



C# and .NET Training - Workstation Setup Guide

Purpose:

Performing development requires a few components to be installed on your workstation. For .NET (C#) development, you will need to have Visual Studio installed.

For those who have Visual Studio Code installed, we will also review setting up that environment for .NET development.

If you want to set up your development environment similar to mine, the instructions below will walk you through the setup process. If you're following these steps, it is best to do them in the order presented.

Table of Contents:

- [Prerequisites](#)
- [Install Visual Studio](#)
- [Prep Your File System](#)
- [Set Up Visual Studio](#)
- [Customize Visual Studio Code](#)
- [Create a Sample Application](#)
- [Customize Visual Studio](#)
- [Clone the .NET Training Repository](#)
- [Install SQL Server](#)
- [Install SQL Server Management Studio](#)
- [Restore the Sample Database](#)
- [\(optional\) Install Codeium](#)

Prerequisites

Prior to any of the steps listed below, please complete all tasks in the guide titled:

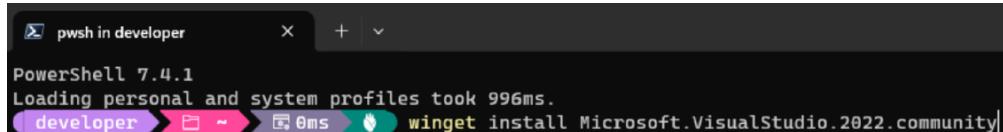
[01 Setting Up Your Workstation for Development Training.pdf](#)

Install Visual Studio

Microsoft Visual Studio is the industry standard IDE for .NET development. While there are alternative platforms, we will be using Visual studio throughout the training for both C# and later the Unity API.

One very important note: If you are using the Community Edition of Visual Studio, you cannot develop, build, or deliver software to clients. Commercial use requires a licensed edition. If you believe you require a license for visual studio, discuss acquiring one with your leadership.

1. In your terminal, run one of the following commands (based on the edition you are using):



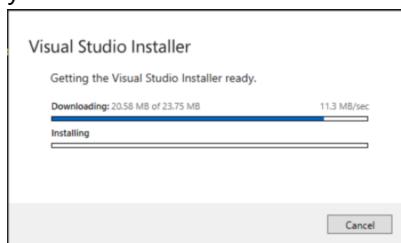
```
pwsh in developer
PowerShell 7.4.1
Loading personal and system profiles took 996ms.
developer ➜ 0ms ➜ winget install Microsoft.VisualStudio.2022.community
```

- Community Edition:
`winget install Microsoft.VisualStudio.2022.Community`
- Professional Edition:
`winget install Microsoft.VisualStudio.2022.Professional`
- Enterprise Edition:
`winget install Microsoft.VisualStudio.2022.Enterprise`

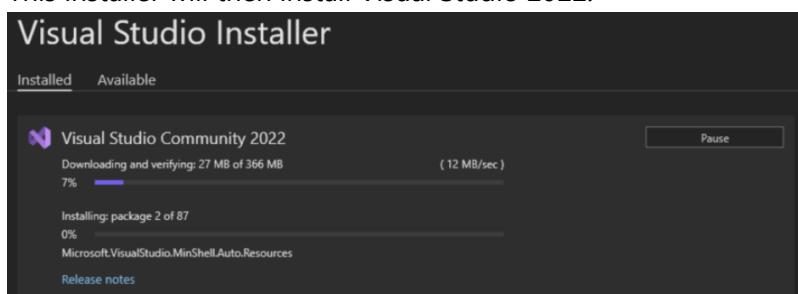
2. Click "Yes" on the UAC prompt.



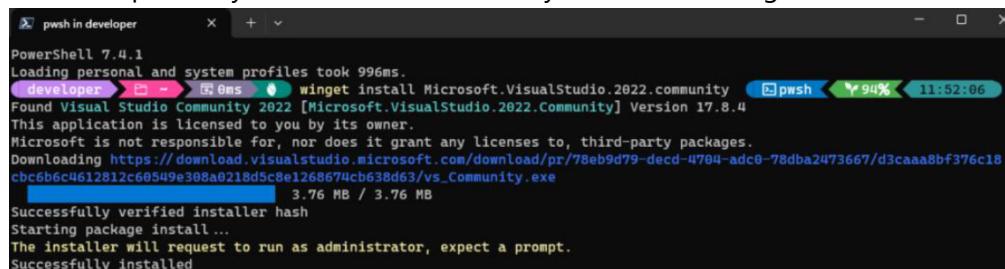
3. First, the Visual Studio Installer will be installed. This component can be used later to modify or update your installation.



