## Getting Started with the Tabular Object Model (TOM)

The purpose of these hands-on lab exercises is to provide campers with experience programming the Tabular Object Model (TOM).

We have had .NET Framework and .NET Core which have been seen as two different platforms. The new version of .NET aims to unify these different platforms into a single platform. In this lab we will refer to the new unified runtime as .NET 5 just to make things more clear. Note that Microsoft will begin referring to the new runtime as simply .NET instead of .NET 5..

This lab was inspired by a set of blog posts by Phil Seamark.

### Setup: Install Visual Studio Code and the .NET 5 SDK

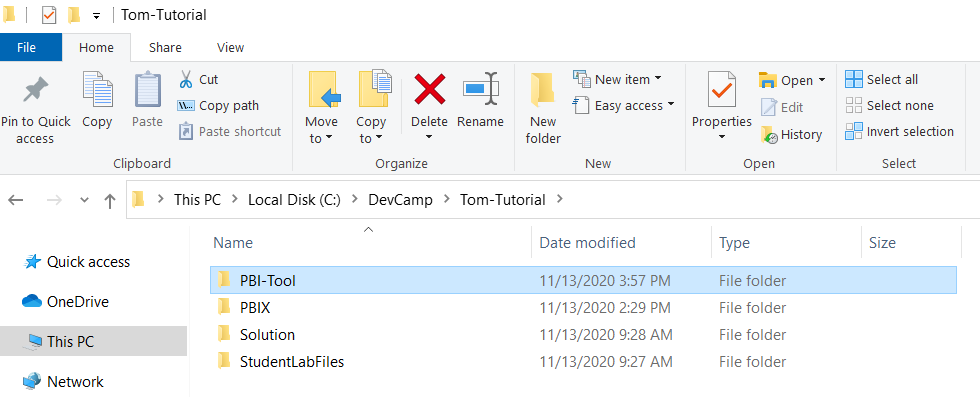
In this exercise, you will download xx.

1. Install the .NET 5 SDK.
   1. <https://dotnet.microsoft.com/download/dotnet/thank-you/sdk-5.0.100-windows-x64-installer>
   2. More info: <https://dotnet.microsoft.com/download>
2. Install Visual Studio Code
   1. Open a browser and navigate to [code.visualstudio.com](https://code.visualstudio.com/)
   2. download and run the installer for the current version for Windows.
   3. Use default settings when prompted during the install.

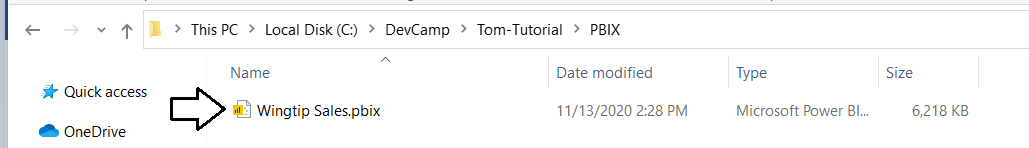
### Exercise 1: xxx

In this exercise, you will download xx.

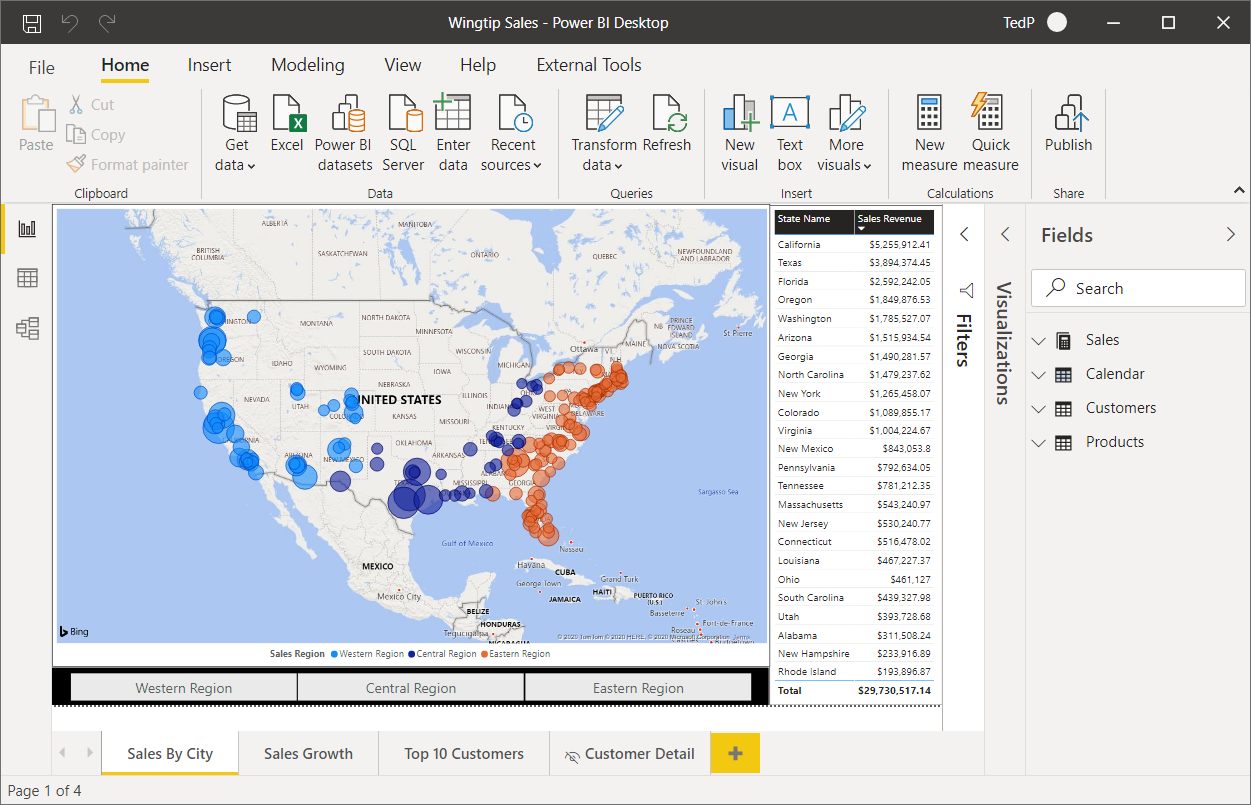
1. Create a new folder for the tutorial.
   1. Create a new folder on your local hard drive named **Tom-Tutorial**.
   2. Create a child inside the **Tom-Tutorial** folder named **PBI-Tool**.



