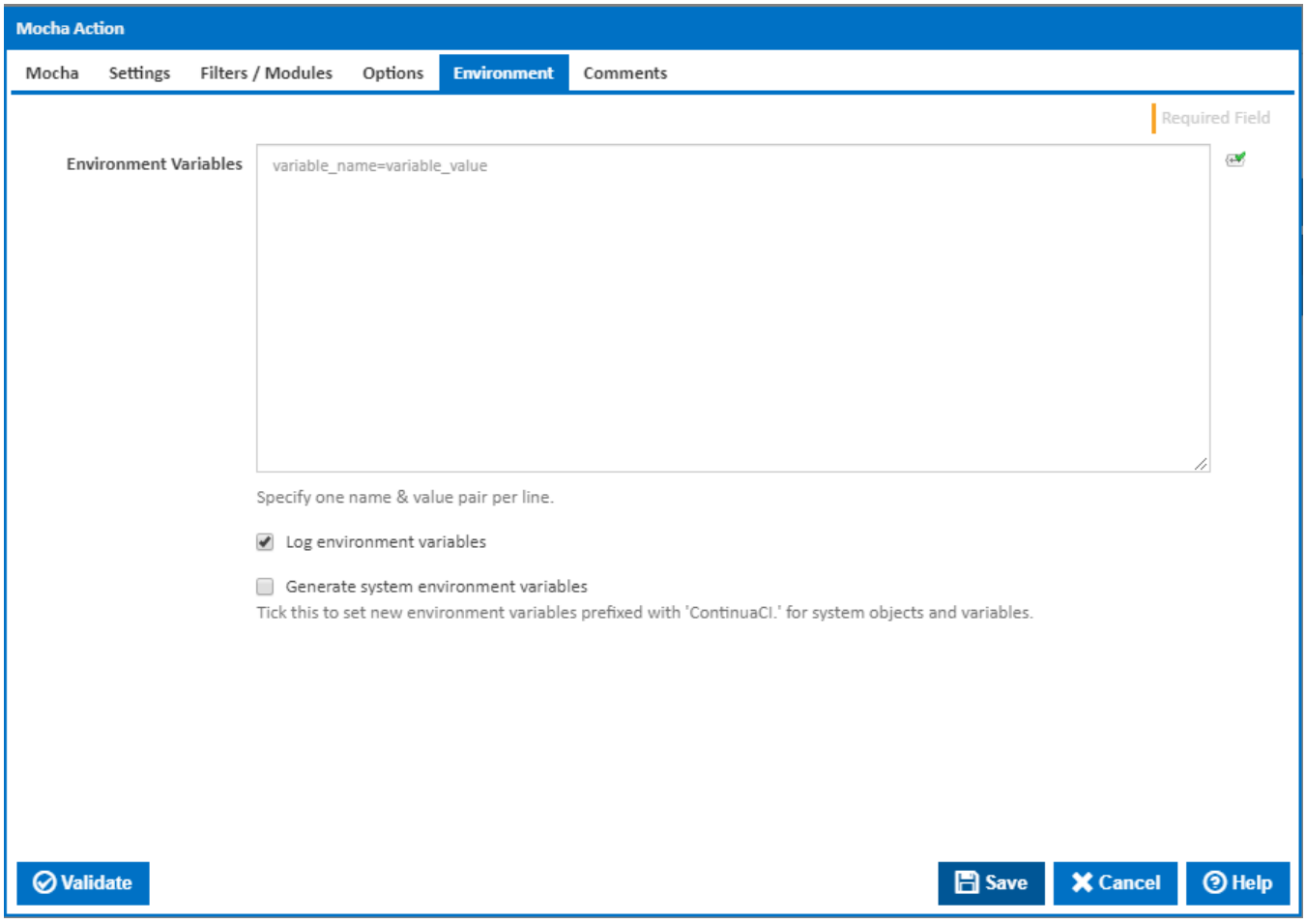
Treat failure as warning

Tick to continue build on failure marking the action with a warning status.

Ignore warnings

If this is ticked, any warnings logged will not mark the action with a warning status.

Environment



Environment Variables

Multiple environment variables can be defined - one per line. These are set before the command line is run.

Log environment variables

If this is ticked, environment variable values are written to the build log.

Generate system environment variables

Tick this checkbox to set up a list of new environment variables prefixed with 'ContinuaCI.' for all current system expression objects and variables.

Mask sensitive variable values in system environment variables

This checkbox is visible only if the '**Generate system environment variables**' checkbox is ticked.

If this is ticked, the values of any variables marked as sensitive will be masked with **** when setting system environment variables. Clear this to expose the values.