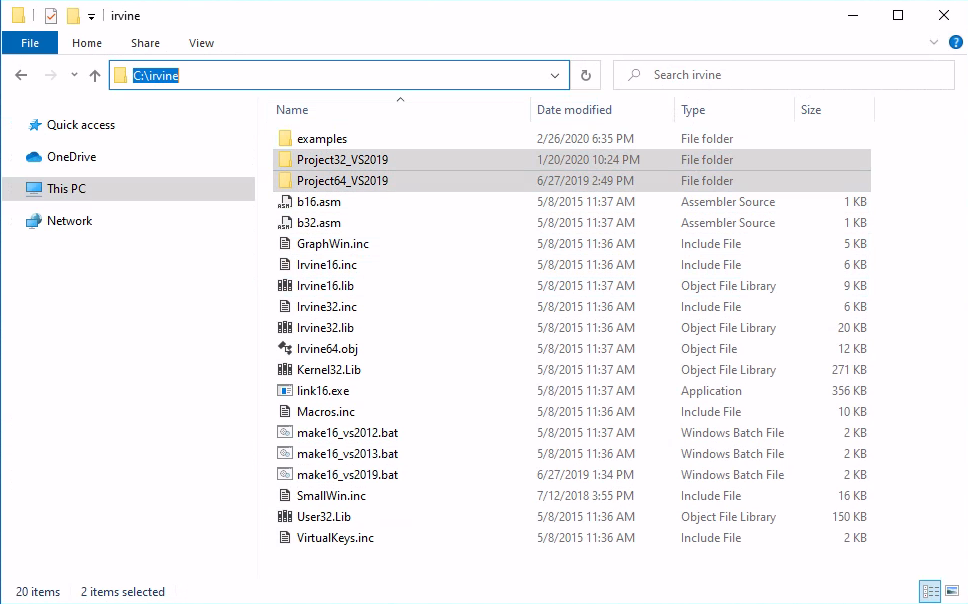
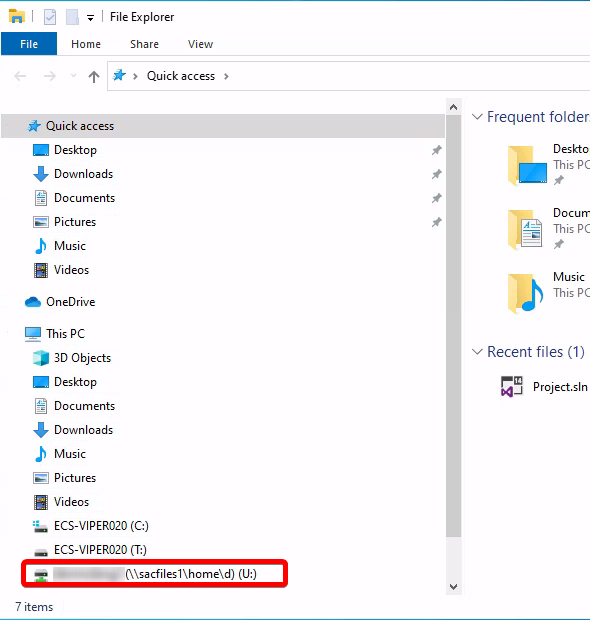
Instructions to set up a MASM project on Visual Studio 2019

# Retrieving sample Project Files

On the File Explorer, navigate to [C:\irvine](file:///C:\irvine). You will see that there are two folders inside here, “Project32\_VS2019” and “Project64\_VS2019”.

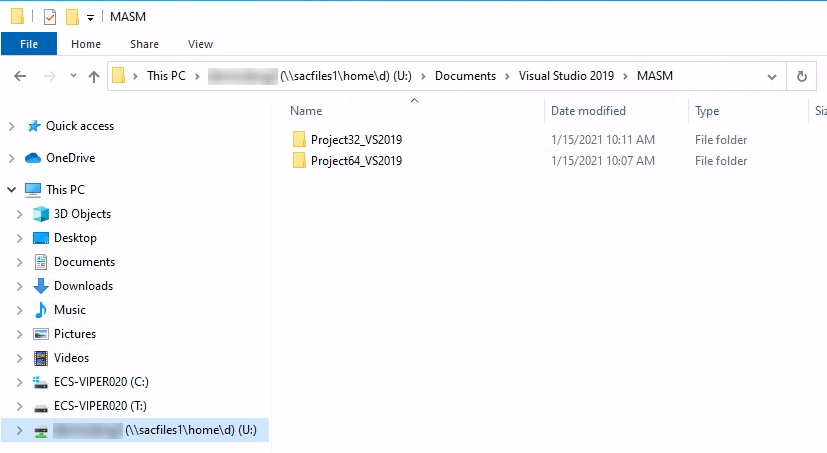


1. Copy both “**Project32\_VS2019**” and “**Project64\_VS2019**” folders and store them anywhere under your **U: Drive** connected storage space under “**This PC > (your username)**”



