

MSTest.exe Command-Line Options

MSTest.exe is the command-line command that is used to run tests. This command has several options you can use to customize your test run. You can use many of these options in conjunction with one another; in fact, you must use certain options in conjunction with others, as described in the following sections. You can specify these options in any order on the MSTest.exe command line.

MSTest.exe does not interpret the options nor the values you specify for them in a case-sensitive manner.

The following tables list all the options for MSTest.exe and short descriptions of them. You can see a similar summary by typing **MSTest /h** at a command line.

General Command Line Options

/testcontainer: [<i>file name</i>]	Load a file that contains tests. Example: /testcontainer:tests.dll For more information, see /testcontainer .
/testmetadata: [<i>file name</i>]	Load a file that contains test metadata. For more information, see /testmetadata .
/testlist: [<i>test list path</i>]	Specify the test list, as specified in the metadata file, to be run. For more information, see /testlist .
/test: [<i>test name</i>]	Specify the name of a test to be run. For more information, see /test .
/runconfig: [<i>file name</i>]	Use the specified run configuration file. Example: /runconfig:localtestrun.Testrunconfig For more information, see /runconfig .
/resultsfile: [<i>file name</i>]	Save the test run results to the specified file. Example: /resultsfile:testResults.trx For more information, see /resultsfile .
/unique	Run the test only if one unique match is found for any specified /test . For more information, see /unique .
/detail: [<i>property id</i>]	Specify the name of a property that you want to show values for, if any, in addition to the test outcome. For more information, see /detail .
/help	Display the MSTest.exe usage message (short form: /? or /h).
/nologo	Display no startup banner and copyright message.

Command-Line Options for Publishing Test Results

For more information about these options, see [Command-Line Options for Publishing Test Results](#).

/publish: [<i>server name</i>]	Publish results to the operational store of the specified server.
/publishresultsfile: [<i>file name</i>]	Specify the results file name to be published. If no results file name is specified, use the file produced by the current run.
/publishbuild: [<i>build id</i>]	Publish test results using this build ID.
/teamproject: [<i>team project name</i>]	Specify the name of the team project to which the build belongs.
/platform: [<i>platform</i>]	Specify the platform of the build against which test results should be published.
/flavor: [<i>flavor</i>]	Specify the flavor of the build against which test results should be published.

Using MSTest Options

The following sections describe many of the options for MSTest.exe in more detail. The options used for publishing test results are not included here; for information about those options, see [Command-Line Options for Publishing Test Results](#).

/testcontainer
/testcontainer:[*file name*]

The test container is a file that contains the tests you want to run. For example, for ordered tests, the test container is the .orderedtest file that

defines the ordered test. For unit tests, it is the assembly built from the test project that contains the unit test source files.

Note

For unit tests, this is the assembly that contains test code, not the assembly that contains the code of the application you are testing. For example, if your solution contains a project named `BankAccount` and a corresponding test project named `BankAccountTest`, specify **`/testcontainer:BankAccountTest.dll`**.

Because a test assembly contains the tests for one test project, the number of test containers for a solution matches the number of test projects in the solution.

Note

Because the test metadata file also lists tests that you can run, you must not specify both the **`/testcontainer`** and **`/testmetadata`** options in a single command line. Doing this would be ambiguous and would produce an error.

`/testmetadata`

`/testmetadata:` *[file name]*

You can use the **`/testmetadata`** option to run tests in multiple test containers.

The test metadata file is created for your solution when you create test lists using the Test Manager window. This file contains information about all the tests listed in the Test Manager window. These are all the tests that exist in all test projects in your solution.

You can change the test metadata file only by making changes that are reflected in the Test Manager window, such as creating or deleting tests, or changing a test's properties.

Note

Because the test container contains tests that you can run, you must not specify both the **`/testcontainer`** and **`/testmetadata`** options in a single command line. Doing this would be ambiguous and would produce an error.

The test metadata file is an XML file that is created in the solution folder. This file is shown in Solution Explorer under the **Solution Items** node. A test metadata file has the extension `.vsmdi`, and is associated with the Test Manager window; that is, if you double-click a `.vsmdi` file in Windows Explorer, the file opens Visual Studio, and its contents. All the tests in a solution's test projects are displayed in the Test Manager window.

When you use the **`/testmetadata`** option, it is recommended that you indicate specific tests to run by using the **`/test`** option or the **`/testlist`** option, or both.