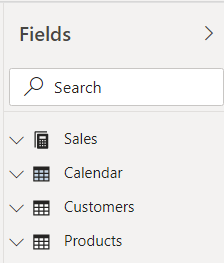
1. Locate Wingtip Sales.pbix and open it in Power BI Desktop.
   1. Aaaa



* 1. dddd

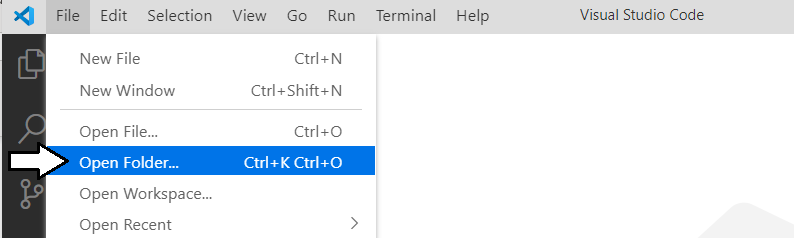


* 1. xxxx

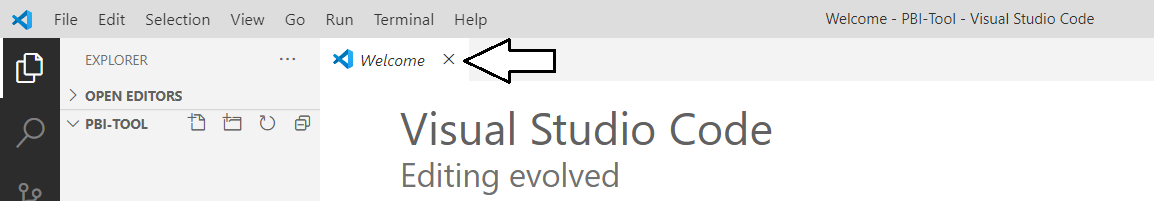


Now you will leave this project open in Power BI desktop. As you will see over the next few steps, you can access the **Wingtip Sales** data model with the Tabular Object Model by connecting through a localhost address with a random port number selected by Power BI Desktop whenever

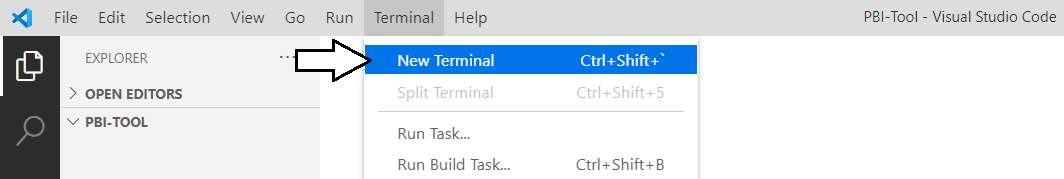
1. Launch Visual Studio Code and open the **PBI-Tool** folder.
   1. Sss