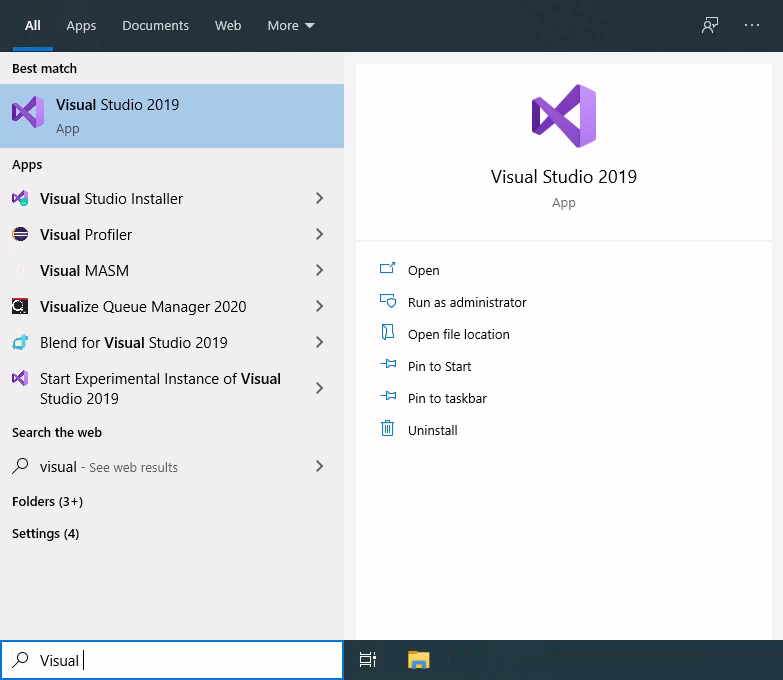
For more information on how U: Drive works, see [Section 6.1](#6.1.U: Drive Storage Space on ECS Lab Machines|outline).

1. In your U: Drive, navigate to Documents > Visual Studio 2019.   
    (**U:\Documents\Visual Studio 2019)  
   Create a new folder called MASM**, and copy both project templates in this directory.  
   (It doesn’t matter where you place them, as long as its under [U:\](file:///U:\) or its subfolders.)



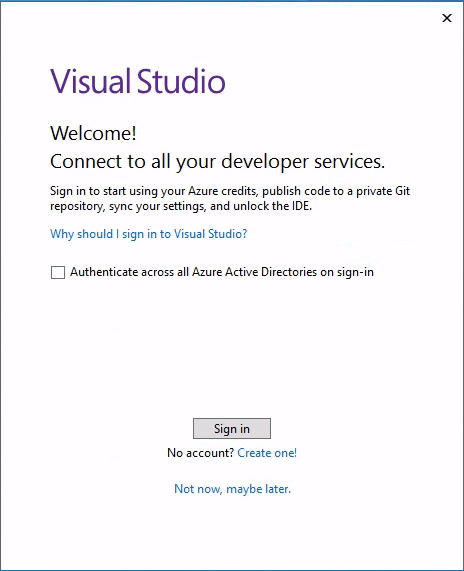
# Importing sample project files

1. Open Visual Studio 2019 on the remote machine by navigating the Start Menu. You can simply go to Start, and type in Visual Studio 2019.

