Visual Studio Setup[[1]](#footnote-1)

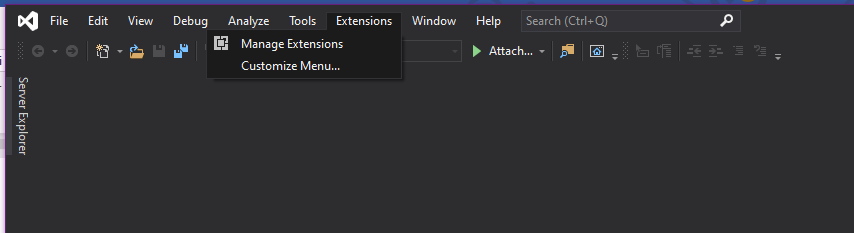
# Installing Visual Studio[[2]](#footnote-2)

Follow the installation guide found on Microsoft’s webpage: <https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2019>

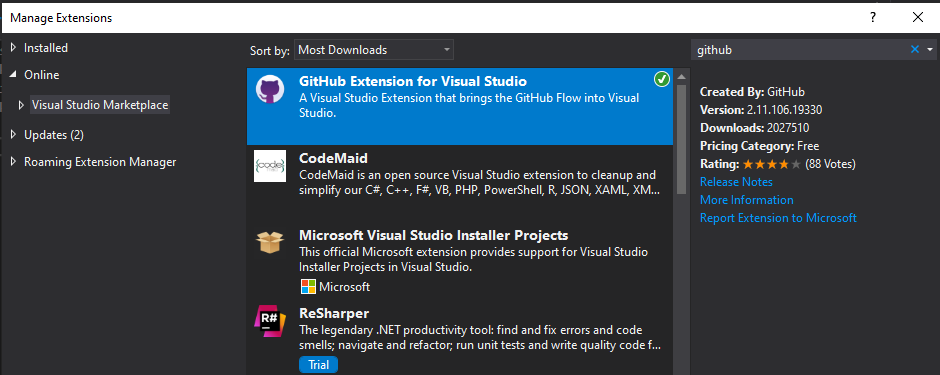
# Adding the GitHub extension

The codebase for this project exists on a GitHub repository in the cloud. As of writing this document, Dr. Tamer Omar hosts the latest codebase on his GitHub account, so ask him for the repository link to get started. Then follow these steps.

1. Open Visual Studio and click on the extensions tab. Select “Manage Extensions” from the tab dropdown

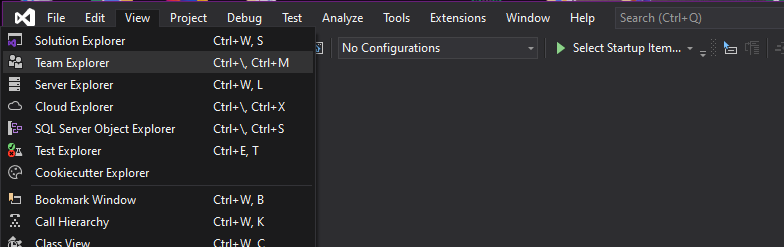


1. Type “github” in the search bar and select the GitHub extension

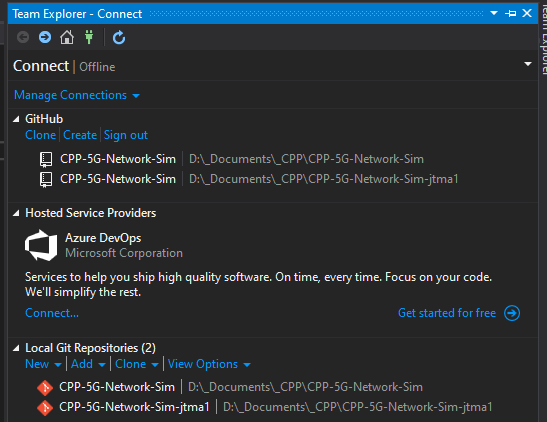


After installing the extension, you may have to restart Visual Studio for the changes to take effect.

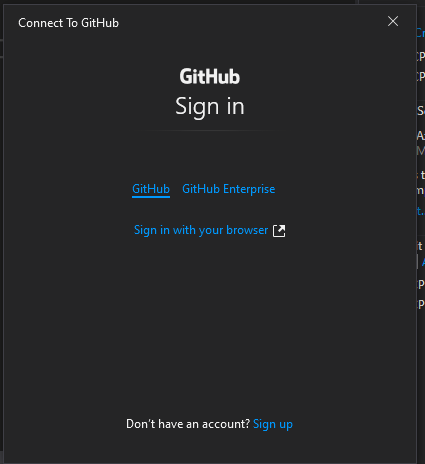
1. Navigate to View > Team Explorer



This will open the Team Explorer, from which you can connect to a GitHub repository. The explorer shows previous GitHub repositories you’ve connected to, as well as local repositories.