4. This installer will then install Visual Studio 2022.



5. After completion, you will see a "successfully installed" message in the terminal.



```

pwsh in developer
PowerShell 7.4.1
Loading personal and system profiles took 996ms.
developer ➜  ~  ➜  0ms  winget install Microsoft.VisualStudio.2022.community  pwsh  94%  11:52:06
Found Visual Studio Community 2022 [Microsoft.VisualStudio.2022.Community] Version 17.8.4
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://download.visualstudio.microsoft.com/download/pr/78eb9d79-decd-4704-adc0-78dba2473667/d3caa8bf376c18
cbc6b6c4612812c66549e308a0218d5c8e1268674cb638d63/vs_Community.exe
3.76 MB / 3.76 MB
Successfully verified installer hash
Starting package install ...
The installer will request to run as administrator, expect a prompt.
Successfully installed

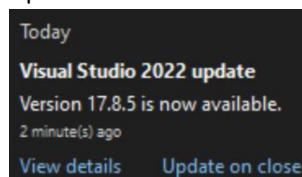
```

We now need to add some development components to the installation.

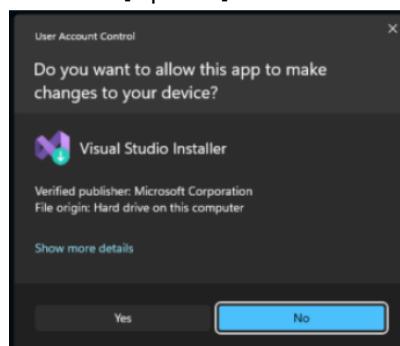
6. Click on the start button. Then search for and run the "Visual Studio Installer"



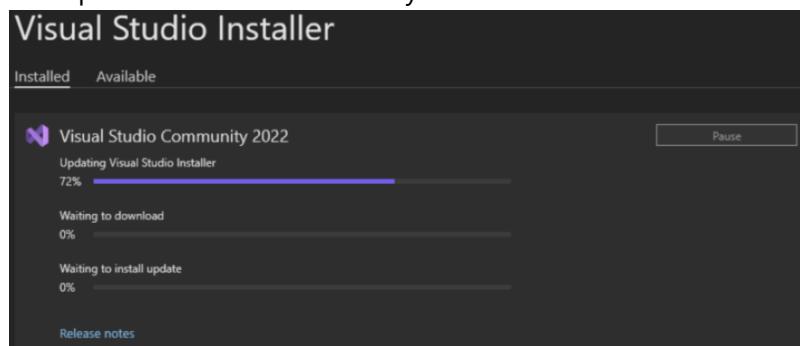
7. You may see an alert like the one below, indicating that an updated version is available. If you do, we'll update the installation first. If not, you can skip to [step 10](#).



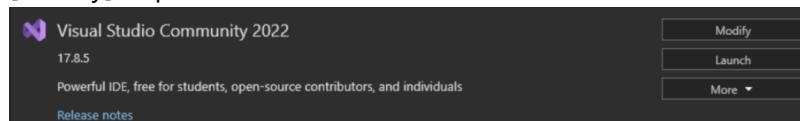
8. Click the [Update] button and respond "Yes" to the UAC alert



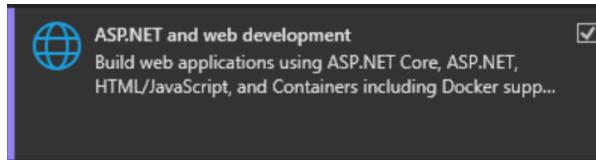
9. The update will run automatically



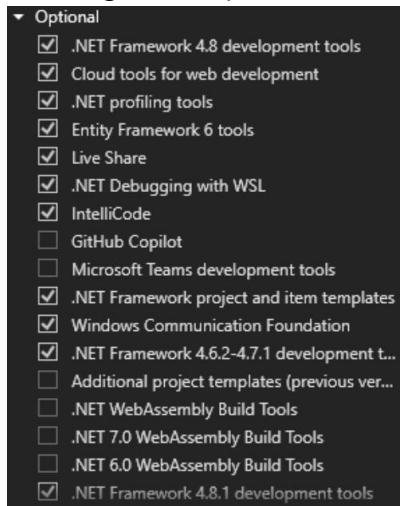
10. After the update (or immediately, if no update was available), you will see the below options. Click on [Modify] to proceed.



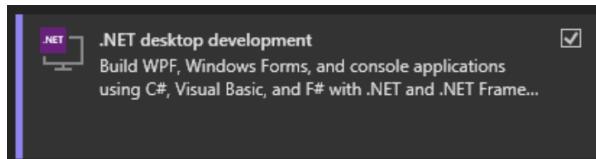
11. In the left-hand pane, check the box on the "ASP.NET and web development" function group.