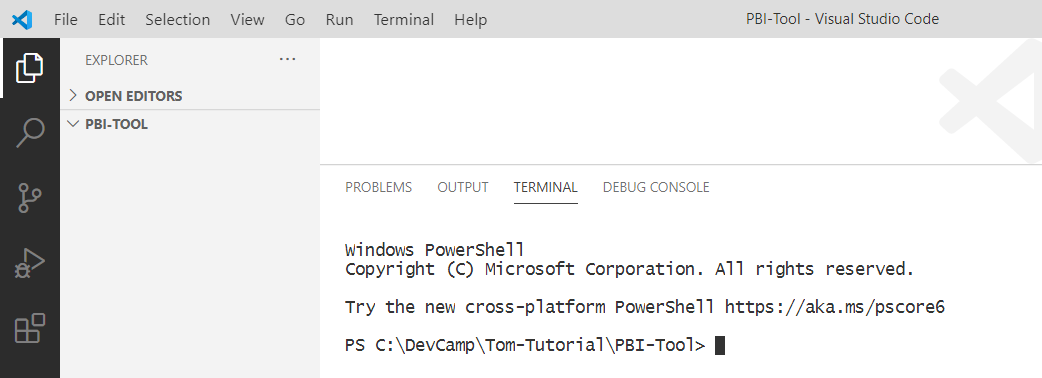
* 1. Close the start page



1. Verify the current version of .NET
   1. Open terminal.



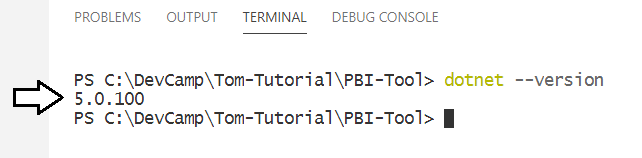
* 1. dddd



* 1. Run this command.

dotnet –version

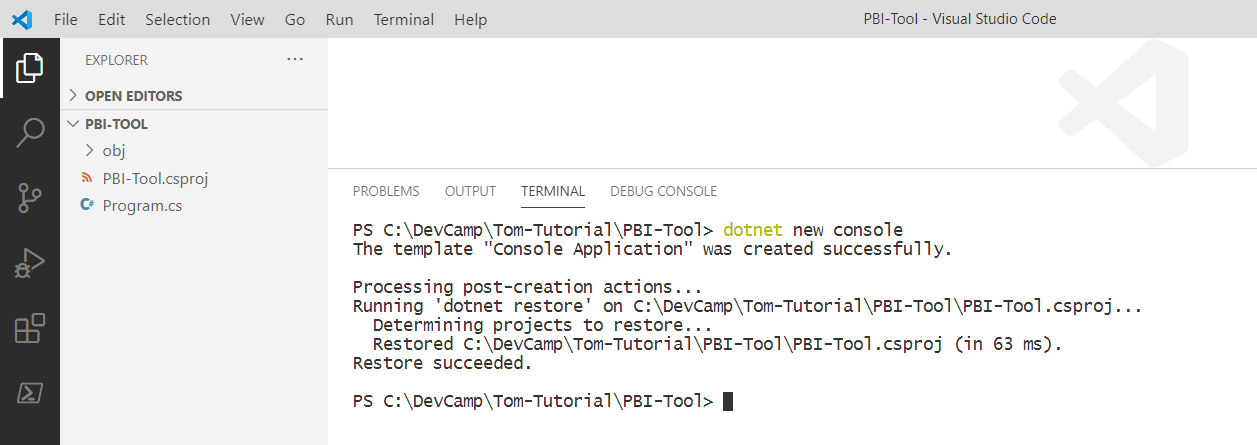
* 1. You should see the version number.



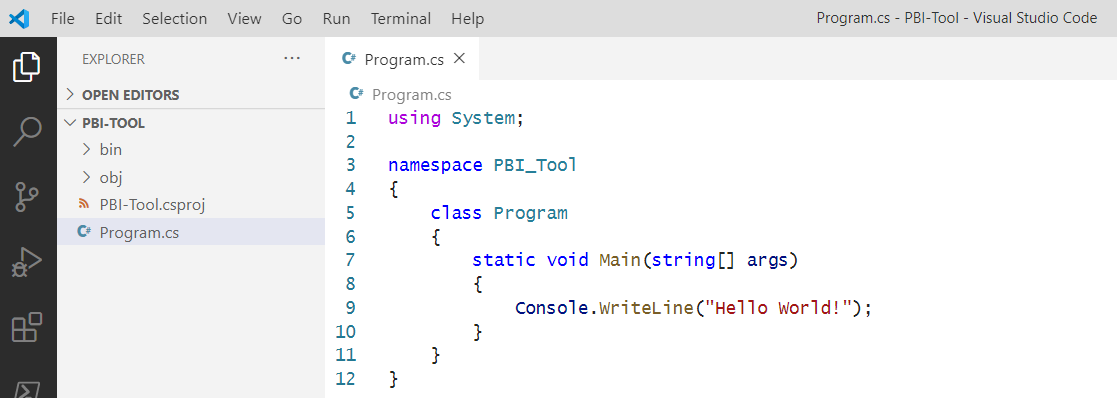
1. Create new folder and navigate to it.
   1. Xx

