**Hands-on .NetCore 3.0**

**NameLength & Debuging**

Table of Contents

[Scaffolding Console App Basics 2](#_Toc23591534)

[NameLength Console App 2](#_Toc23591535)

[Creating the console app 2](#_Toc23591536)

[Writing the function to count characters 3](#_Toc23591537)

[Build the App 3](#_Toc23591538)

[Publish the app 3](#_Toc23591539)

[Running the app 4](#_Toc23591540)

[NameLength VS Console App 4](#_Toc23591541)

[Creating App 4](#_Toc23591542)

[Working with the same code 4](#_Toc23591543)

[Build and Publish 5](#_Toc23591544)

# Scaffolding Console App Basics

We have already run the commands multiple times, here is the info

* dotnet new console
* dotnet restore [pulls in the dependencies needed by the application]
* dotnet run [compiles and run the application]
* dotnet build [compiles the application]
* dotnet publish [packages up the files for reuse]

Take a look at this document for more details:

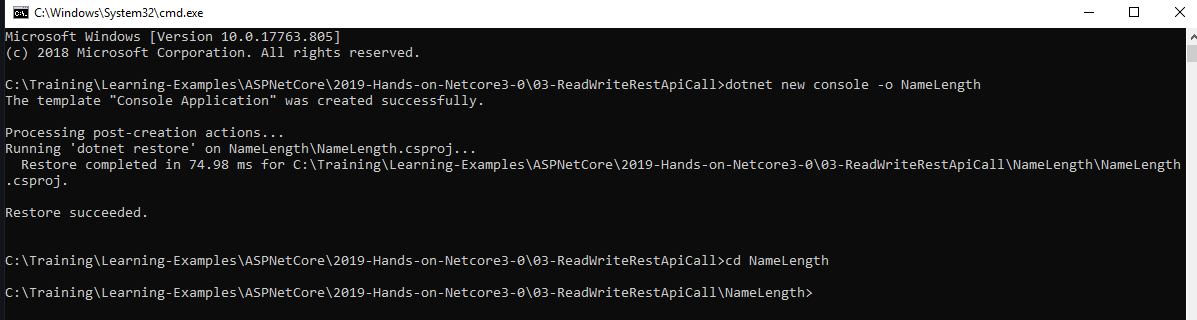
<https://itplate.blogspot.com/2019/11/scaffolding-applications-with-net-cli.html>

# NameLength Console App

## Creating the console app

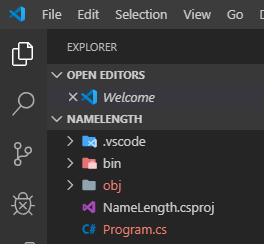
Run command **dotnet new console -o NameLength**

It has done the restore for us as well



Open the app with VS Code by

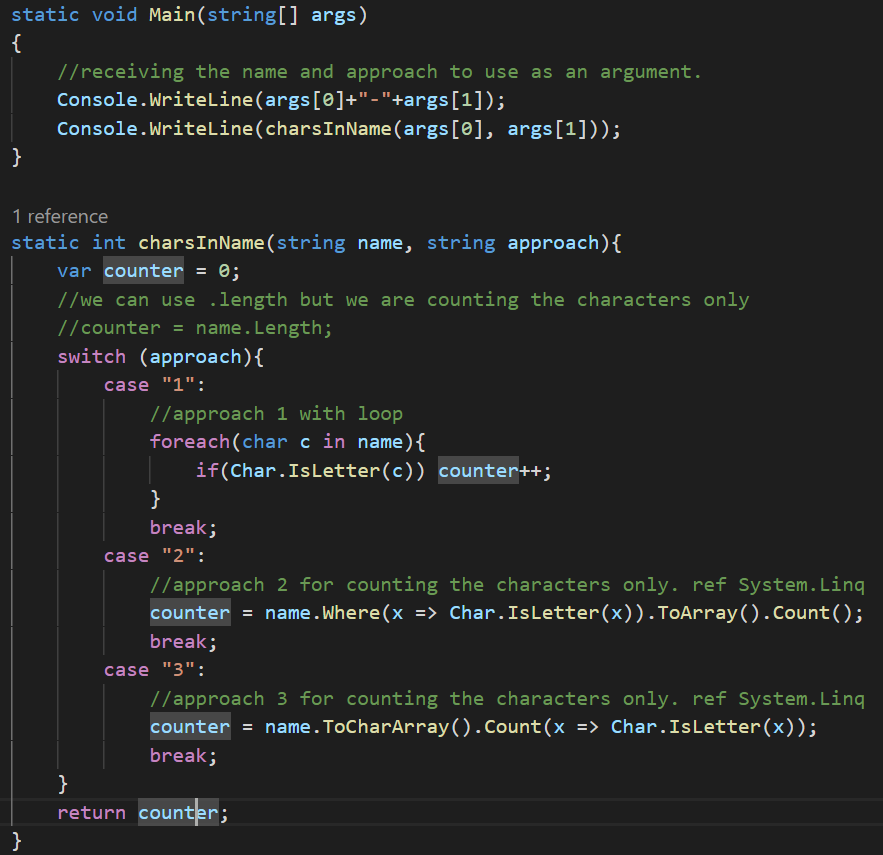
1. **cd NameLength**
2. and then typing **code .** [code space dot]



## Writing the function to count characters

We are here counting the characters in the name, which we’ll pass through the command line arguments when running the console app. There are three approaches for counting, so the second param will be the approach.

1. Write a static function charsInName
2. Call the function in Main



## Build the App

**dotnet build**

## Publish the app

Since I am using win10-x64, I’ll need to publish the app to win10.

**dotnet publish -c Release -r win10-x64**



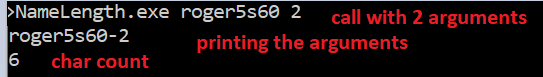
All .net core related dlls will get published in the above folder. Make sure to look inside \bin\Release and not \bin\Debug folder.

Take a look at the following for more options:

<https://itplate.blogspot.com/2019/11/scaffolding-applications-with-net-cli.html>

## Running the app

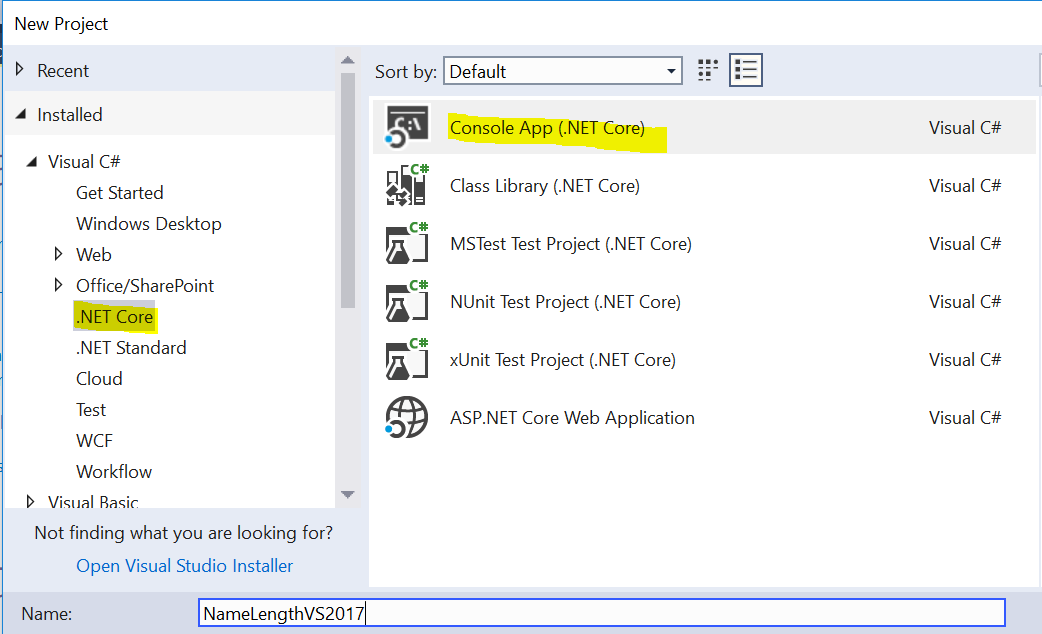
1. Switch to the bin**\Release\netcoreapp3.0\win10-x64** folder
2. Run **NameLength.exe**



# NameLength VS Console App

## Creating App

We’ll create the same console app with VS 2017



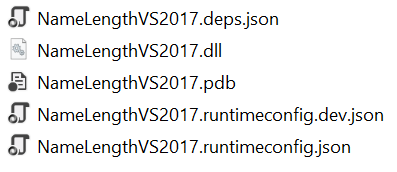
## Working with the same code

Move code to the program.cs file in the root.

## Build and Publish

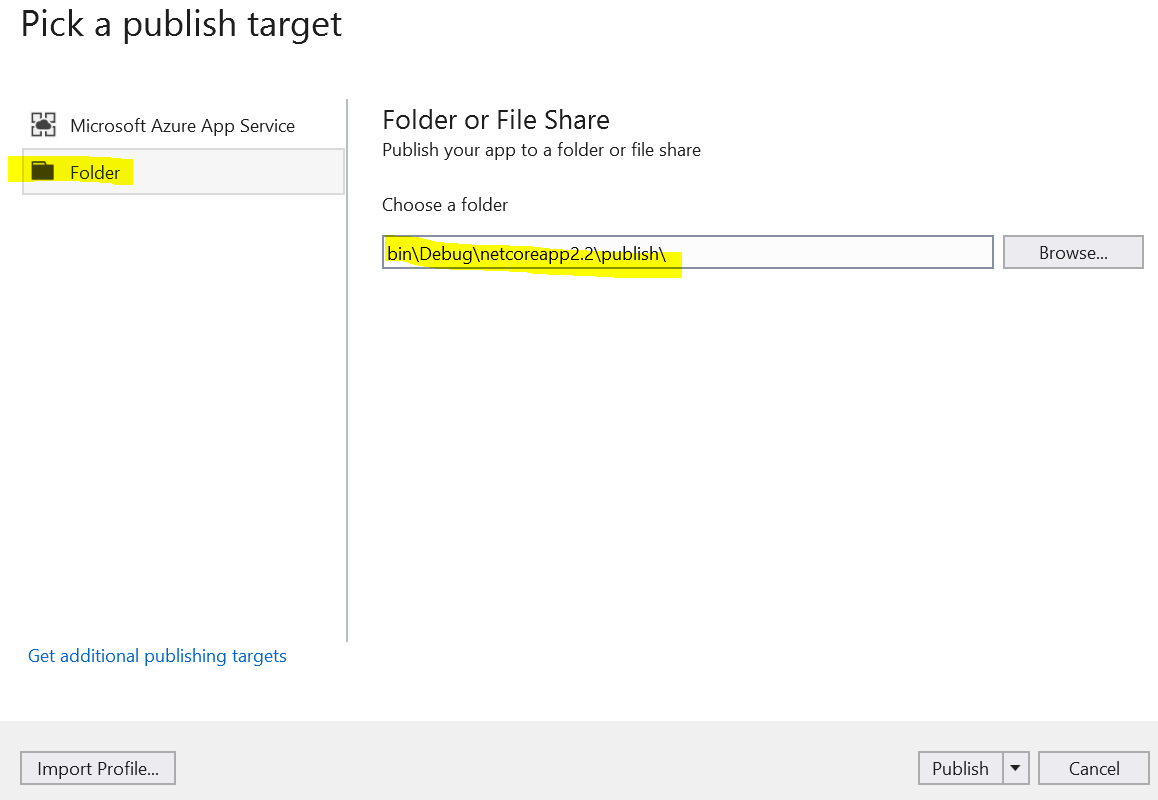
Build the application and take a look at the content of the \bin\netcore2.2 folder. IMP: VS2017 cannot create core 3 app.

C: \NameLengthVS2017\bin\Debug\netcoreapp2.2

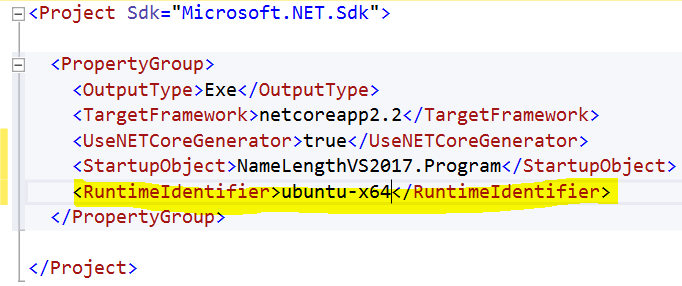


To run it go to the above folder in command prompt and type NameLengthVS2017.dll with params.

To make a windows release, right click the project and click Publish



To add ubuntu, right click the project and then click edit csproj file to add a new RunTimeIdentifier



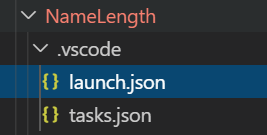
Again build and publish the project. You’ll see a new ubuntu folder in both debug and release folders.

# Debugging

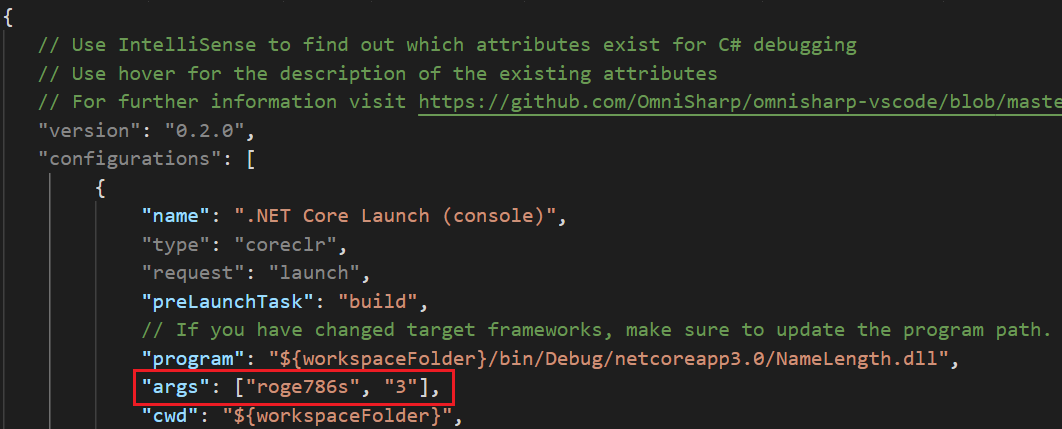
* F5 : continue
* F10 : step over
* F11 : step into
* SHIFT+F11 : step out
* SHIFT+F5 : stop
* F9: to put a break point

## Debugging in VS Code

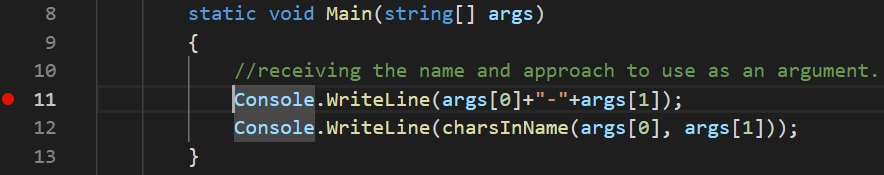
Expand .vscode folder and then open launch.json



Add the parameters to the args element that we have been passing before through command line



Then press F9 to put a break point at the start of the application



And then press F5 to debug

And after that press F10 to step over and F11 to step into.