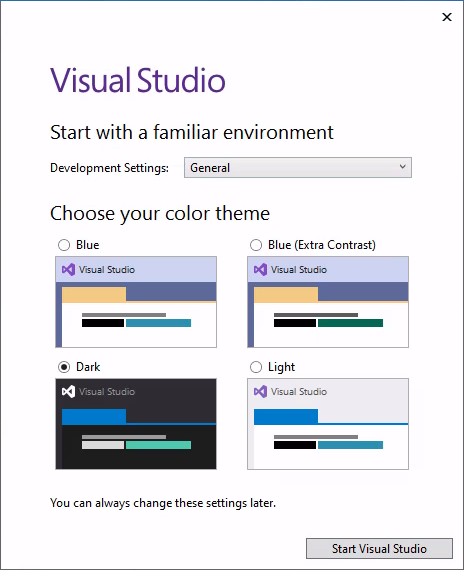


1. If it’s your first time starting up Visual Studio in a specific lab machine, you may see the following pop-up:

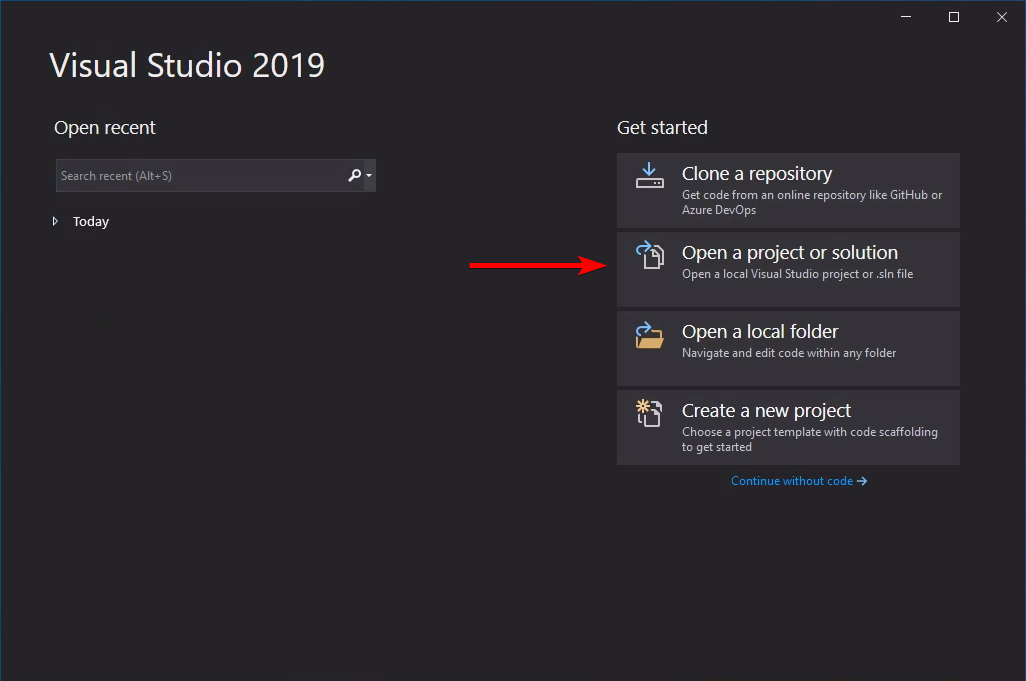
Click “Not now, maybe later”.



1. A screen will pop up to choose your color theme. Choose any to your preference and click “Start Visual Studio”.

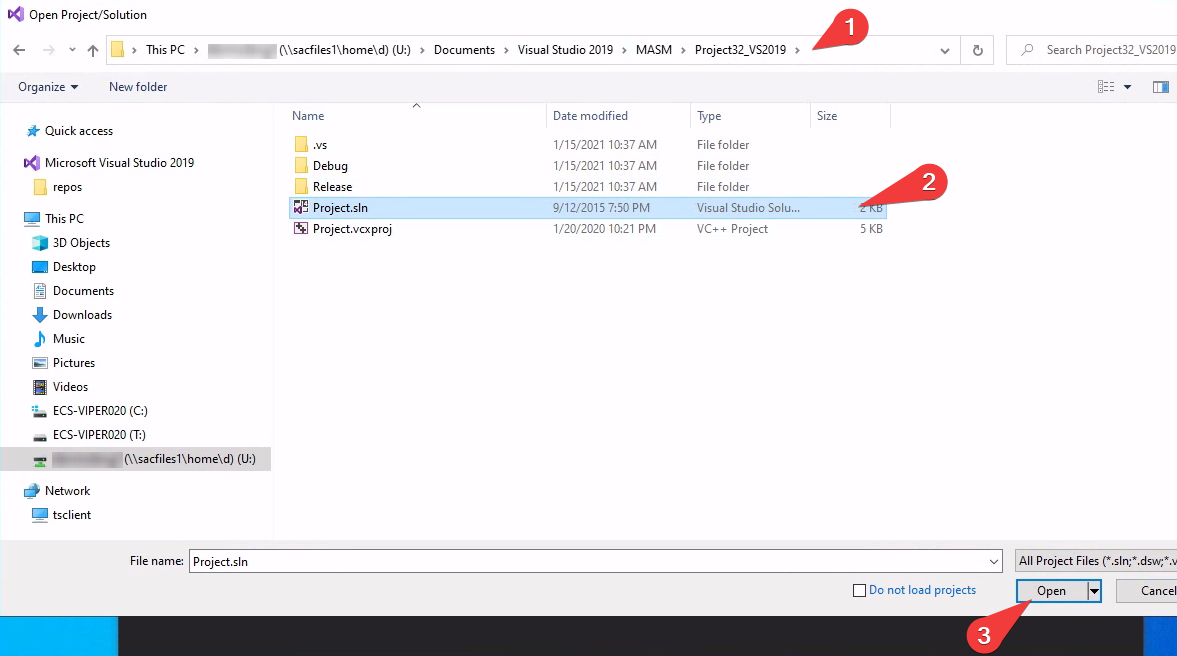


1. You will then see a screen where you can import a project supported by Visual Studio.   
   Click on “**Open a project or solution**”