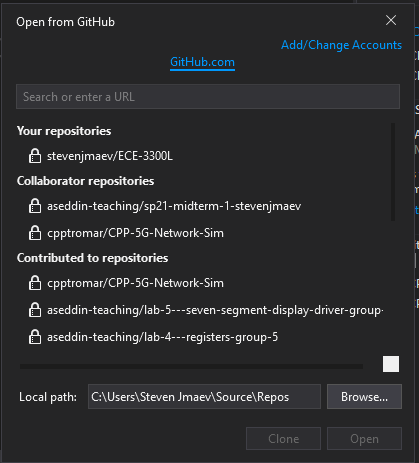
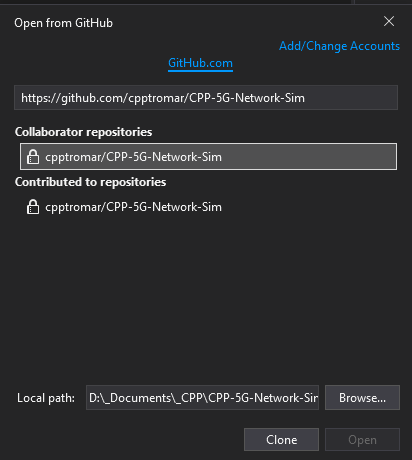


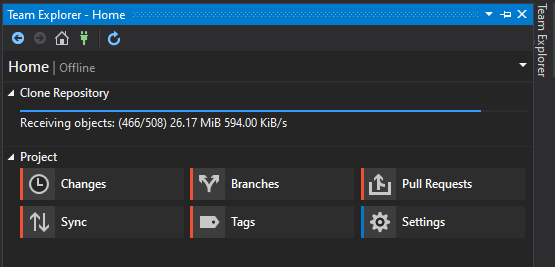
1. Click on Clone in the explorer above, and the following dialog box will appear. First, you need to sign into your GitHub account by selecting “Sign in with your browser.”



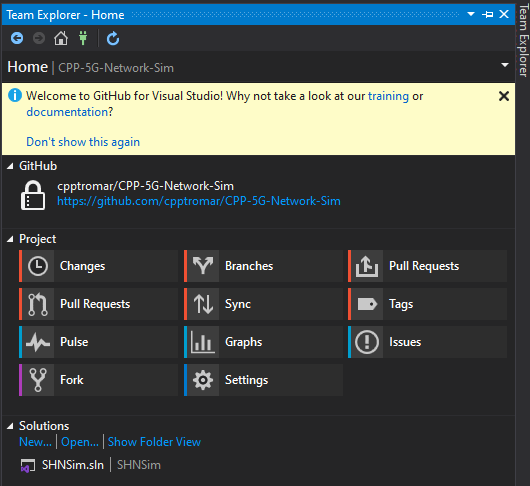
Once you sign in, the box will change to this. In the “Search of enter a URL” text box, input the URL of the repository of the project. In our case it is the 5G Network Simulator repository hosted on Dr. Tamer Omar’s GitHub.

Also, browse for the local path you want to copy the repository contents into. I recommend creating a folder called CPP-5G-Network-Sim or something similar.

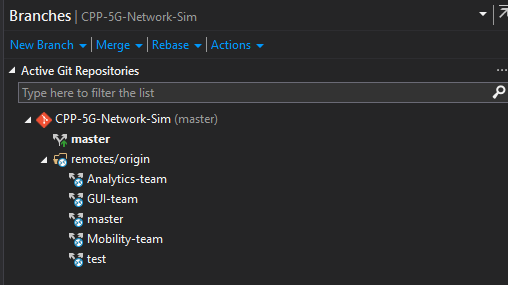
You should then see a progress bar like the following: 

When this finishes, you will see the Team Explorer Homepage look like this:



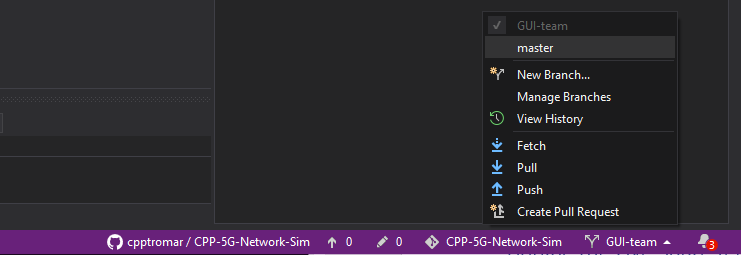
## Description of GitHub functionality in VS