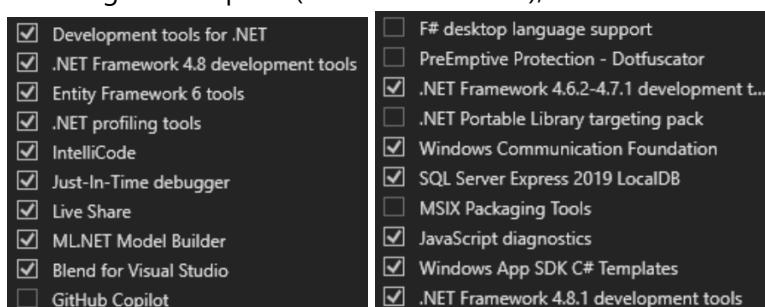
12. In the right-hand pane (Installation Details), make sure all of the below options are checked:



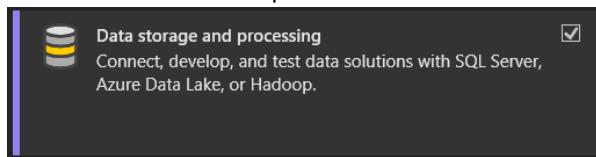
13. Back in the left-hand pane, check the box for ".NET desktop development"



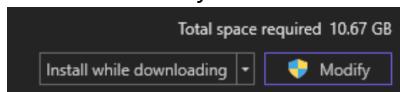
14. In the right-hand pane (Installation Details), make sure all of the below options are checked:



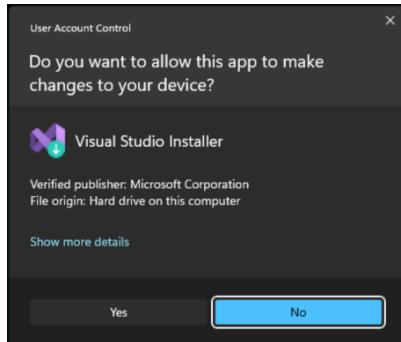
15. Back in the left-hand pane, check the box for ".Data storage and processing"



16. Click on "Modify"

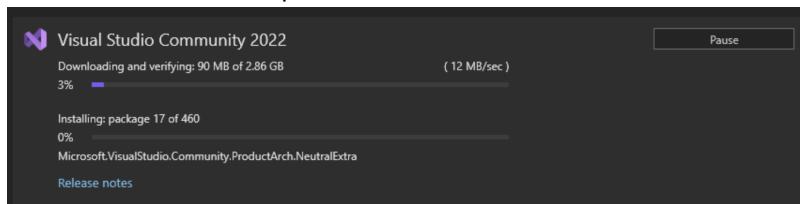


17. As usual, respond "Yes" to the UAC warning:



Note: You can install any additional components you want, but the items listed above are sufficient for everything covered in the training course.

18. The Visual Studio components will now install.

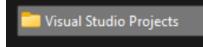


19. After completion, close out of the Visual Studio installer and proceed to the next section.

Prep Your File System

As part of the next section, we will be creating a sample project. We'll need somewhere to place it.

1. On your PC, create a folder called "Visual Studio Projects" (I created mine on the C: root)



Set Up Visual Studio

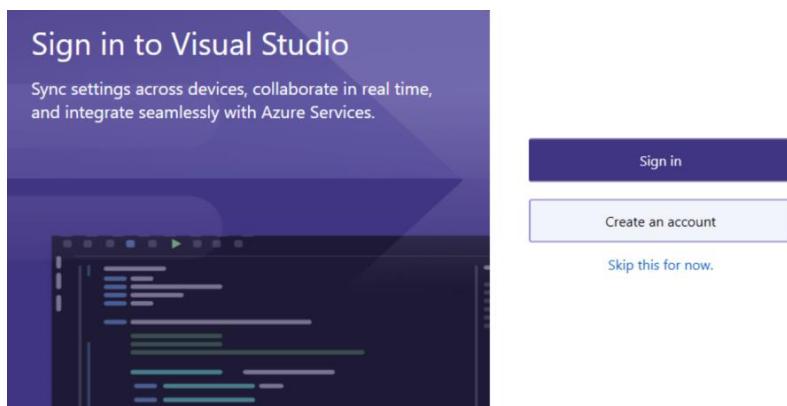
Visual Studio is highly configurable, and we will not be addressing the majority of the options here. However, there are several that will make the experience easier in training, so we will handle them now.

1. Launch Visual Studio

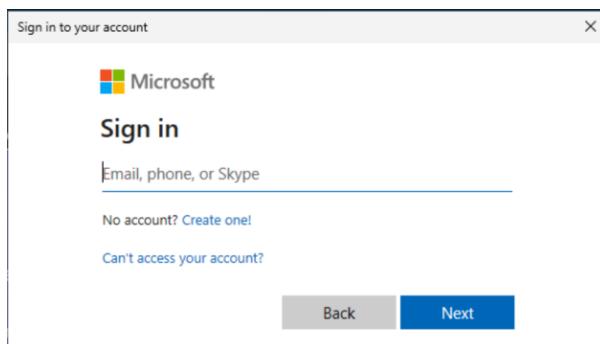


2. You will be prompted to log in with your Microsoft account. Click on "Sign In"

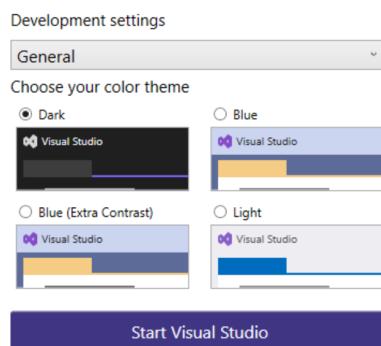
Note: Do not skip this step, as we need to set up with a Microsoft account for a later step