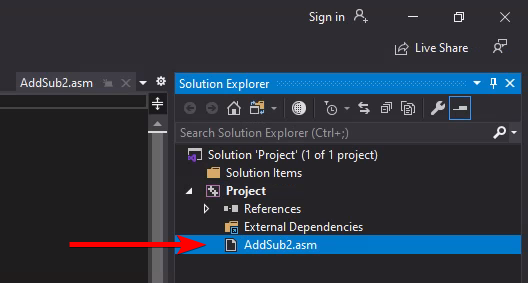


1. Browse for your project folder (from the your U: drive), and import your project solution. Visual Studio Project solutions are usually have their file extensions as a **.sln**

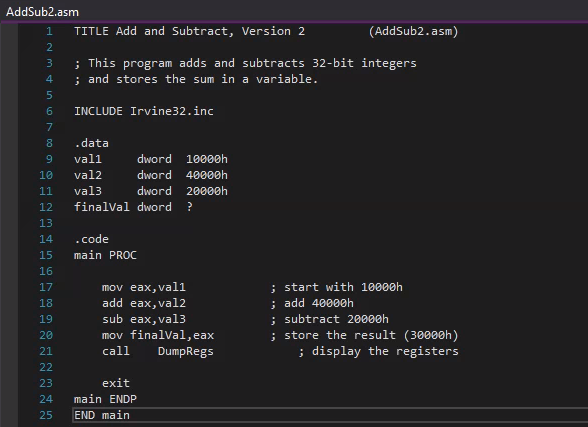
To import your sample projects, select Project.slnandclick Open.



1. Under the Solution Explorer pane, located on the top-right corner of Visual Studio, locate and click on a **.asm** file containing the assembly source code. For this example, if you have imported the 32-bit project, click on AddSub2.asm.



1. After clicking on the **.asm** source file, Visual Studio will attempt to open and display its contents on the main screen:

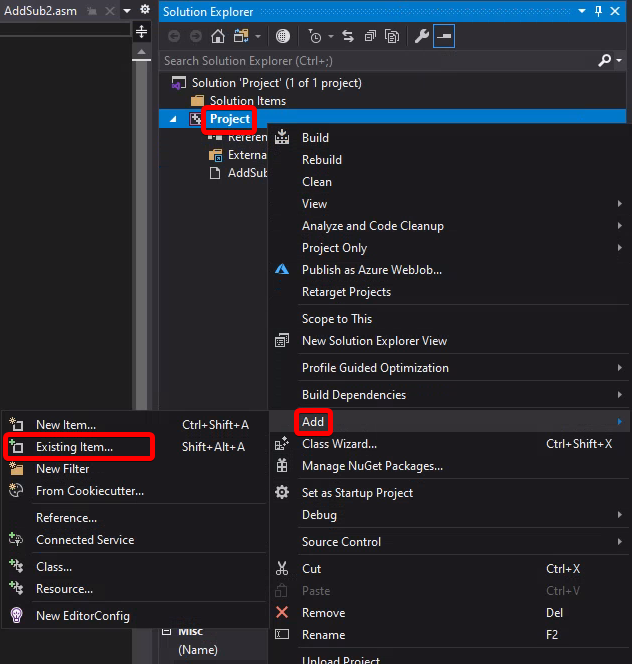


# Importing your own existing source code:

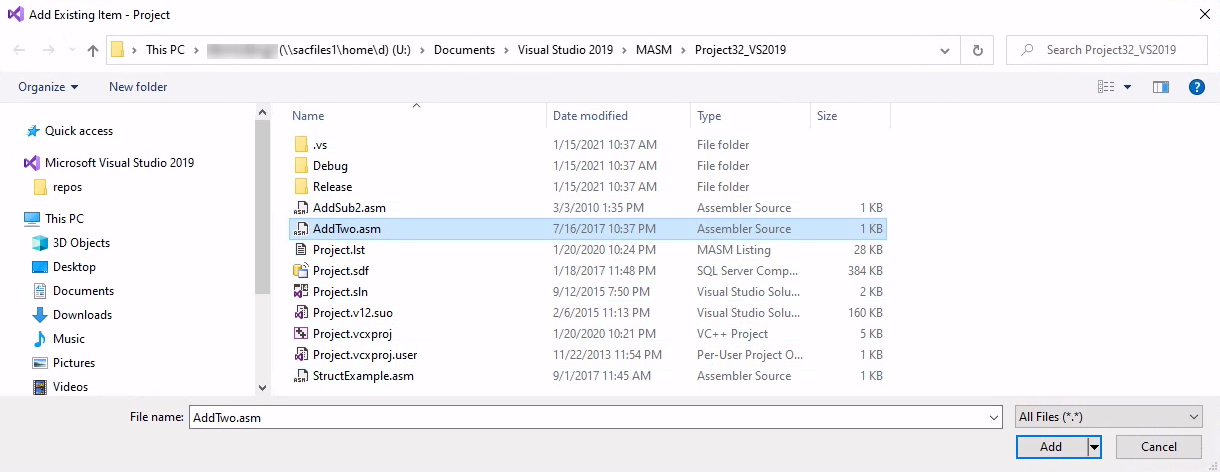
1. Copy and paste your own source code (.asm) to the respective Project folder in your U: drive.

* Please note: Make sure the source code supports the correct architecture. You must import a 32-bit .asm source code inside a 32-bit project. And you must import a 64-bit .asm source code inside a 64-bit project.

1. On the Solution Explorer Pane, under Solution, Right click on ‘Project’. Then click on Add>Existing Item...



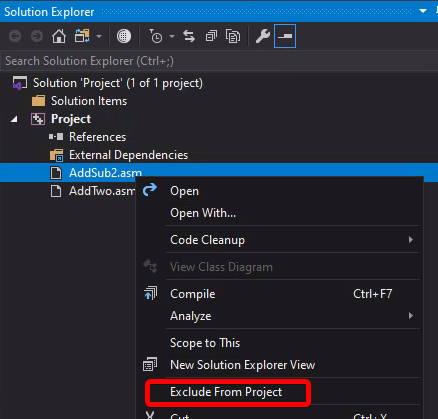
1. Then, browse to the **.asm** source code you would like to include inside the project inside the  
   **U:** drive directory that you copied earlier.



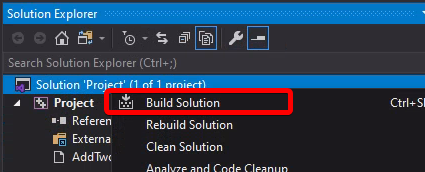
**Click Add** to add this source file to the project. It will appear inside the Solution Explorer.

1. Exclude any **.asm** source code that is not relevant in your project.

Navigate to the Solution Explorer pane. Right click on the .asm source code. Then click “**Exclude from Project**”.

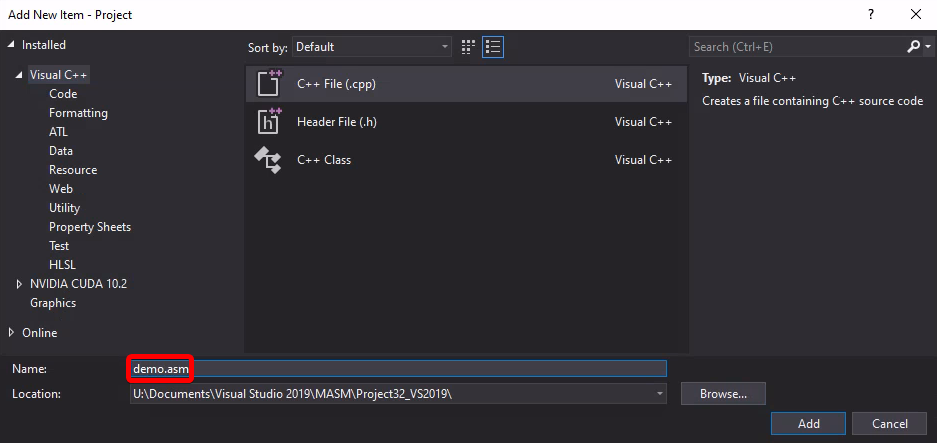


1. Then right click on **Solution > Build Solution** to ensure that you have the latest compilation of your project before you run your code.