### Branches

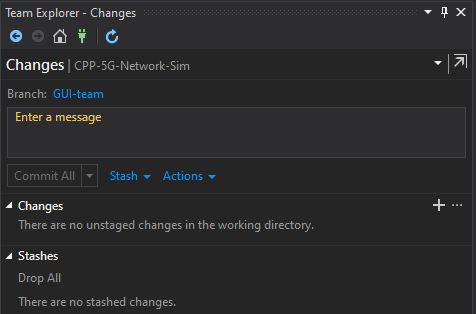


Normally, we want to make changes to the code that is in the branch assigned to us rather than the master branch. This way the master branch remains bug-free. To clone a specific branch, right-click on it and select “Checkout” from the resulting context menu.

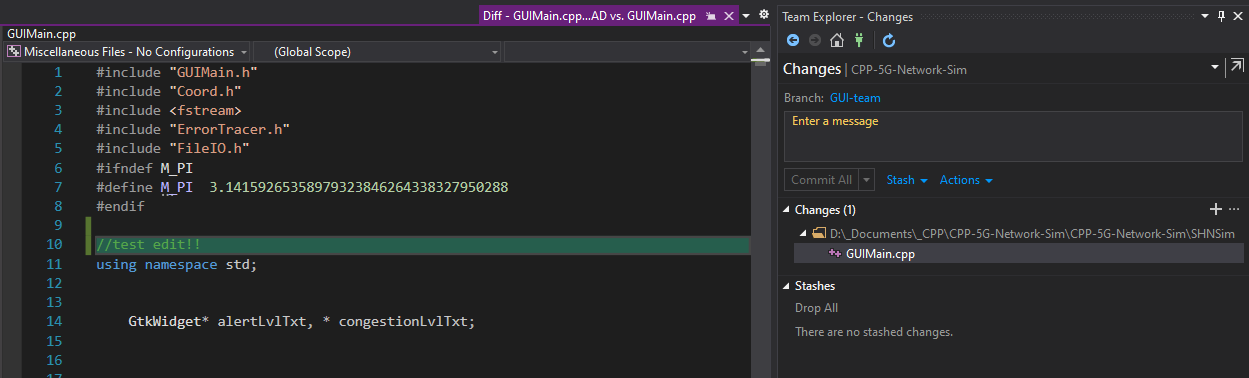
You can see which branch you are currently working on by viewing the bottom of the VS window. Here, you can also click on the branch and change the branch you are viewing/editing.



### Changes



Here we can see any changes that we’ve made since our last synchronization with the GitHub repository. An example test edit is shown below. We can click on the files that have changes and view the changes. We can then commit these changes to prepare them for synchronization.[[3]](#footnote-3)



# Adding and Using the Live Share Feature

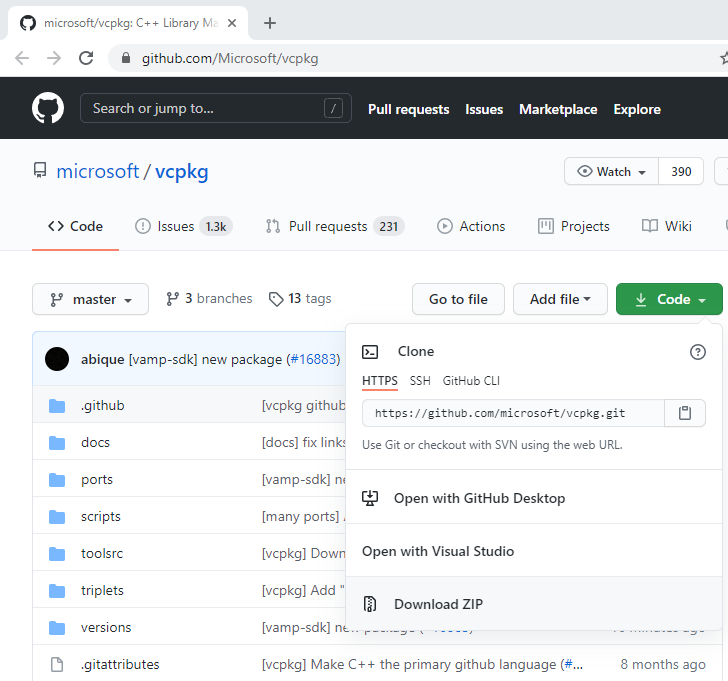
Visual studio includes this feature which allows groups to collaborate on the same solution simultaneously and view each other’s edits real-time.

