

Command Line Options

This page lists the command line options for `dottestcli`.

- [Configuring the Test Configuration](#)
 - `-config <config_url>`
 - `-listconfigs`
- [Configuring the Input Scope](#)
 - `-solution <path>`
 - `-project <path>`
 - `-website <path>`
 - `-solutionConfig`
 - `-projectConfig`
 - `-targetPlatform`
 - `-resource <path relative to the solution>`
 - `-include <absolute path>` and `-exclude <absolute path>`
 - `-reference <path>`
- [Reporting](#)
 - `-report <path>`
 - `-publish`
- [Customizing Configuration](#)
 - `-settings <path>`
 - `-property <key>=<value>`
 - `-showsettings`
- [Collecting Coverage](#)
 - `-staticCoverage`
 - `-runtimeCoverage`
- [Other Options](#)
 - `-machineld`
 - `-encodepass <your password>`
 - `-nobuild`
 - `-out <path>`
 - `-fail`
 - `-version`
 - `-help`

Configuring the Test Configuration

`-config <config_url>`

This option allows you to specify the test configuration that will be used for analysis. The option must be followed by the name of a built-in, user-defined, or DTP-hosted test configuration. Examples:

- `dottestcli -config "builtin://Critical Rules"`
- `dottestcli -config "user://My_Config"`
- `dottestcli -config "dtp://New_Rules"`

`-listconfigs`

This option prints a list of available test configurations and can be used to obtain a valid test configuration name you can pass with the `-config` option.

See [Configuring Test Configurations](#) for details.

Configuring the Input Scope

`-solution <path>`

This option specifies the path to the solution to be analyzed. You can use Ant-style wildcards to specify patterns. Example:

- `-solution "C:\Devel\FooSolution\FooSolution.sln"`

You can use this option multiple times on the same command line to specify multiple solutions.

`-project <path>`

This option specifies the path to the project to be analyzed when the solution is not provided. You can use Ant-style wildcards to specify patterns. Example:

- `-project "C:\Devel\FooSolution\FooProject.csproj"`

You can use this option multiple times on the same command line to specify multiple projects.

`-website <path>`

This option specifies the path to the website directory to be analyzed when the solution is not provided. Example:

- `-website "C:\Devel\MyWebsite"`

You can use this option multiple times on the same command line to specify multiple resources.

`-solutionConfig`

This option specifies the solution configuration. Example

- `-solutionConfig Debug`

`-projectConfig`

This option specifies the solution configuration. Example

- `-projectConfig Debug`

`-targetPlatform`

This option specifies the target platform of the solution (or project) configuration. Examples:

- `-solution "C:\Devel\FooSolution\FooSolution.sln" -solutionConfig Debug -targetPlatform "Any CPU"`
- `-project "C:\Devel\FooSolution\FooProject.csproj" -projectConfig Debug -targetPlatform AnyCPU`

`-resource <path relative to the solution>`

This option narrows down the primary testing scope defined with the `-solution`, `-project`, and `-website` options. You can specify one of the following:

- a path to a project in a solution
- a path to a directory of files in a project
- a path to a file

You can use this option multiple times on the same command line to specify multiple resources. You can use Ant-style wildcards to specify the resource. Examples:

- `-resource "FooSolution/MyProject1"`
- `-resource "FooSolution/MyProject1/QuxDirectory"`
- `-resource "FooSolution/MyProject1/QuxDirectory/BazFile.cs"`

`-include <absolute path> and -exclude <absolute path>`

These options narrow down the primary testing scope defined with the `-solution`, `-project`, and `-website` options. You can use Ant-style wildcards to specify the resource. Example:

- `-exclude "C:/Devel/FooSolution/*.Tests/**/*.*"`
- `-include "C:/ProductX/Solution1/src"`

Alternatively, you can specify a list of patterns in a `.lst` file and pass the file with the `-include` or `-exclude` switch. Each line in the file will be treated as a single entry.

`-reference <path>`

This setting specifies the path to additional assemblies needed to resolve dependencies of the analyzed projects. Use this option if you receive an "Unable to find reference assembly" message. You can use Ant-style wildcards and relative paths to the current working directory. Example:

- `-reference C:\MySolution\ExternalAssemblies*.dll`
- `-reference C:\Solution1\Assemblies2018*.exe`
- `-reference C:\FooSolution\Ext***.dll`

See [Configuring the Test Scope](#) for details.

Reporting

`-report <path>`

This option specifies the path to the directory where the report will be created.

`-publish`

This option sends the results to DTP. See [Sending Results and Publishing Source Code to DTP](#).

Customizing Configuration

`-settings <path>`

This option specifies the path to a custom `.properties` file that includes customized settings in the following format: `key=value` (for example, `report.format=pdf`). You can use this option multiple times to specify several `.properties` files. Entries with the same key will be overwritten. Example:

- `-settings Project1Config.properties`

`-property <key>=<value>`

This option allows you to configure a single setting directly in the command line. Use the following format: `key=value`.

You can use this option multiple times to configure several settings on the same command line. Earlier entries with the same key will be overwritten. Examples:

- `-property session.tag=sa_linux -property report.dtp.publish=true -property techsupport.create.on.exit=true`

`-showsettings`

This option prints the current settings and customizations.

See [Configuration Settings](#) for the list of settings you can configure.

Collecting Coverage

`- staticCoverage`

This option specifies a path to the `coverage.xml` file that contains static coverage data.

`- runtimeCoverage`

This option specifies the path to the `runtime_coverage_[timestamp].data` file downloaded via CAM or generated by SOAtest. You can provide a path to a folder that contains many `.data` files from multiple testing sessions.

See [Application Coverage for Web Applications](#) for details.

Other Options

`-machineId`

This option prints your machine ID.

`-encodepass <your password>`

This option prints an encoded password that can be used in the `.properties` configuration file.

`-nobuild`

This option skips the building phase of a solution or project.

`-out <path>`

This option specifies the path to the folder where the console output is saved.

`-fail`

This option fails the command by returning a non-zero exit code if some findings are reported.

`-version`

This option prints information about the version of the Parasoft tool you are using.

`-help`

This option prints the command line help.