dotnet new console

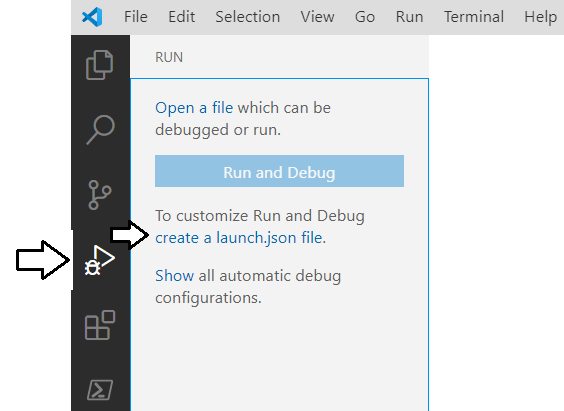
* 1. Dddd



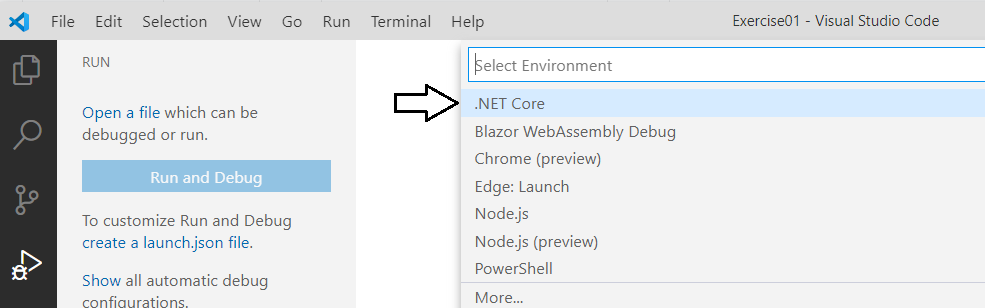
* 1. Xxxxx



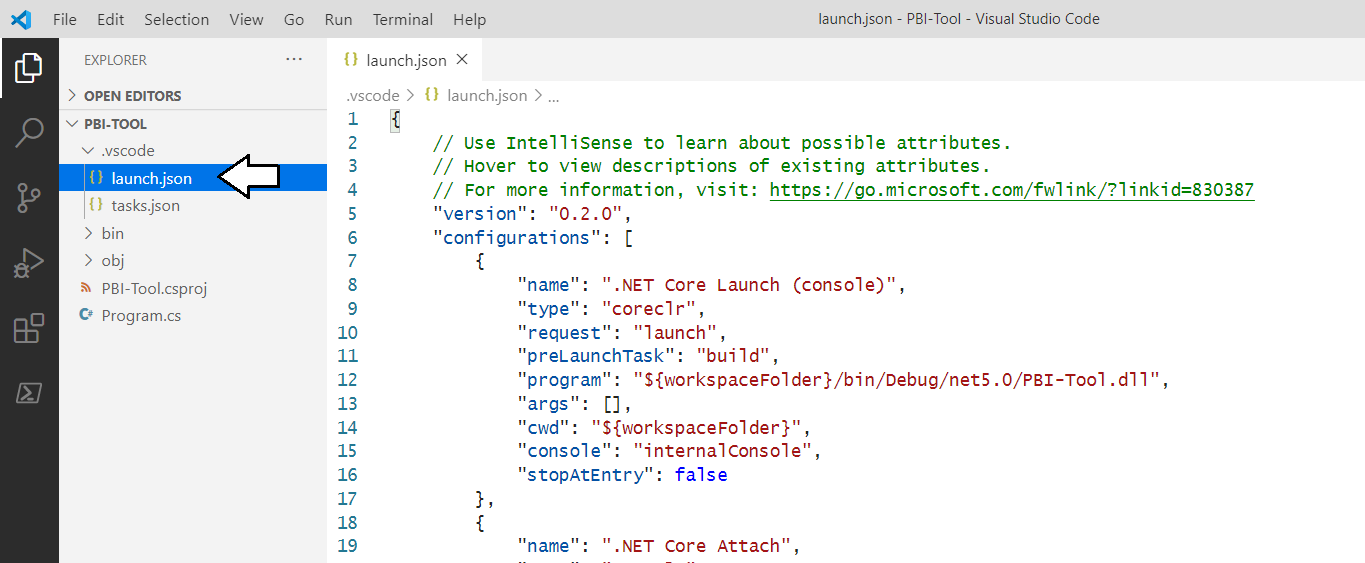
* 1. X



* 1. X



* 1. sssss



Ssss

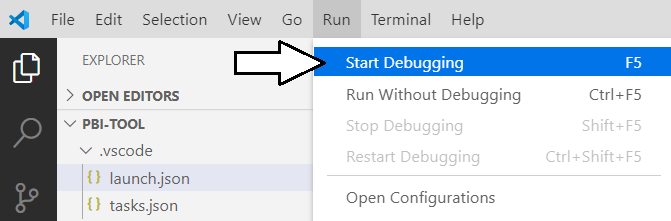
"logging": { "moduleLoad": false }

X

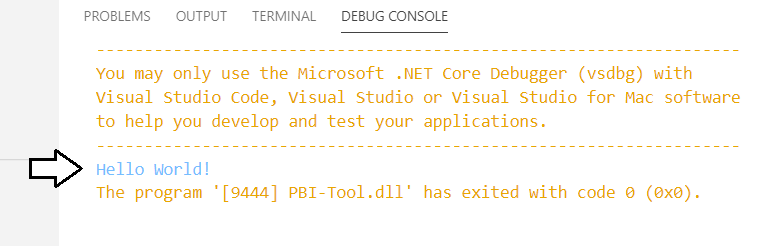


The reason for doing this is

* 1. sssss



* 1. dddd



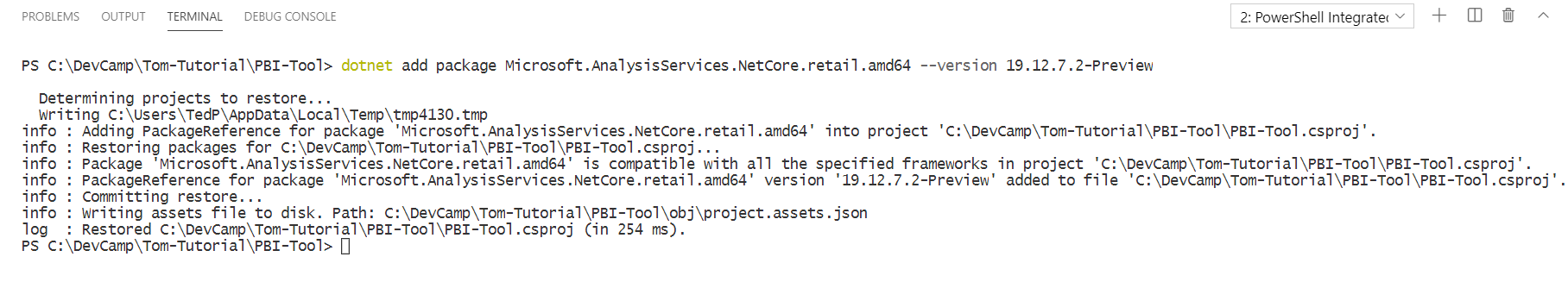