Many of you may be familiar with Google Docs and/or OneDrive and their collaboration capabilities. Visual Studio Live Share is very similar to this, except that Live Share is **session-based,** and the edited file is not hosted on a web server, but on one of the collaborator’s local machines.

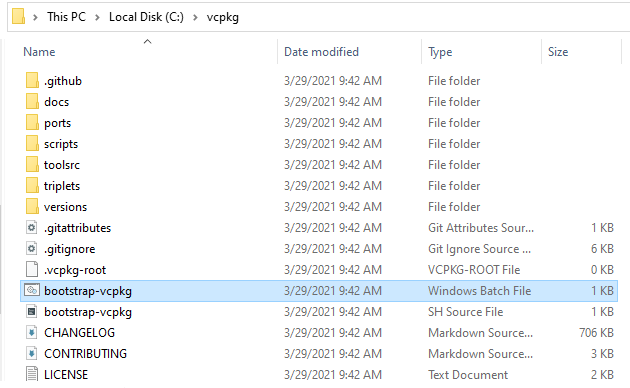
# Adding the libraries necessary for GUI development[[4]](#footnote-4)

To add extra libraries, we will be using the vcpkg manager described [here](https://docs.microsoft.com/en-us/cpp/build/vcpkg?view=msvc-160#how-to-get-and-use-vcpkg). This package manager is hosted on GitHub, and is easily installed by following these steps:

1. Visit the vcpkg GitHub page at <https://github.com/Microsoft/vcpkg>
2. Download the all the available files as ZIP



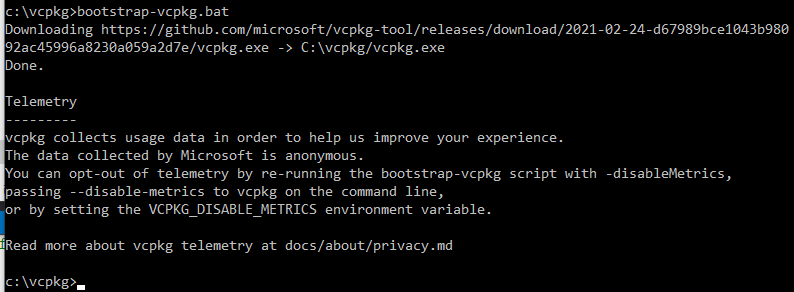
1. Unzip the downloaded file into a local folder, we recommend creating a folder on your host OS drive[[5]](#footnote-5) and call it “vcpkg”
2. Run the bootstrap-vcpkg.bat file that is in the newly created file. This will finish installing the package manager



You can also do this by navigating to this file via console and typing in the console:

bootstrap-vcpkg.bat.

If done this way, you will see a message like that shown below:



Now the package manager can be used, but we must first integrate it into Visual Studio so that it searched for the libraries installed using vcpkg.

1. Now that we have installed our package manager, we will install the following libraries which are used by the GUI portion of the codebase:

gtk Library for creating graphical user interfaces

cairo Library for 2D graphics and drawings

For references on these libraries, please refer to the following links:

<https://developer.gnome.org/gtk3/stable/>

<https://www.cairographics.org/>

**\*\*\*Before you proceed, you MUST disable any antivirus that is on your machine. An antivirus will prevent the subprocesses from accessing files that are necessary for installation!!\*\*\***

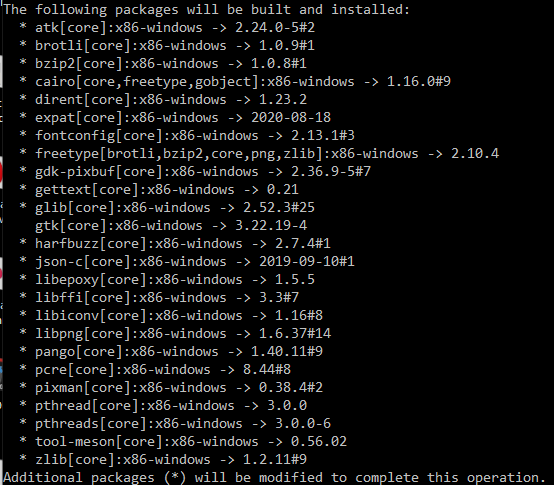
To install these libraries, run the following commands:

vcpkg install gtk (vcpkg install gtk:x64-windows) (for 64-bit Windows OS)

vcpkg install cairo

Note that each of these operations may take a while to complete, depending on your network speed, since these libraries depend on other libraries that the package manager must install first, and these can all have quite a large volume.

