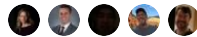


Create unit test method stubs with the Create Unit Tests command

03/31/2019 • 2 minutes to read •  +3

In this article

[Get started](#)

[Set unit test traits](#)

[Use third-party unit test frameworks](#)

[When should I use this feature?](#)

[See also](#)

The **Create Unit Tests** command creates unit test method stubs. This feature allows easy configuration of a test project, the test class, and the test method stub within it.

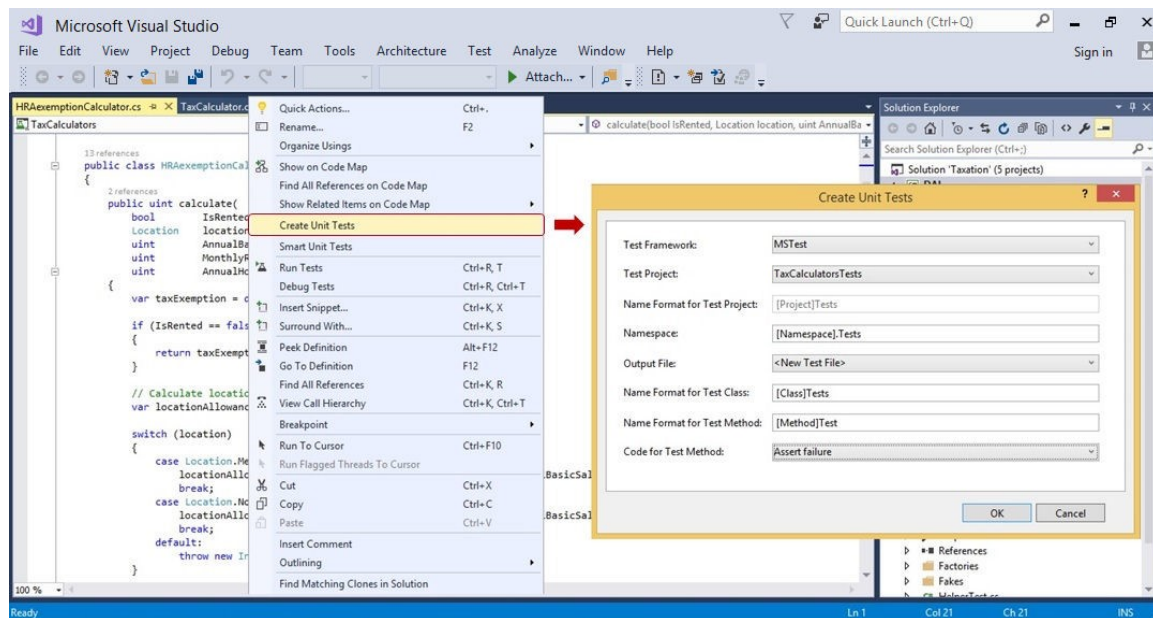
Note

The **Create Unit Tests** menu command is only available for managed code that targets .NET Framework (but not .NET Core).

The **Create Unit Tests** menu command is extensible and can be used to generate tests for MSTest, MSTest V2, NUnit, and xUnit.

