**NUNIT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Affected Section** | **Author** |
| 1.0 | 25/03/2019 | Initial Draft | Matthew Mc Colgan |

Contents

[Introduction 2](#_Toc4621179)

[Installation 2](#_Toc4621180)

[Local 2](#_Toc4621181)

[Build Environment 3](#_Toc4621182)

[Sample 3](#_Toc4621183)

[References 4](#_Toc4621184)

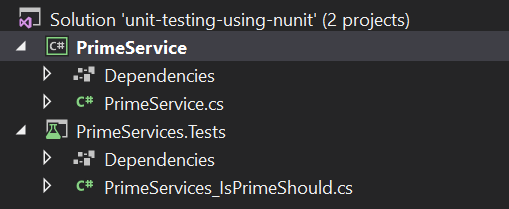
# Introduction

NUnit is an open-source unit-testing framework for all .NET languages. For the CrowSoft project, NUnit must be installed on our local user machines for writing the unit tests and in the build environment and build sandbox environment for automated testing.

# Installation

## Local

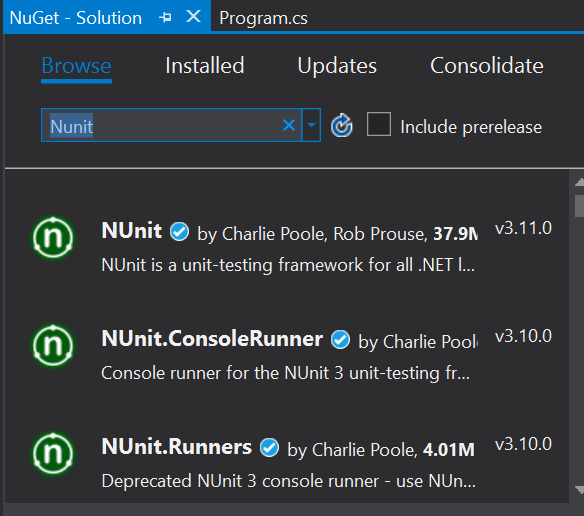
Open your project in visual studio



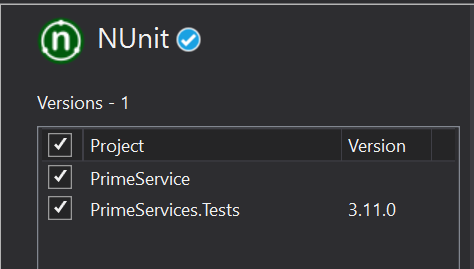
Right click on the solution tab and find “Manage NuGet Packages for Solution”



Under “Browse” you can search for the specific package, in this instance NUnit.



From here select the “NUnit” package. A pane on the right will open with the available projects to install it on.



Select your project.

The latest version will automatically be chosen and click install



You will have to do the same as the step above for “NUnit3TestAdapter”

## Build Environment

Nunit is included when installing .NET libraries. When creating an NUnit test project, all that is required is a reference to the necessary nunit packages e.g.

<ItemGroup>

<PackageReference Include="nunit" Version="3.10.1" />

<PackageReference Include="NUnit3TestAdapter" Version="3.10.0" />

<PackageReference Include="Microsoft.NET.Test.Sdk" Version="15.8.0" />

</ItemGroup>

The sample above is created in the *AppName*.Tests.csproj file. This is a reference to the test SDK, NUnit test framework and the NUnit runner. The test runner picks up a source code directory that contains tests and executes them. TestRunner is a CMD tool, though, with the reference to the NUnit3TestAdapter in the project file, Jenkins will be able to run the tests directly.

After a check-in is made to GitHub, Jenkins will pull the code, build it, run it and with .NET installed on the build server Jenkins will have a reference to the required packages and libraries to run the tests.

# Sample

Below is a sample of a couple of test cases.

[Test]

public void ReturnFalseGivenValueOf1()

{

var result = \_primeService.IsPrime(1);

Assert.IsFalse(result, "1 should not be prime");

}

[TestCase(-1)]

[TestCase(0)]

[TestCase(1)]

public void ReturnFalseGivenValuesLessThan2(int value)

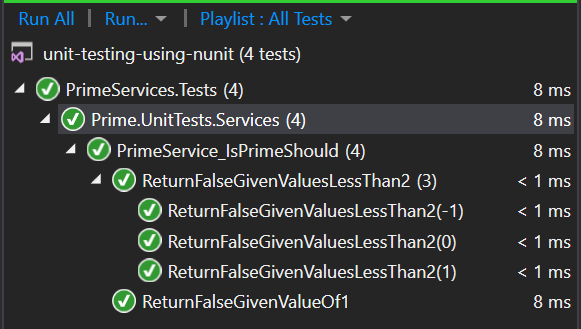
{

var result = \_primeService.IsPrime(value);

Assert.IsFalse(result, $"{value} should not be prime");

}

To follow along with the sample provided check out <https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-nunit>. It is a Microsoft unit testing example which is very easy to follow.



I was able to create the project using command line in linux or windows. You can run the tests there. I then opened the project in Visual Studio and ran the suite of tests from VS also.

# References

<https://marketplace.visualstudio.com/items?itemName=NUnitDevelopers.NUnit3TestAdapter>

<https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-nunit>

<https://www.quora.com/What-is-a-test-runner>