Note: when executed the first command to install GTK libraries, I saw this list of packages that vcpkg manager queued for installation:

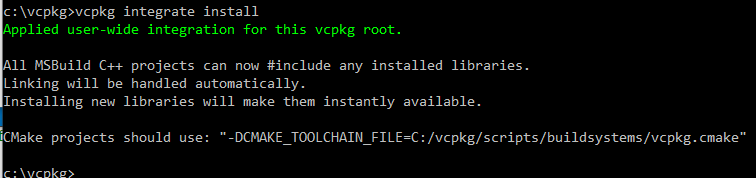


You can see that this list includes the gtk and the cairo libraries.

1. To use vcpkg with Visual Studio, run the following command[[6]](#footnote-6):

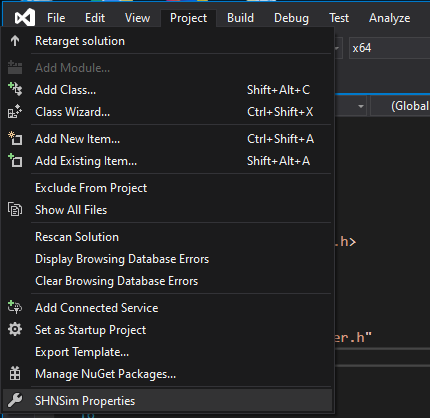
vcpkg integrate install

A successful run will return the following message.

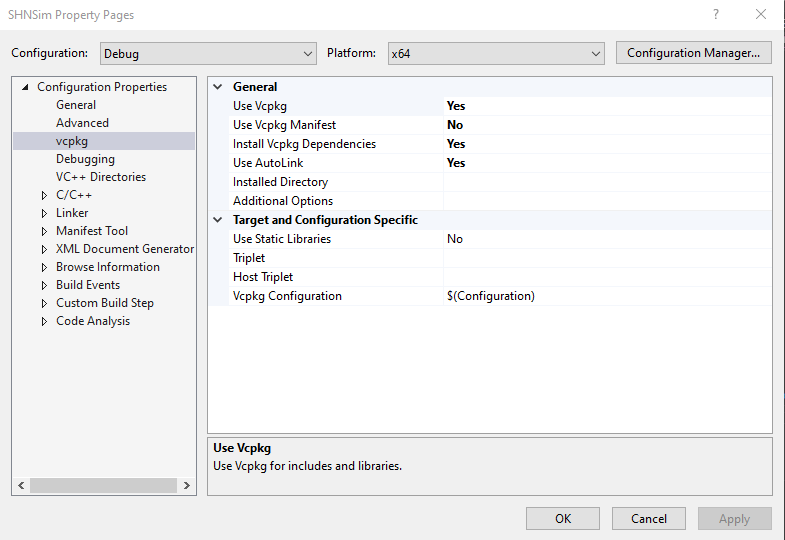


1. Make sure that Visual Studio is using the vcpkg manager to find files. Go to:

Project > Properties > vcpkg



Then set all the following fields to match those below (Use vcpkg, Use Vcpkg Manifest, Install Vcpkg Dependencies, Use AutoLink).



1. For more documentation on using the vcpkg manager, refer to the Microsoft webpage at <https://docs.microsoft.com/en-us/cpp/build/vcpkg?view=msvc-160#how-to-get-and-use-vcpkg>.

1. This guide is based on the 2019 Community version of Visual Studio, steps may be slightly different for any different or newer versions. [↑](#footnote-ref-1)
2. Many of you may already have MS Visual Studio installed, so this step might be unnecessary. [↑](#footnote-ref-2)
3. Before we can commit our changes, we must enter a message. This message is a summary of the main changes made to the codebase. (in the example shown, we may enter a message such as “Added a test edit in the form of a comment.”) [↑](#footnote-ref-3)
4. If your group will not be developing the GUI, this step is not very helpful or necessary. Also, this step is not required for successful compilation of the C++ code, though it allows us to use Visual Studio’s Intellisense functionality to help us when writing new code. [↑](#footnote-ref-4)
5. For Windows, the OS drive letter is typically C:/ [↑](#footnote-ref-5)
6. If you are using Windows PowerShell, you must add a “./” so you have “./vcpkg” [↑](#footnote-ref-6)