OK, you got hello world out of the way

1. Add the NuGet package with the Tabular Object Model.
   1. Return to the Terminal console.
   2. Type and execute the following command to add the NuGet package

dotnet add package Microsoft.AnalysisServices.NetCore.retail.amd64 --version 19.12.7.2-Preview

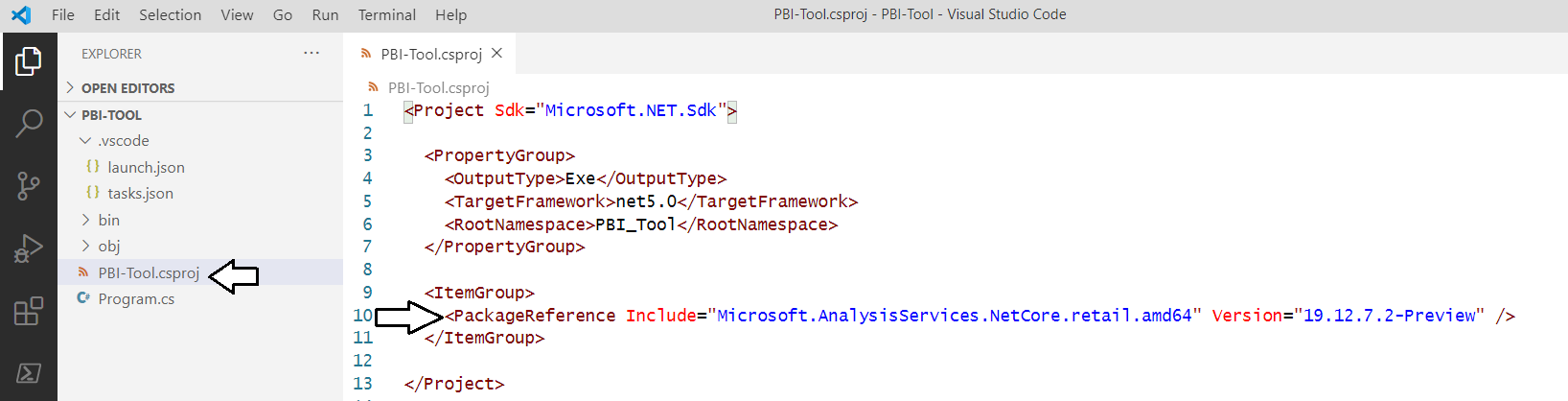
Check this URL to discuss the latest release: <https://www.nuget.org/packages/Microsoft.AnalysisServices.NetCore.retail.amd64>