3. On the "Sign in" dialogs, enter the email and password for your Microsoft Account You should already have an MS account associated with the email address you used to obtain your Visual Studio license.



4. Once you have successfully logged in, you will be prompted to select a color theme. I use the default "Dark" theme, but you can configure yours however you like.



Once you select your theme, click on [Start Visual Studio] and continue to the next section

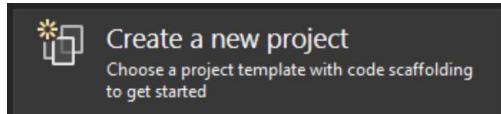
Create a Sample Application

Now we need to make sure everything installed properly by creating a sample application in Visual Studio.

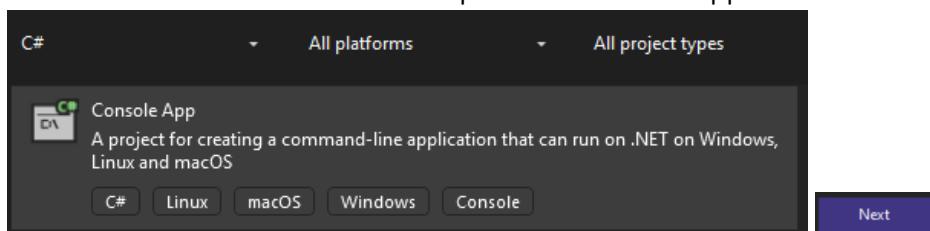
1. If it's not already running, launch Visual Studio



2. On the startup dialog, select the option to "Create a new project"



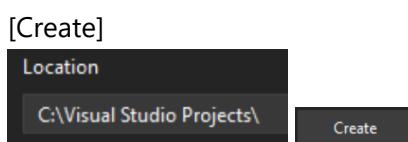
3. Search for "console" and select the option for "Console App" Then click "Next"



4. Name the project "HelloWorld"



5. Modify the location to point to the "Visual Studio Projects" folder you created previously, then click [Create]



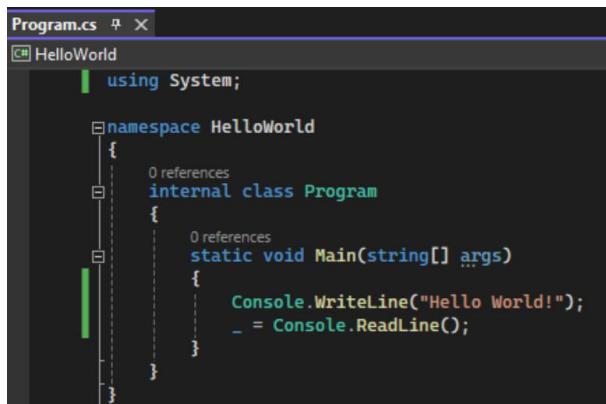
6. Select ".NET 8.0" for the target framework, then click [Create]



7. Add the following code in the Program.cs file that gets created for you (don't worry about understanding it for now):

```
using System;

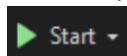
namespace HelloWorld
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            _ = Console.ReadLine();
        }
    }
}
```



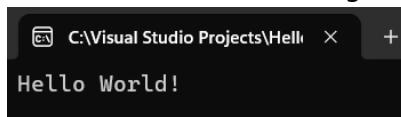
```
Program.cs  X
C# HelloWorld
using System;

namespace HelloWorld
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            _ = Console.ReadLine();
        }
    }
}
```

8. At the top of the screen, click the "Start" button to run the program.



9. You should see the following in a new terminal window:

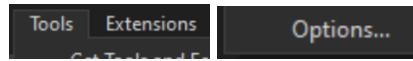


```
C:\Visual Studio Projects\HelloWorld  +
Hello World!
```

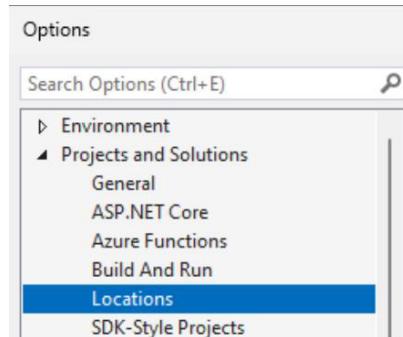
Customize Visual Studio

There are a few optional convenience settings I recommend in Visual Studio.

