Running selective unit tests

2 minutes to read • +3

With the `dotnet test` command in .NET Core, you can use a filter expression to run selective tests. This article demonstrates how to filter which test are run. The following examples use `dotnet test`. If you're using `vstest.console.exe`, replace `--filter` with `--testcasefilter:`.

Note

Using filters that include exclamation mark (!) on `*nix` requires escaping since `!` is reserved. For example, this filter skips all tests if the namespace contains IntegrationTests: `dotnet test --filter FullyQualifiedName\!~IntegrationTests`. Note the backslash that precedes the exclamation mark.

MSTest

```
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace MSTestNamespace
{
    [TestClass]
    public class UnitTest1
    {
        [TestCategory("CategoryA")]
        [Priority(1)]
        [TestMethod]
        public void TestMethod1()
        {
        }

        [Priority(2)]
        [TestMethod]
        public void TestMethod2()
        {
        }
    }
}
```

Expression	Result
`dotnet test --filter Method`	Runs tests whose `FullyQualifiedName`

Expression	Result
	contains `Method`. Available in `vstest 15.1+`.
`dotnet test --filter Name~TestMethod1`	Runs tests whose name contains `TestMethod1`.
`dotnet test --filter ClassName=MSTestNamespace.UnitTest1`	Runs tests which are in class `MSTestNamespace.UnitTest1`. **Note:** The `ClassName` value should have a namespace, so `ClassName=UnitTest1` won't work.
`dotnet test --filter FullyQualifiedName!=MSTestNamespace.UnitTest1.TestMethod1`	Runs all tests except `MSTestNamespace.UnitTest1.TestMethod1`.
`dotnet test --filter TestCategory=CategoryA`	Runs tests which are annotated with `[TestCategory("CategoryA")]`.
`dotnet test --filter Priority=2`	Runs tests which are annotated with `[Priority(2)]`.

Using conditional operators | and &

Expression	Result
`dotnet test --filter "FullyQualifiedName~UnitTest1|TestCategory=CategoryA"`	Runs tests which have `UnitTest1` in `FullyQualifiedName` **or** `TestCategory` is `CategoryA`.
`dotnet test --filter "FullyQualifiedName~UnitTest1&TestCategory=CategoryA"`	Runs tests which have `UnitTest1` in `FullyQualifiedName` **and** `TestCategory` is `CategoryA`.
`dotnet test --filter "(FullyQualifiedName~UnitTest1&TestCategory=CategoryA)|Priority=1"`	Runs tests which have either `FullyQualifiedName` containing `UnitTest1` **and** `TestCategory` is `CategoryA` **or** `Priority` is 1.

xUnit

```
using Xunit;

namespace XUnitNamespace
{
    public class TestClass1
    {
        [Trait("Category", "CategoryA")]
        [Trait("Priority", "1")]
        [Fact]
        public void Test1()
        {
        }

        [Trait("Priority", "2")]
        [Fact]
        public void Test2()
        {
        }
    }
}
```

Expression	Result
`dotnet test --filter DisplayName=XUnitNamespace.TestClass1.Test1`	Runs only one test, `XUnitNamespace.TestClass1.Test1`.
`dotnet test --filter FullyQualifiedName!=XUnitNamespace.TestClass1.Test1`	Runs all tests except `XUnitNamespace.TestClass1.Test1`.
`dotnet test --filter DisplayName~TestClass1`	Runs tests whose display name contains `TestClass1`.

In the code example, the defined traits with keys `Category` and `Priority` can be used for filtering.

Expression	Result
`dotnet test --filter XUnit`	Runs tests whose `FullyQualifiedName` contains `XUnit`. Available in `vstest 15.1+`.
`dotnet test --filter Category=CategoryA`	Runs tests which have `[Trait("Category", "CategoryA")]`.

Using conditional operators | and &

Expression	Result
`dotnet test --filter "FullyQualifiedName~TestClass1|Category=CategoryA"`	Runs tests which has `TestClass1` in

Expression	Result
`dotnet test --filter "FullyQualifiedName~TestClass1&Category=CategoryA"`	`FullyQualifiedName` **or** `Category` is `CategoryA`. Runs tests which has `TestClass1` in `FullyQualifiedName` **and** `Category` is `CategoryA`.
`dotnet test --filter "(FullyQualifiedName~TestClass1&Category=CategoryA)|Priority=1"`	Runs tests which have either `FullyQualifiedName` containing `TestClass1` **and** `Category` is `CategoryA` **or** `Priority` is 1.

NUnit

```
using NUnit.Framework;

namespace NUnitNamespace
{
    public class UnitTest1
    {
        [Category("CategoryA")]
        [Property("Priority", 1)]
        [Test]
        public void TestMethod1()
        {
        }

        [Property("Priority", 2)]
        [Test]
        public void TestMethod2()
        {
        }
    }
}
```

Expression	Result
`dotnet test --filter Method`	Runs tests whose `FullyQualifiedName` contains `Method`. Available in `vstest 15.1+`.

Expression	Result
`dotnet test --filter Name~TestMethod1`	Runs tests whose name contains `TestMethod1`.
`dotnet test --filter FullyQualifiedName~NUnitNamespace.UnitTest1`	Runs tests which are in class `NUnitNamespace.UnitTest1`.
`dotnet test --filter FullyQualifiedName!=NUnitNamespace.UnitTest1.TestMethod1`	Runs all tests except `NUnitNamespace.UnitTest1.TestMethod1`.
`dotnet test --filter TestCategory=CategoryA`	Runs tests which are annotated with `[Category("CategoryA")]`.
`dotnet test --filter Priority=2`	Runs tests which are annotated with `[Priority(2)]`.

Using conditional operators | and &

Expression	Result
`dotnet test --filter "FullyQualifiedName~UnitTest1|TestCategory=CategoryA"`	Runs tests which have `UnitTest1` in `FullyQualifiedName` **or** `TestCategory` is `CategoryA`.
`dotnet test --filter "FullyQualifiedName~UnitTest1&TestCategory=CategoryA"`	Runs tests which have `UnitTest1` in `FullyQualifiedName` **and** `TestCategory` is `CategoryA`.
`dotnet test --filter "(FullyQualifiedName~UnitTest1&TestCategory=CategoryA)|Priority=1"`	Runs tests which have either `FullyQualifiedName` containing `UnitTest1` **and** `TestCategory` is `CategoryA` **or** `Priority` is 1.