* 1. Ssss

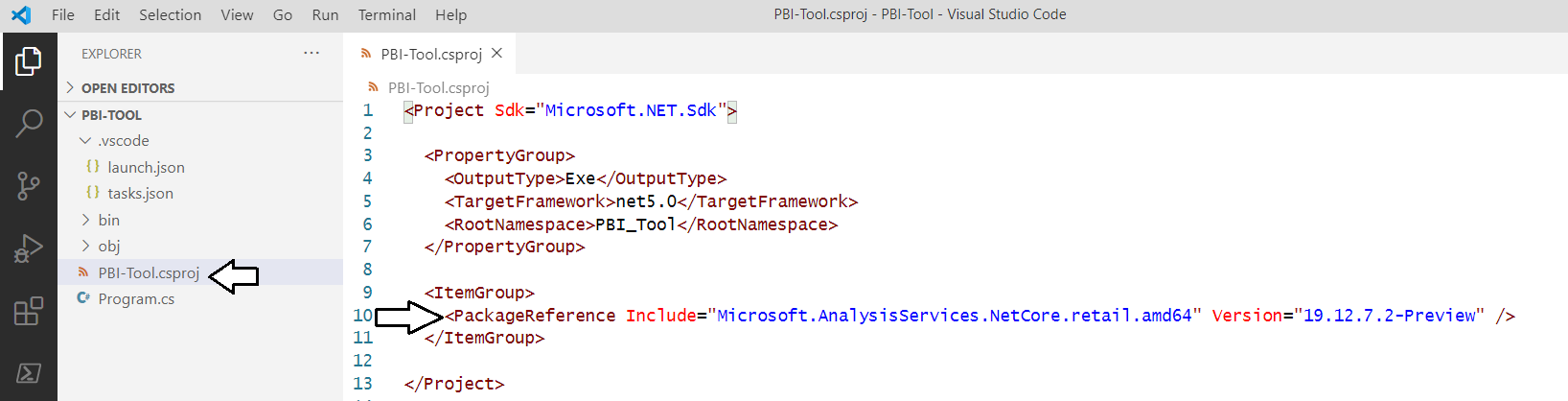


Now you can program against the AMO object which include the Tabular Object Model

* 1. X



* 1. ddddd



* 1. Close **PBI-Tool.csproj** without saving any changes.

1. Modify the C# code in the program.
   1. Sss

using System;

using Microsoft.AnalysisServices.Tabular;

class Program {

const string connectString = "localhost:50000"; // update for port number on your machine

static void Main(string[] args) {

Server server = new Server();

server.Connect(connectString);

Model model = server.Databases[0].Model;

foreach(Table table in model.Tables) {

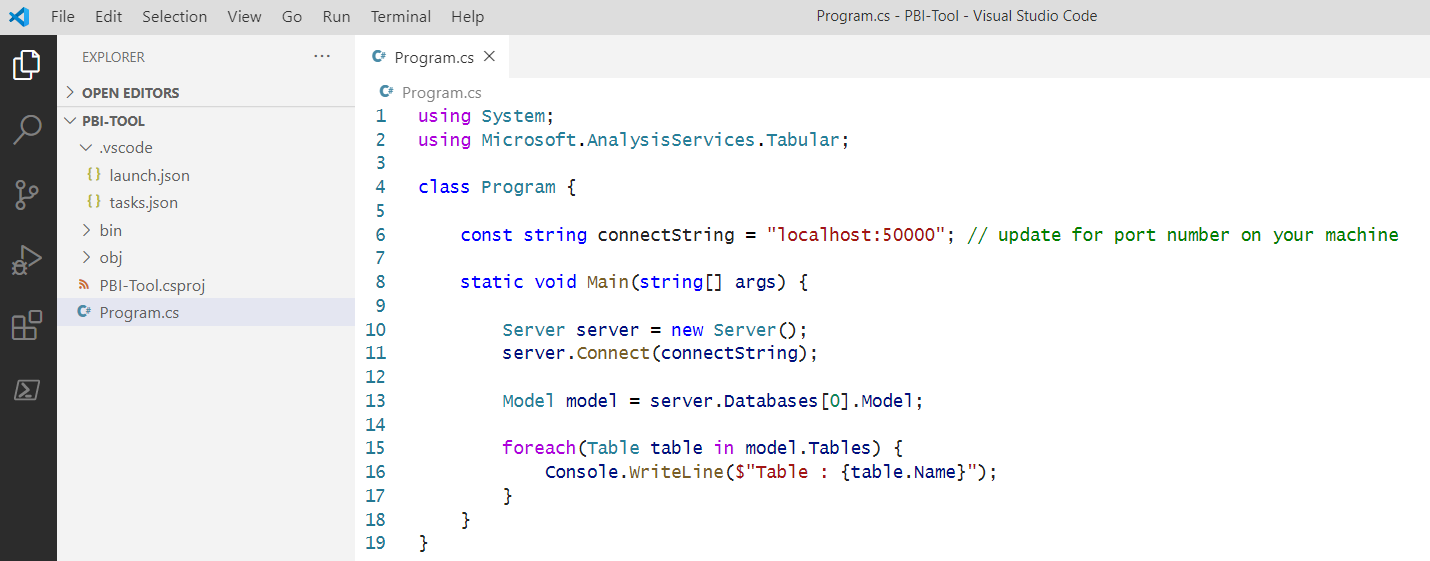
Console.WriteLine($"Table : {table.Name}");