# Creating your own source code

1. If you want to start from scratch, open up an existing 64-bit or 32-bit MASM project. On the Solution Explorer, right click on Project > Add > New item...
2. Replace your source file name ending with .cpp to ending with **.asm**

****

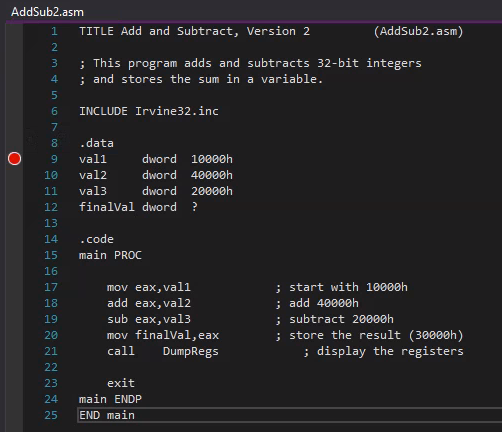
Then, exclude any files that are not part of your project. See [Step 4](#Excluding files from Project) from the previous section on how to do this.

# Debugging

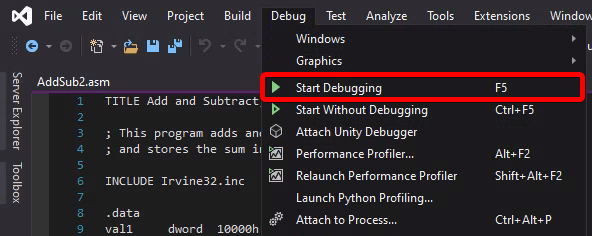
This section explores some useful features Visual Studio provides for debugging MASM projects.

If you have previously worked with IDEs similar to jGRASP or Eclipse, Visual Studio also has a feature to add breakpoints:

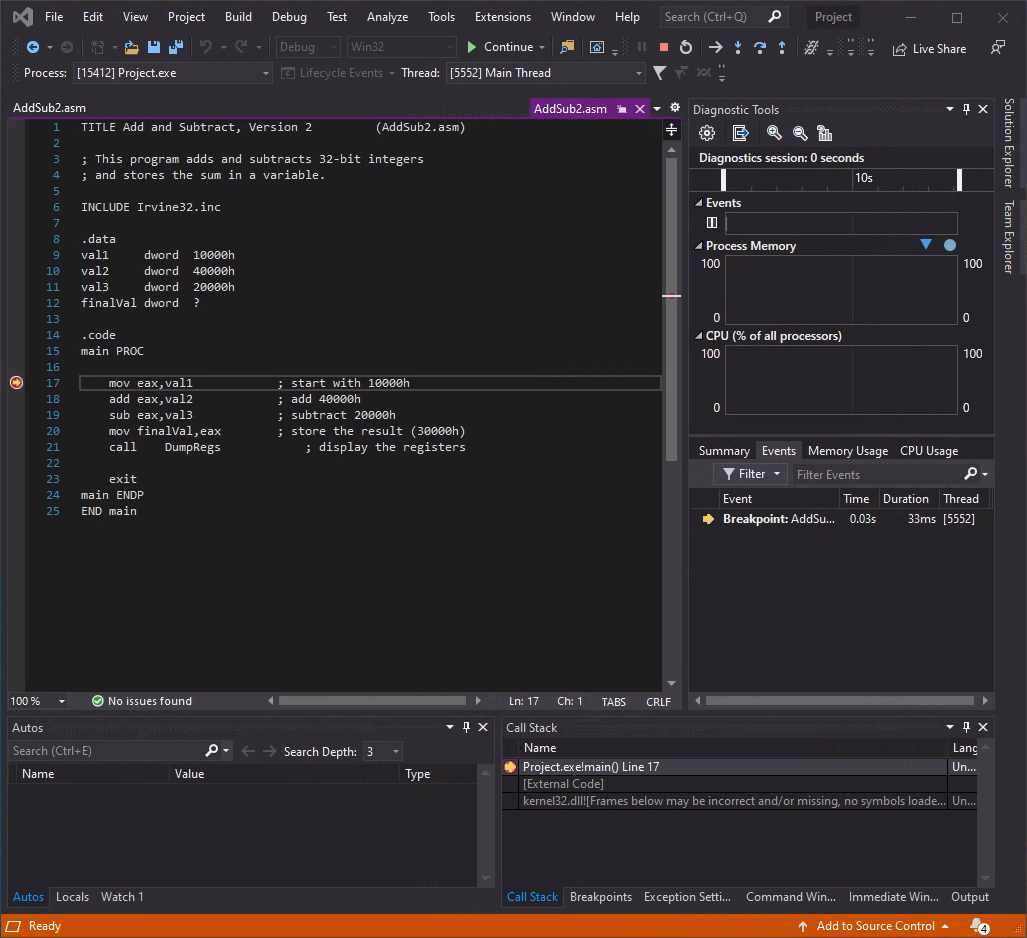
Click on the left side of each corresponding line number to add a breakpoint:



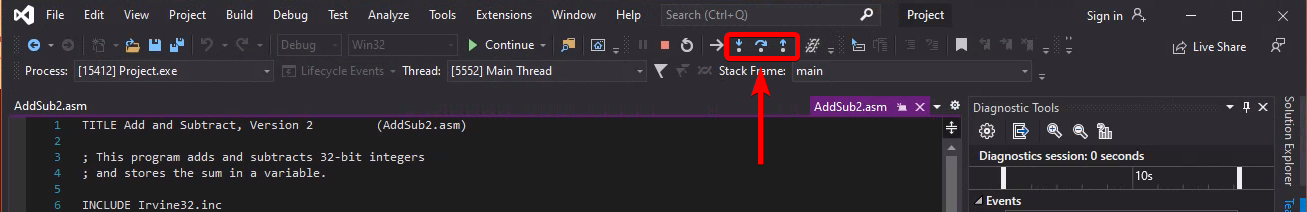
To start debugging, Navigate to Debug>Start Debugging:

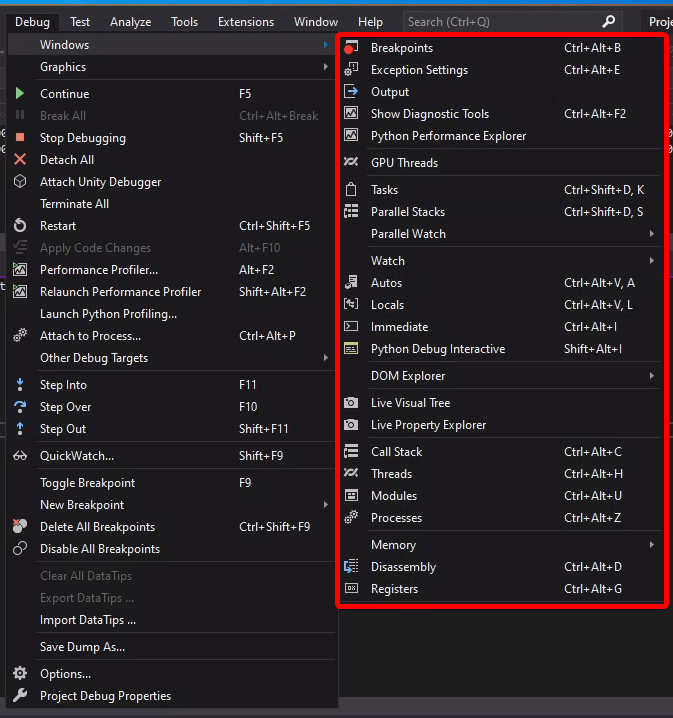


You will then notice that Visual Studio will start building your project. After it is done setting up your debug environment, you will see a similar screen to this:



You can then step-in, step-over, step-out of program execution. These controls are located at the top panel:

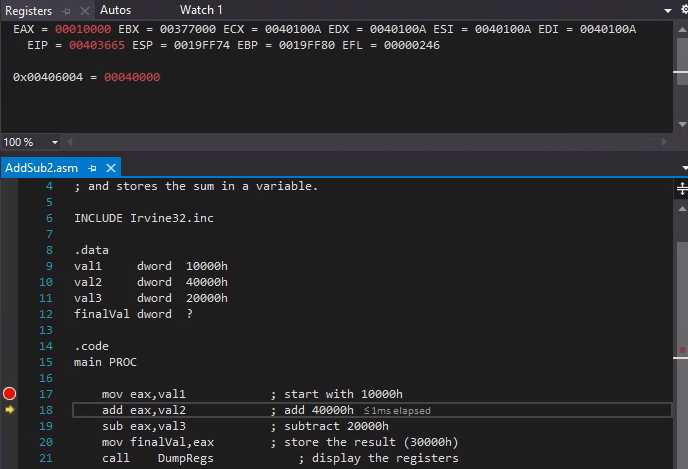
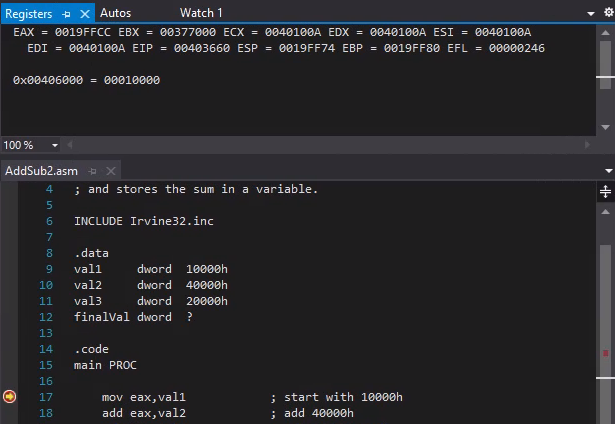


Under the Debug>Windows, you can add in ‘windows’ that can allow more visual about more analysis of the source code as instructions are executed.