`/testlist`

`/testlist:` *[test list path]*

The **`/testlist`** option is a list of tests, as specified in the test metadata file, to be run. To run the tests contained in multiple test lists, use the **`/testlist`** option multiple times.

Note

You can use the **`/testlist`** option only if you also use the **`/testmetadata`** option.

You can use the **`/testlist`** option and the **`/test`** option together. This is equivalent to selecting both a test list and one or more individual tests in the Test Manager window and then clicking **Run Tests**.

`/test`

`/test:` *[test name]*

Use the **`/test`** option to specify individual tests to run. To run multiple tests, use the **`/test`** option multiple times.

Note

You can use the **`/test`** option with either the **`/testcontainer`** option or with the **`/testmetadata`** option, but not with both.

You can use the **`/testlist`** option and the **`/test`** option together. This is equivalent to selecting both a test list and one or more individual tests in the Test Manager window and then clicking **Run Tests**.

The string you specify with the **`/test`** option is used to match the names of tests in either a test container or a test metadata file. This means that you can specify multiple tests by using a single value for **`/test`**. For example, specifying **`/test:ittest`** would produce matches for tests named `DebitTest` and `CreditTest` because both test names contain the substring 'ittest'.

Note

The value you specify with the **`/test`** option is tested against not only the name of the test, but also the path to that test, as seen in Solution Explorer, or, with unit tests, to their fully-qualified name.

Two usage examples follow:

Unit-test example: The file UnitTest1.cs in the project TestProject2 contains a unit test named TestMethod1. Specifying a value of 'ittest' for the /test option would also match this test, because the string is tested against the fully-qualified name "TestProject2.UnitTest1.TestMethod1" and the string 'ittest' also appears in 'UnitTest1'.

Generic-test example: The following command line runs the specified generic test and, in the test results, displays the full path to the test.

```
mstest /testcontainer:"C:\Documents and Settings\\My Documents\Visual Studio\Projects  
\TestProject2\TestProject2\generictest1.generic" /test:testproject32\generic
```

/runconfig

/runconfig:[file name]

Use this option to specify a run configuration file. For example: **/runconfig:localtestrun.Testrunconfig**

You can specify a run configuration file in other ways, such as with the /testmetadata option. The rules that govern the specification of run configuration files are described here.

- If you use the **/runconfig** option, the file it specifies will be used, whether or not you also use the **/testmetadata** option.
- If you use the **/testmetadata** option to point to a metadata file that specifies the active run configuration file, that run configuration file will be used if you do not use the **/runconfig** option.
- If you do not use the **/runconfig** option and you also do not specify a run configuration file in the test metadata file, the test run uses the default run configuration file.

/resultsfile

/resultsfile:[file name]

Use this option to save the test run results to the named file. For example: /resultsfile:testResults.trx.

/unique

/unique

Use the **/unique** option in conjunction with the **/test** option. The **/unique** option instructs MSTest.exe to run only a single test whose name matches the value you supply with the **/test** option.

For example, the test container MyTestProject contains tests named MethodTest1 and MethodTest10.

The command line:

```
mstest /testcontainer:testproject2.dll /test:MethodTest1
```

runs both tests, MethodTest1 and MethodTest10, because 'MethodTest1' is a substring of 'MethodTest10'.

But the command line:

```
mstest /testcontainer:testproject2.dll /test:MethodTest1 /unique
```

runs only the test MethodTest1.

/detail

/detail:[property id]

This option is used to display additional test case properties, if they exist. You can pass more than one instance of the **/detail** option, each with only one property id, in single command line. Valid property id's for the **/detail** option follow:

adapter	id	projectrelativepath
computername	isautomated	readonly
debugtrace	link	spoolmessage
description	longtext	stderr
displaytext	name	stdout
duration	outcometext	storage
errormessage	owner	testcategoryid
errorstacktrace	parentexecid	testname
executionid	priority	testtype
groups	projectname	traceinfo



Note

The actual selection of property IDs you can use with the /detail option varies according to test type. Therefore, this list is only an approximation. In particular, if you are using test types other than those that are included in the Team System testing tools, the selection of properties will be different. To know which propertyIDs you can use, examine the test results file produced by the test run. For more information about test results files, see [How to: Export Test Results](#).

If a property exists for specified test case, its information is included in the output result summary.

For example, the command line:

mstest /testcontainer:Errors.dll /detail:testtype

produces the following output, that contains test type information:

```
Results      ...
-----
Inconclusive
[testtype] = Unit Test
...

Top Level Tests
-----
TestProject2.BankAccountTest.CreditTest
```