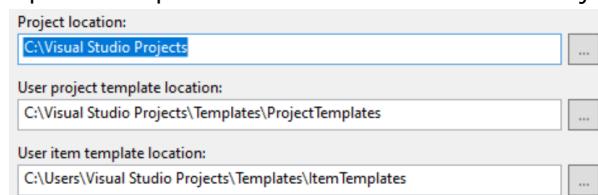
1. In Visual Studio, click on TOOLS > OPTIONS



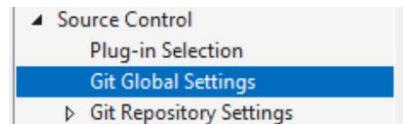
2. Under "Projects and Solutions," select "Locations"



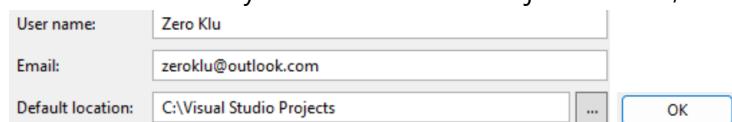
3. Update the paths to use the "Visual Studio Projects" folder you created earlier in the process.



4. Under "Source Control," select "Git Global Settings"



5. Enter the username and email address you use for Git (these may already be filled in), and change the default location to your "Visual Studio Projects" folder, then click [OK].



Customize Visual Studio Code

Although Visual Studio is the flagship IDE for .NET development, you can do many kinds of C# development with Visual Studio Code (the main exception being UI development, since VS Code does not have a visual UI designer like Visual Studio does). In order to use Visual Studio Code, you will need to install the C# extension:

- [C# Dev Kit](#)

As noted previously, even when using VS Code for development, you need to be signed into an account associated with a license for Visual Studio.

Clone the .NET Training Repository

You'll need to clone a copy of the training repository to work with.

The repository is located here:

- GitHub: <https://github.com/ZeroKlu/csharp-training-gh>

1. In a browser, navigate to the GitHub repository and make sure you're able to access it.

- If you cannot, request access from zeroklu@protonmail.com.

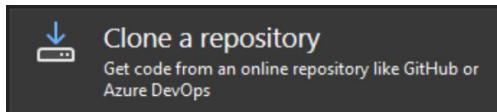
2. If you will be using Visual Studio, continue to [step 3](#).

If you will be using Visual Studio Code, skip to [step 7](#).

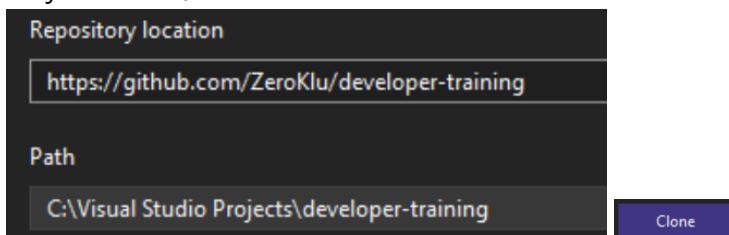
3. Launch Visual Studio



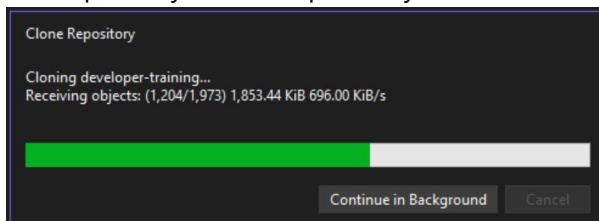
4. On the startup dialog, select the option to "Clone a repository".



5. in the URL to the repository you want to clone, verify that the "Path" target is your Visual Studio Projects folder, and click "Clone."



6. The repository will be copied to your local machine.



If you will be using Visual Studio, skip to [Install SQL Server](#).

7. Launch Visual Studio Code



8. Click on the Source Control icon



9. Click on the button to "Clone Repository"

Clone Repository

10. Enter the URL to the repository you want to clone, then click "Clone from URL"

<https://github.com/ZeroKlu/csharp-training-gh>

Clone from URL <https://github.com/ZeroKlu/csharp-training-gh>

 Clone from GitHub

Install SQL Server

1. Download the installation you prefer:

[SQL Server Developer Edition](#) or [SQL Server Express Edition](#)

* I recommend SQL Server Developer Edition

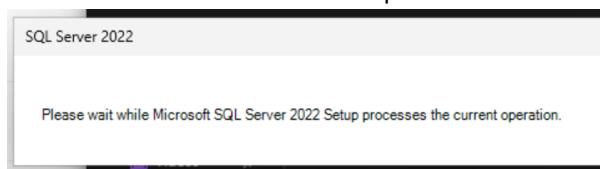
2. Right-click the downloaded ISO file and choose "Mount"



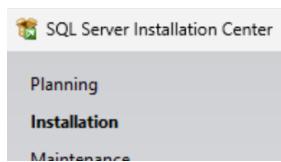
3. From the mounted drive, run setup.exe



4. Wait for the initial load to complete.



5. When the installer dialog loads, select "Installation."



6. Select the option to create a "New SQL Server standalone installation..."



7. Select your free edition and click [Next]



8. Accept the license terms, then click [Next]



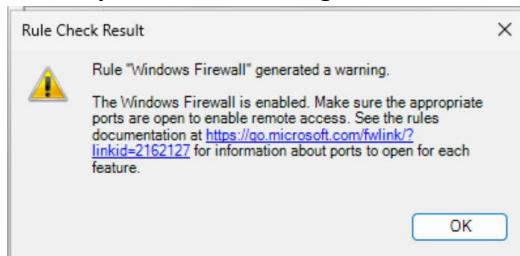
9. Your system will be scanned for compatibility with the install.

Operation completed. Passed: 4. Failed 0. Warning 1. Skipped 0.



Result	Rule	Status
✓	Machine Learning Server shared feature support	Passed
✓	Consistency validation for SQL Server registry keys	Passed
✓	Computer domain controller	Passed
⚠	Windows Firewall	Warning
✓	Microsoft .NET Framework 4.7.2, or newer, is required	Passed

10. You may receive a warning about the Windows Firewall. This is not a problem for our purposes



11. Uncheck the option for the Azure extension and click [Next]. We won't be working in Azure.

Azure Extension for SQL Server

12. Under "Instance Features," we only need "Database Engine Services." Click [Next]

Instance Features
 Database Engine Services

13. Leave the default selection of "Default instance" and click [Next]

Default instance

14. Leave the default service settings and click [Next]

Service	Account Name	Password	Startup Type
SQL Server Agent	NT Service\SQLSERVERA...		Manual ▾
SQL Server Database Engine	NT Service\MSSQLSERVER		Automatic ▾
SQL Server Browser	NT AUTHORITY\LOCAL ...		Disabled ▾

15. Change the "Authentication Mode" to "Mixed Mode..." and enter a password for the SA user.

Authentication Mode

Windows authentication mode
 Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password: ······

Confirm password: ······

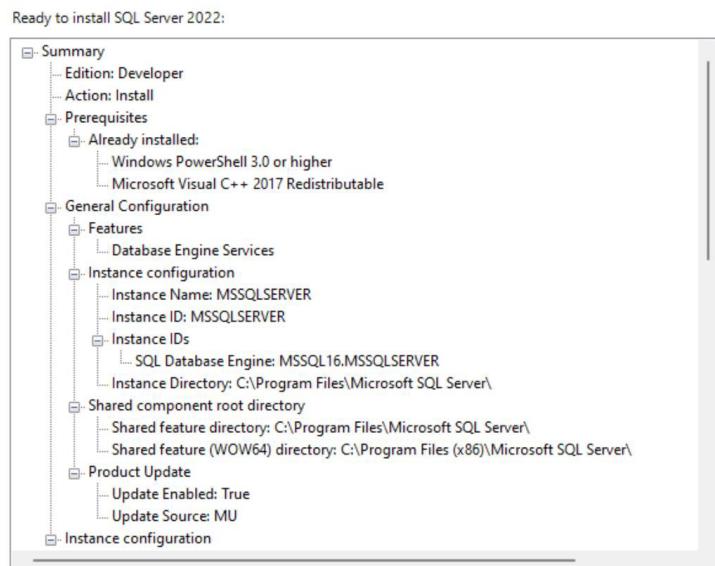
16. Click the button labeled "Add Current User"

17. After your Windows user appears in the list, click [Next]

Specify SQL Server administrators

Win11Dev\developer (developer)

18. Review the installation summary, and click "Install"



19. SQL Server will install. This may take several minutes, so be patient.



Running package: sql_engine_core_shared : Record Install Start in Windows Event Log

20. When the install completes, you should see the following:

Information about the Setup operation or possible next steps:

Feature	Status
Database Engine Services	Succeeded
SQL Browser	Succeeded
SQL Writer	Succeeded
Setup Support Files	Succeeded

Details:

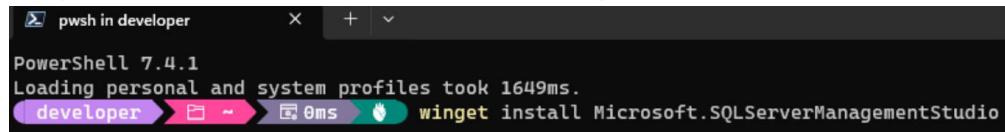
Install successful.

Install SQL Server Management Studio (SSMS)

SQL Server Management Studio no longer installs along with the database engine. It is a separate, version-independent component. So, we'll need to install it now.

1. In your terminal, run the following command:

```
winget install Microsoft.SQLServerManagementStudio
```

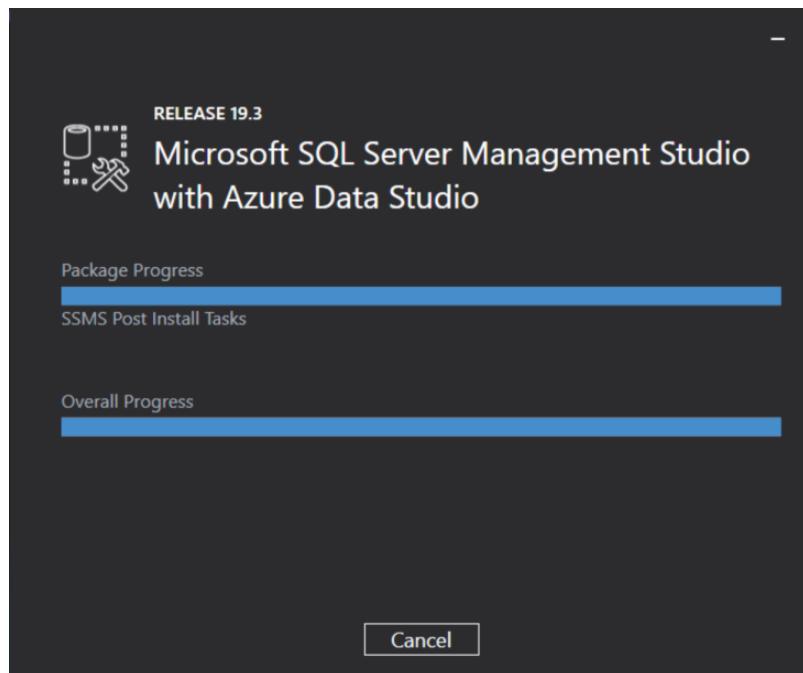


A screenshot of a PowerShell window titled 'pwsh in developer'. The command 'winget install Microsoft.SQLServerManagementStudio' is being typed into the terminal. The output shows 'PowerShell 7.4.1' and 'Loading personal and system profiles took 1649ms.' followed by a progress bar indicating the installation process.

2. Respond "Yes" to the UAC prompt



3. The installer will run



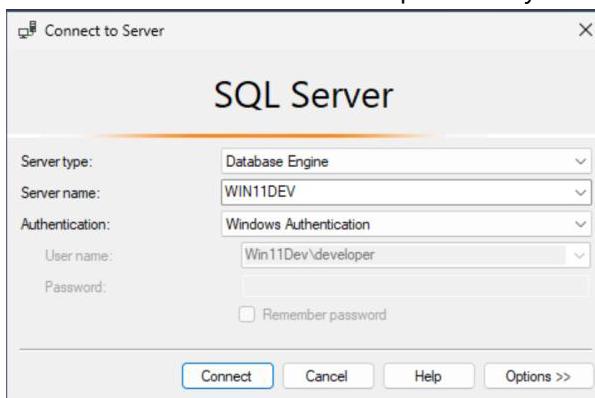
[Restore the Sample Database]

I have provided an example database that includes the data we will use in training. We need to make the database available in the SQL Server instance we just set up.

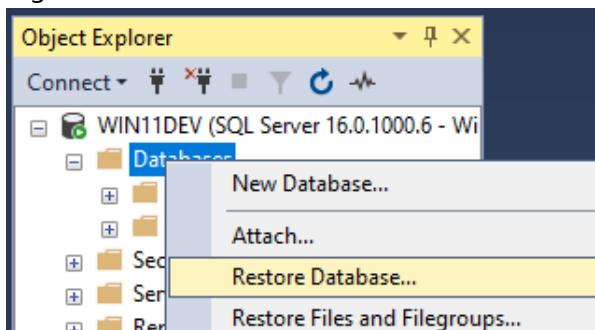
1. Launch SQL Server Management Studio



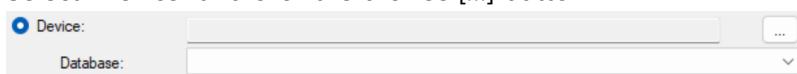
2. Log in either using Windows authentication as the user you added during the installation or SQL Server authentication with the SA user password you set up during installation.