I will not go in-depth through all of the debugging tools displayed here, as the purpose of this documentation is to get you started on your first Assembly projects through Visual Studio 2019.

A useful debugging tool window to keep open is the Registers window.   
(Debug > Windows > Registers)

This window gives a snapshot of which values are stored in each corresponding CPU register at the current breakpoint:



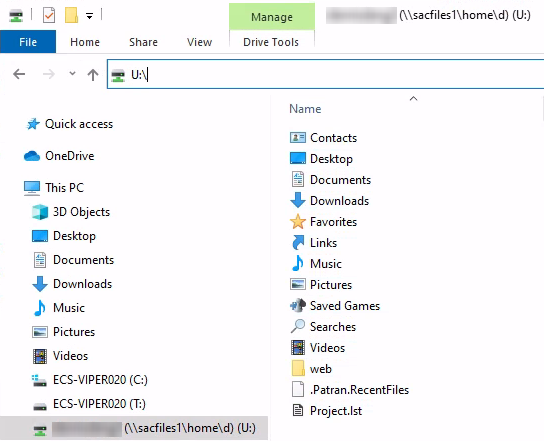
So, if I step-over to the next instruction, I expect to have the value ‘00010000’ stored in the EAX register:

# Additional Notes

## U: Drive Storage Space on ECS Lab Machines

Our remote lab machines on Sassafras are integrated with U: drive. This is a shared network space linked to your CSUS account. Any files you save under this U: directory (and any of its subfolders) will be synchronized and be carried over any Sassafras machines that you log into with your CSUS account.

You can find your U: drive by opening File Explorer. On the left-hand pane, you will see that there will be a network drive that you can access. It is typically named under your CSUS username. Alternatively, you can type in the address on File Explorer: ‘[U:\](file:///U:\)’ (No single quotes)

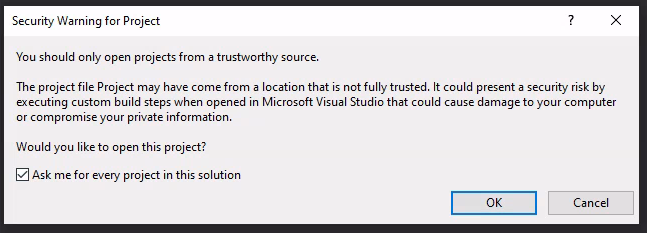
  
It is imperative that you store your projects on your U: Drive. If you save it anywhere else outside this directory on the computer, there’s no guarantee that it will still be there the next time you log in. All new files added in the C: drive will be wiped on a daily basis.

This storage solution is not absolutely necessary. You can save these files elsewhere, such as emailing these files to your self, saving it to a USB drive, uploading to Google Drive, etc. However, we recommend using U: Drive because we believe its a more convenient way of accessing your files between our lab machines.

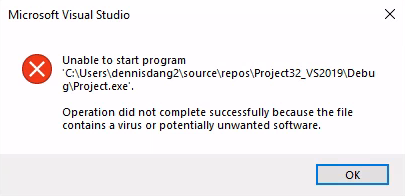
## Security Warnings

If you are importing several Microsoft Assembly (MASM) source code from the internet, Visual Studio and our antivirus programs may flag this as suspicious:

1. A Security Warning may pop up after trying to open the project. **Click OK to continue.**



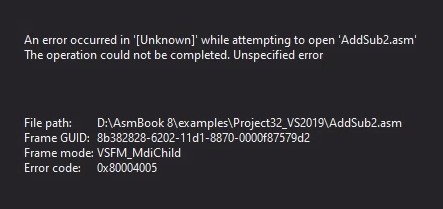
1. Upon running or debugging your source code, you may see a similar warning:



This is a known issue that we are working to solve. We witnessed this happening on one machine, but was unable to reproduce the same issue one the same machine and user session. If this happens, we recommend recopying the project folders and try rebuilding the Solution.

## Known Errors

1. Upon importing a sample project, you may see a similar error:



Please do not mind this at all. Upon opening up a project, Visual studio tends to open up the previous file that was last worked on. The error exists because it was trying to navigate to a non-existing file path, which appears to be a file structure from the source code developer’s machine.

Clicking on the .asm file on the Solution explorer will appear the source code on the main screen:

