

Visual Studio Tests

The Visual Studio Tests runner integrates [MSTest](#) runner and [VSTest console runner](#). Support for both frameworks enables TeamCity to execute tests and automatically import their test results.

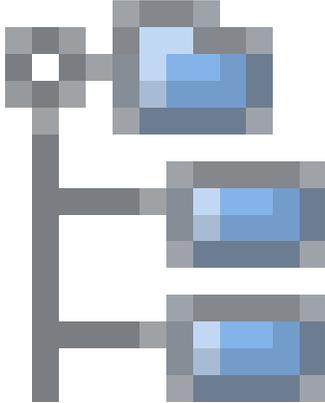
The Visual Studio Test Runner requires Visual Studio Test Agent or Microsoft Visual Studio installed on the build agent.

On this page:

- [Visual Studio Tests runner settings](#)
 - [VSTest Settings](#)
 - [Custom test logger](#)
 - [MSTest settings](#)

Visual Studio Tests runner settings

Option	Description
Test engine type	Select the tool used to run tests: VSTest or MSTest.
Test engine version	Select the version of the tool from the drop-down. By default, the available VSTest and MSTest installations are autodetected by TeamCity: <ul style="list-style-type: none">▪ MSTest versions 2005-2015, and since TeamCity 10.0.4 - 2017 are supported.▪ VSTest versions 2012-2015, and since TeamCity 10.0.4 - 2017 are supported. You can specify a custom path to the test runner here as well. TeamCity parameters are supported. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> You can specify the agent property <code>teamcity.dotnet.vstest.VS_VERSION.install.dir</code> pointing to a VSTest distribution path and use it here, where <code>VS_VERSION</code> can be 11.0, 12.0,14.0. or since TeamCity 10.0.4 - 15.0. The property will have priority over registry values.</div>
Test file names	This field is mandatory for VSTest and optional for MSTest. Specify the new-line separated list of paths to assemblies to run tests on in the included assemblies list. Exclude assemblies from the test run by specifying paths to them in the corresponding field. Wildcards are supported.  Paths to the assemblies must be relative to the build checkout directory .

Run configuration file	<p>(Optional) Specify the typical <code>.runsettings</code> file for VSTest or <code>.testsettings</code> file for MSTest. You can use</p>  <p>the file browser</p>
Additional command line parameters	<p>Enter additional command line parameters for the selected test engine.</p> <p>i Note that tests run with MSTest are not reported on-the-fly. The Microsoft Developer Network lists the available options for VSTest and MSTest.</p>

The rest of settings will vary depending on the engine to run tests with:

- [VSTest Settings](#)
- [MSTest settings](#)

VSTest Settings

Option	Description
Target platform	Select the platform bitness. Note that specifying x64 target platform will force the <code>vstest.console</code> process to be run in the isolated mode
Framework	If the default is specified, <code>vstest.console</code> will select the target framework automatically. You can also choose the .Net platform manually using the drop-down.
Test names	<p>(optional) Of all tests discovered in the included assemblies, only the tests with the names matching the provided values will be run. For multiple values, use new line.</p> <p>If the field is empty, all tests will be run. See details in the Microsoft documentation. ! Cannot be used with the option below.</p>
Test case filter	Run tests that match the given expression. See details in the Microsoft documentation . ! Cannot be used with the option above.
Run in isolation	Run the tests in an isolated process

Use real-time test reporting	<ul style="list-style-type: none"> • If this checkbox is selected, a custom TeamCity test logger will be used for real-time reporting. See the next section for details. • If not selected, VSTest will report tests to TeamCity after the run (the discovery of custom logger will not be attempted). <div style="border: 1px solid #f0e68c; padding: 5px; margin-top: 10px;">  When using this option, it is recommended that you check the number of tests in the produced <code>trx</code> file against the build log. In case of inconsistencies, switch to the custom logger. </div>
------------------------------	---

Custom test logger

VSTest.Console supports custom loggers, i.e. libraries that can handle events that occur when tests are being executed. TeamCity 9.0+ has a custom logger that provides real-time test reporting. The logger must be installed manually on the agent machine, as it requires dlls to be copied to the `Extensions` folder of the VSTest.Console. No agent restart is needed when the custom logger is installed.

To install the custom logger:

- a. Download the [custom logger](#)
- b. Extract the contents of the downloaded archive on the agent machine:
 - for VisualStudio 2017 update 5 onwards - to `PROGRAM_FILES(x86)\Microsoft Visual Studio\2017\<Edition>\Common7\IDE\Extensions\TestPlatform\Extensions`
 - for VisualStudio 2017 up to update 4 - to `PROGRAM_FILES(x86)\Microsoft Visual Studio\2017\<Edition>\Common7\IDE\CommonExtensions\Microsoft\TestWindow\Extensions`
 - for VisualStudio 2015 - to `PROGRAM_FILES\Microsoft Visual Studio 14.0\Common7\IDE\CommonExtensions\Microsoft\TestWindow\Extensions`
 - for VisualStudio 2013 - to `PROGRAM_FILES\Microsoft Visual Studio 12.0\Common7\IDE\CommonExtensions\Microsoft\TestWindow\Extensions`
 - for VisualStudio 2012 - to `PROGRAM_FILES\Microsoft Visual Studio 11.0\Common7\IDE\CommonExtensions\Microsoft\TestWindow\Extensions`
- Check that the custom logger was installed correctly by executing `vstest.console.exe /ListLoggers` in the console on the agent machine. If the logger was installed correctly, you will see the logger with FriendlyName TeamCity listed:

```
VSTest.TeamCityLogger.TeamCityLogger
Uri: logger://TeamCityLogger
FriendlyName: TeamCity{info}
```

MSTest settings

Option	Description
MSTest metadata	Enter a value for <code>/testmetadata:file</code> argument. See details in the Microsoft documentation
Testlist from metadata to run	Edit the Testlist. Every line will be translated into <code>/testlist:line</code> argument. See details in the Microsoft documentation .
Test	Names of individual tests to run. This option will be translated into the series of <code>/test:</code> arguments. See details in the Microsoft documentation .
Unique	Run the test only if one unique match is found for any specified test in test section

Results file

Enter a value for the `/resultsfile:file` name command-line argument. Parameter references are supported. It is recommended to leave the field blank to generate a temporary `*.trx` file in the build temporary directory. Note that the directory may be cleaned between the builds.

To save the test run results to a named non-default file, enter a value for the `/resultsfile:file` name command-line argument.

- If the path specified is relative, it will be relative to the build checkout directory. If the specified file already exists in the checkout directory, the build agent will attempt to delete the file. If the agent fails to delete it, the build will fail.



If you need to generate your results file in the checkout directory, consider adding the [Swabra](#) build feature to your configuration.

To create a `*.trx` report in the build temporary directory, use `%system.teamcity.build.tempDir%`.

- If the path specified is absolute, it will be used "as is".