}

}

}

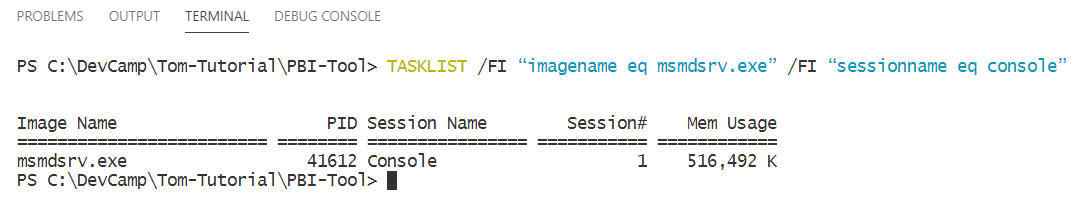
* 1. Aaa



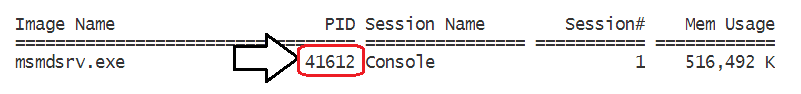
1. Xxxssss
   1. sss

TASKLIST /FI “imagename eq msmdsrv.exe” /FI “sessionname eq console”

* 1. Looks like this



* 1. xxx

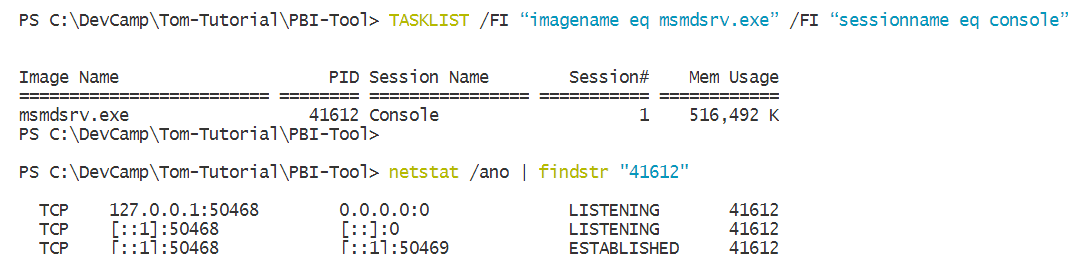


You're half way there. Now you have the process ID (PID) but you still have to find the port number.

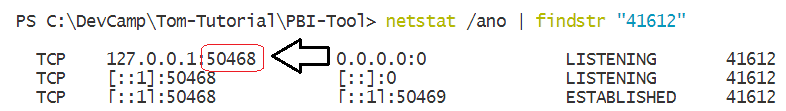
* 1. X

netstat /ano | findstr "41612"