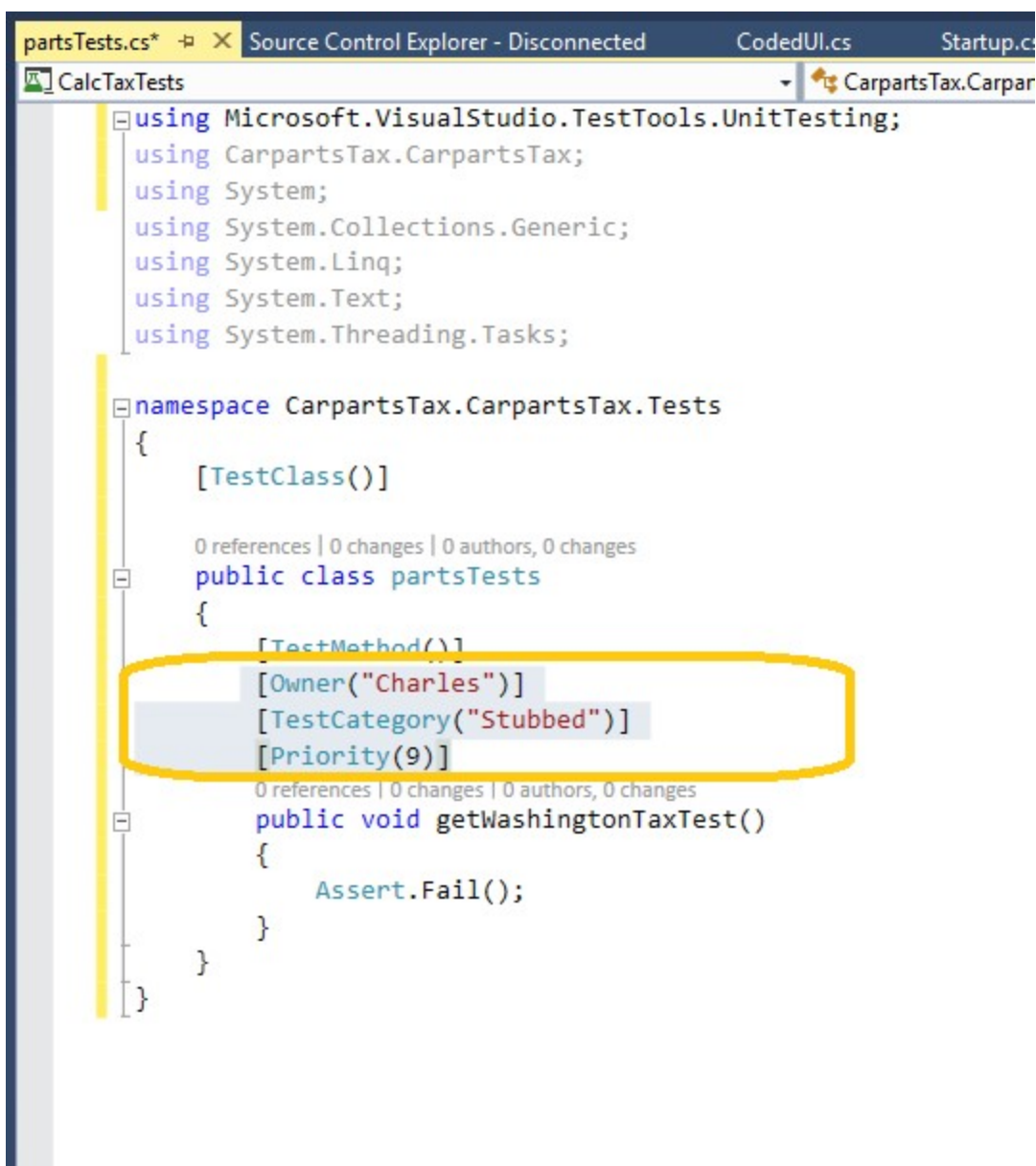
Get started

To get started, select a method, a type, or a namespace in the code editor in the project you want to test, right-click, and then choose **Create Unit Tests**. The **Create Unit Tests** dialog opens where you can configure how you want the tests to be created.



Set unit test traits

If you plan to run these tests as part of the test automation process, you might consider having the test created in another test project (the second option in the dialog above) and setting unit test traits for the unit test. This enables you to more easily include or exclude these specific tests as part of a continuous integration or continuous deployment pipeline. The traits are set by adding metadata to the unit test directly, as shown below.



Use third-party unit test frameworks

To automatically generate unit tests for NUnit or xUnit, install one of these test framework extensions from Visual Studio Marketplace:

- [NUnit extension for test generators](#)
- [xUnit.net extension for test generators](#)

When should I use this feature?

Use this feature whenever you need to create unit tests, but specifically when you are testing existing code that has little or no test coverage and no documentation. In other words, where there is limited or non-existent code specification. It effectively implements an approach similar to [Smart unit tests](#) that characterizes the observed behavior of the code.

However, this feature is equally applicable when a developer starts by writing some code and then uses that to bootstrap unit tests. Within the flow of coding, the developer might want to quickly create a unit test method stub (with a suitable test class and a suitable test project) for a particular piece of code.

See also

- [Creating unit test method stubs with "Create Unit Tests"](#)
- [Unit testing blog posts](#)