IN4089 - Data Visualization - Project 2: Volume Visualization

Install Guide Windows (Visual Studio)

Note: This install guide has been tested with Windows 10 and Visual Studio 2019 and 2022. Variations might work, but you will be responsible for adjusting the setup yourself.

⚠️ **Please use pathes/folders for the framework and vcpkg without spaces or special characters.** ⚠️

Third party software

Download Microsoft Visual Studio 2019 or 2022 Community Edition from:
`https://visualstudio.microsoft.com/free-developer-offers/`.

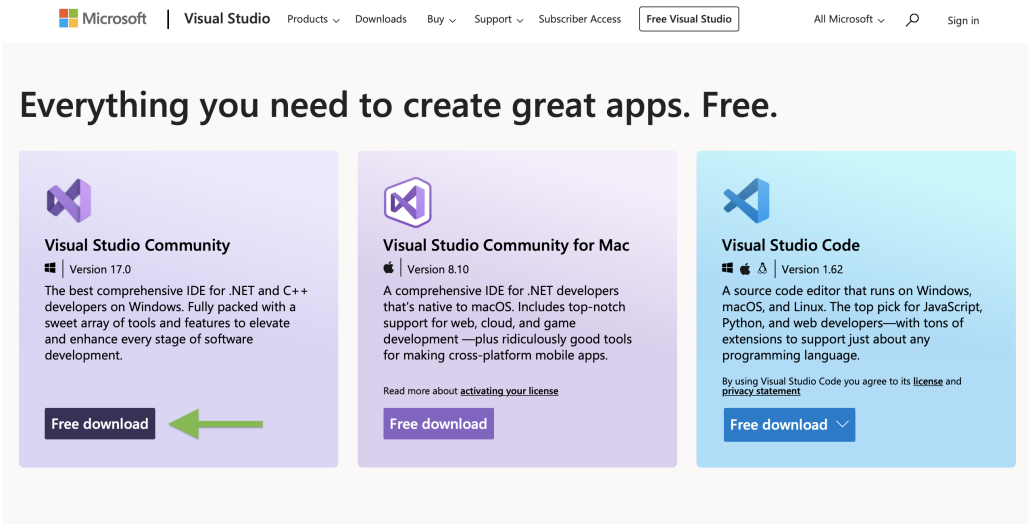

Figure 1: Download Visual Studio Community.

In the window asking about workloads, enable Desktop development with C++, then click install.

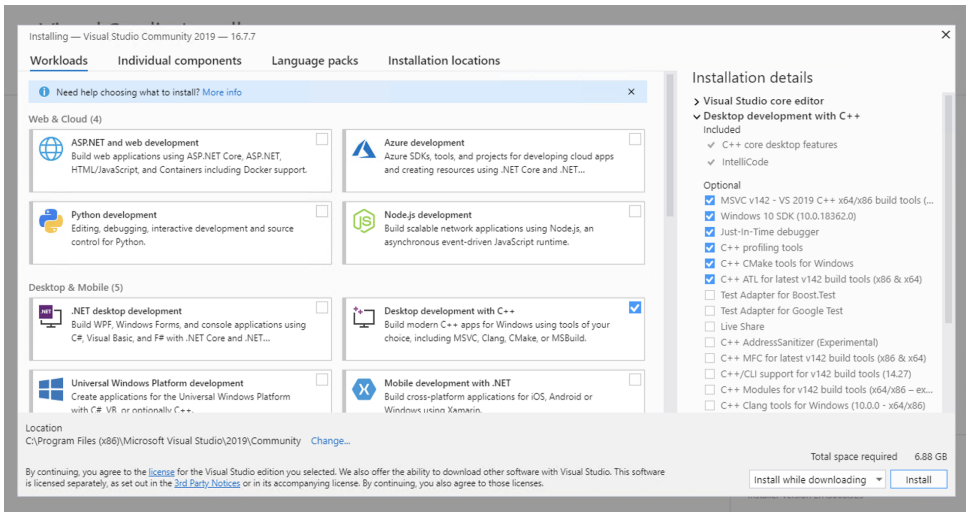

Figure 2: Enable Desktop development with C++.

When you open Visual Studio for the first time you are asked to sign-in. Although Visual Studio Community Edition is free, it requires a sign-in to keep using it after 30 days. You can use your TU Delft credentials to sign into Visual Studio.

We will need git versioning software to install the vcpkg package manager. Install git using the official installer: `https://git-scm.com/`. Make sure to select the PATH option as shown in Figure 3 (it should be the default).