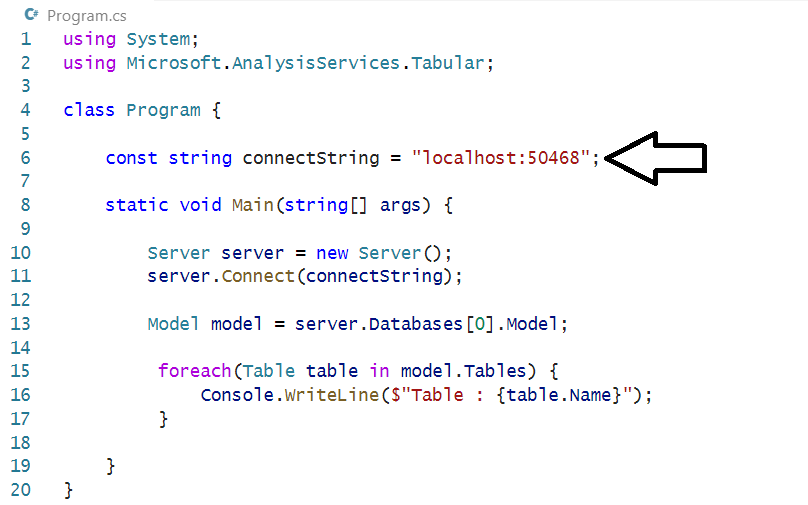
* 1. X



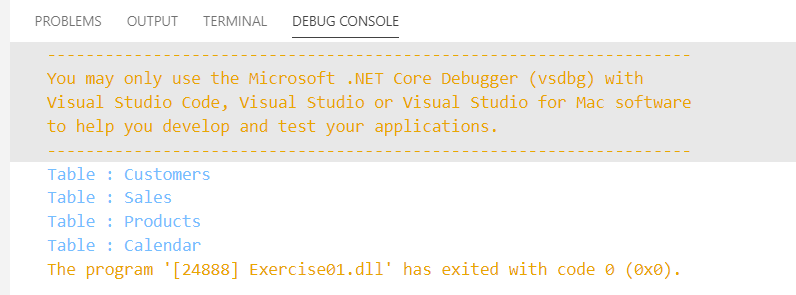
* 1. Z



* 1. x

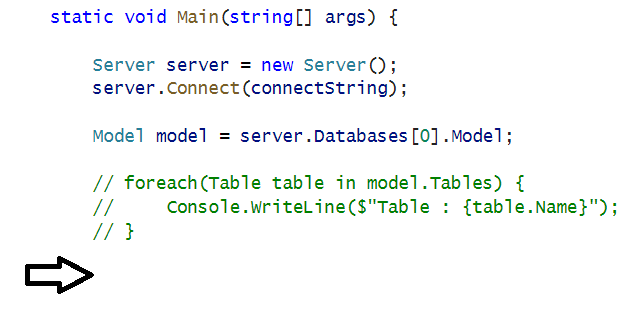


1. zzzzz
   1. ss



Congratulations. You can now say you have programmed the Tabular Object Model. You're friends will be so jealous.

1. x
   1. caddd



* 1. x

Table table = model.Tables["Sales"];

if (table.Measures.ContainsName("VS Code Measure")) {

Measure measure = table.Measures["VS Code Measure"];

measure.Expression = "\"Hello Again World\"";

}

else {

Measure measure = new Measure() {

Name = "VS Code Measure",

Expression = "\"Hello World\""

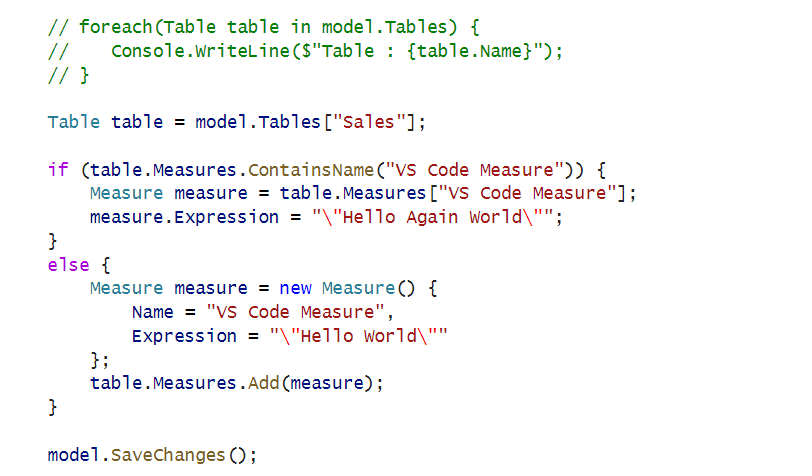
};

table.Measures.Add(measure);

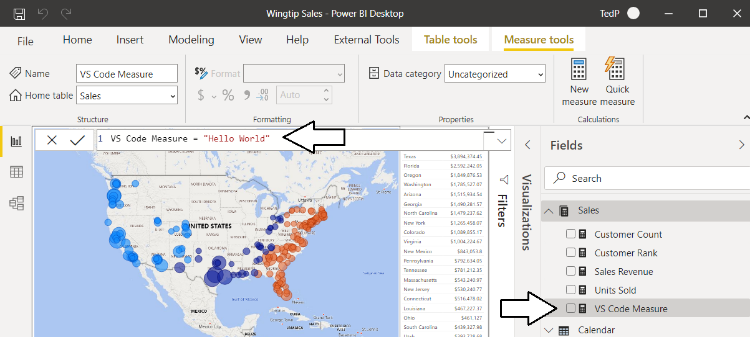
}

model.SaveChanges();

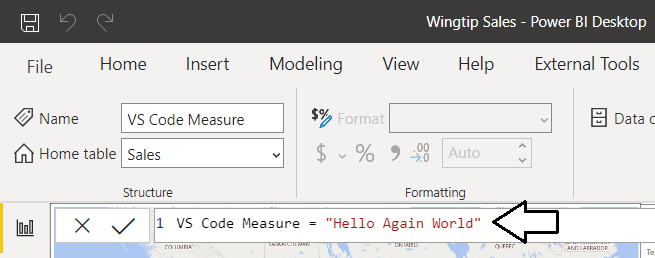
* 1. x



* 1. x



* 1. ddd



Conclusion

### Exercise 2: xxx

In this exercise, you will connect to an instance of Power BI Desktop

Iterate through every Table in the model

Iterate through every Column in the “current” table from the outer loop

If the Column is numeric and not hidden, create a simple [Sum of <column>] measure

1. Xxxxx
   1. Xxxx

### Exercise 3 xxx

In this exercise, you will download xx.

1. Xxxxx
   1. xxxx

### Exercise 4: xxx

In this exercise, you will download xx.

1. Xxxxx
   1. xxxx