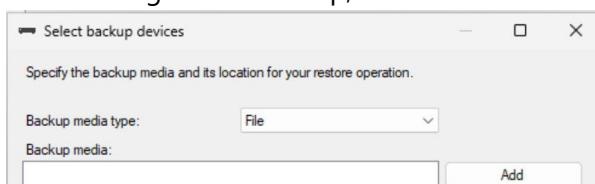
3. Right-click on the "Databases" node and select "Restore Database"



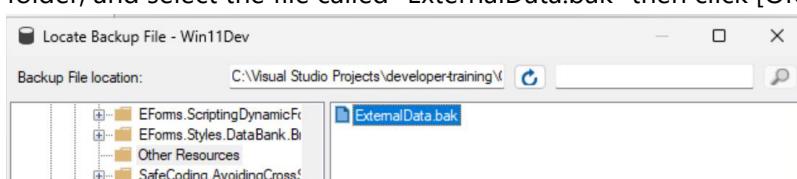
4. Select "Device" and click the browse [...] button



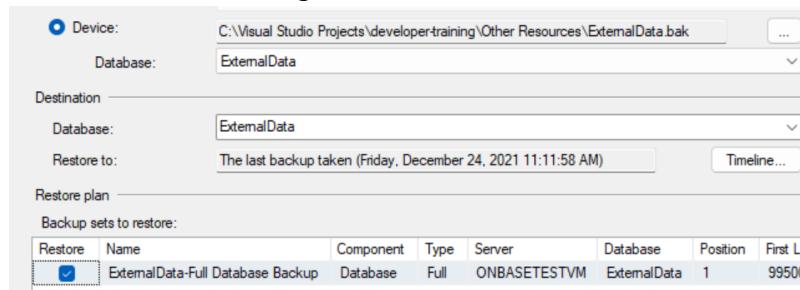
5. In the dialog that comes up, click the "Add" button



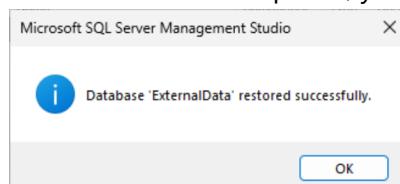
6. Navigate into the repository you cloned earlier and into [00 - Resources/Setup Documents/database](#) folder, and select the file called "ExternalData.bak" then click [OK] and [OK] again.



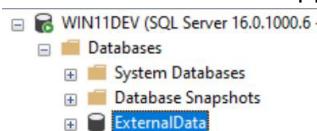
7. Leave the default settings and click [OK]



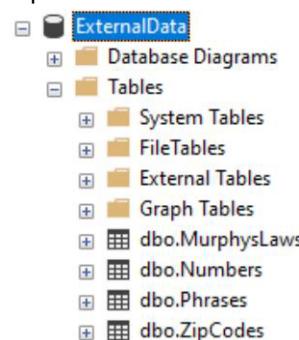
8. After the restore completes, you should see the following message:



9. The database will now appear under the "Databases" node. Select it.



10. Expand the "Tables" node. There should be four tables:



11. Click the "New Query" button at the top of the UI



12. Enter the following SQL Query and click the "Execute" button or press [F5]

```
SELECT TOP 10 * FROM dbo.ZipCodes
```



13. You should see the following results of the query:

	Id	State	County	City	ZipCode
1	1	AA	NONE	APO	34002
2	2	AA	NONE	APO	34003
3	3	AA	NONE	APO	34009
4	4	AA	NONE	APO	34020
5	5	AA	NONE	APO	34021
6	6	AA	NONE	APO	34022
7	7	AA	NONE	APO	34023
8	8	AA	NONE	APO	34024
9	9	AA	NONE	APO	34025
10	10	AA	NONE	APO	34030

(optional) Install Codeium

The team of developers I run use [Codeium](#) as an AI companion while coding. I don't recommend relying on an AI assistant to write your code for you, but it's definitely useful to have an AI companion that can provide suggestions and answer questions. In particular, I find it especially useful to be able to ask an AI to explain code you did not write yourself rather than copy-pasting.

You can create a free account here: <https://codeium.com/account/register>

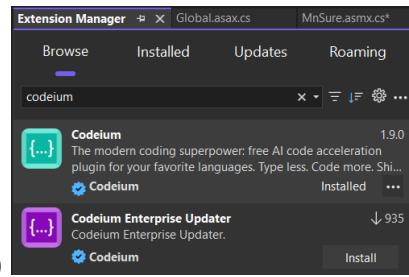
That said, the folks at Codeium are really great, so if you're getting value from the product, I'd ask that you consider subscribing to their premium tier at \$15/month.

To install Codeium:

- In Visual Studio:

Manual download located [here](#)

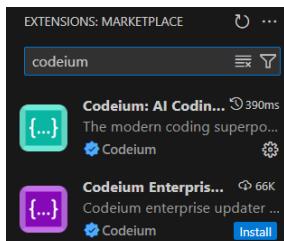
1. Click on "Extensions" > "Manage Extensions"
2. Search for "Codeium"



3. Select the "Codeium" extension (not "Enterprise")
4. Click the "Install" button

- In Visual Studio Code: Manual download located [here](#)

1. Click on the Extensions icon
2. Search for "Codeium"
3. Select the "Codeium" extension (not "Enterprise")



4. Click the "Install" button

Congratulations! Your system is set up for C# training.

Happy Coding!