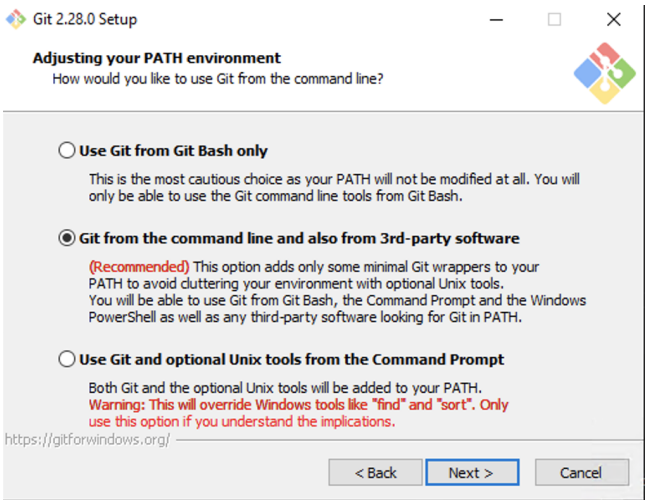

Figure 3: Enable git for 3rd party software.

vcpkg

Next, we will install vcpkg, a package manager that provides 3rd party libraries needed for the provided framework. Open Command Prompt as administrator (search for command prompt, right click on command prompt and click run as administrator). All text in `monospace` below is a command in the commmand prompt. Navigate to the folder where you want to store vcpkg (avoid spaces and special characters in the path to this folder) Note: this path has no relation to the path where you put the volume visualization framework. Then run the following commands:

- Clone vcpkg from github
  `git clone https://github.com/microsoft/vcpkg.git`
- Change into the checked out directory
  `cd vcpkg`
- Inside the directory, compile vcpkg
  `bootstrap-vcpkg.bat -disableMetrics`
- Now we enable vcpkg integration in Visual Studio.
  `vcpkg integrate install`
  If you receive an error like

  ```
  Error: Failed to copy file:
  C:\Users\<account>\Development\vcpkg\vcpkg\scripts\buildsystems\tmp\vcpkg.user.targets ->
  C:\Users\<account>\AppData\Local\vcpkg/vcpkg.user.targets
  ```

  manually create the folder C:\Users\<account>\AppData\Local\vcpkg and run the command again.

Building/Running the Framework

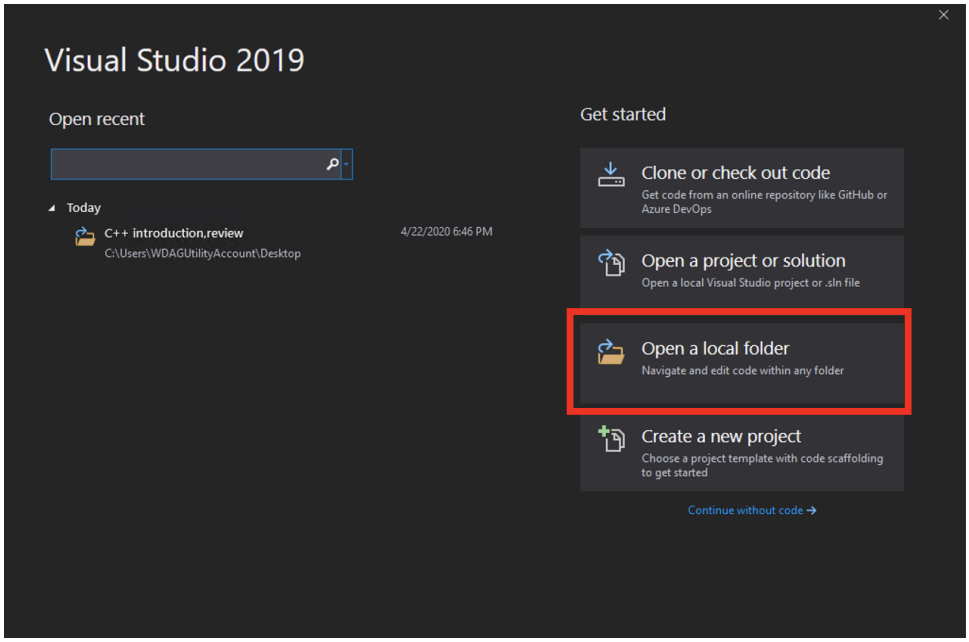

Figure 4: Open Visual Studio 2019/2022 from the start menu and in the window that appears, click on *Open a local folder*

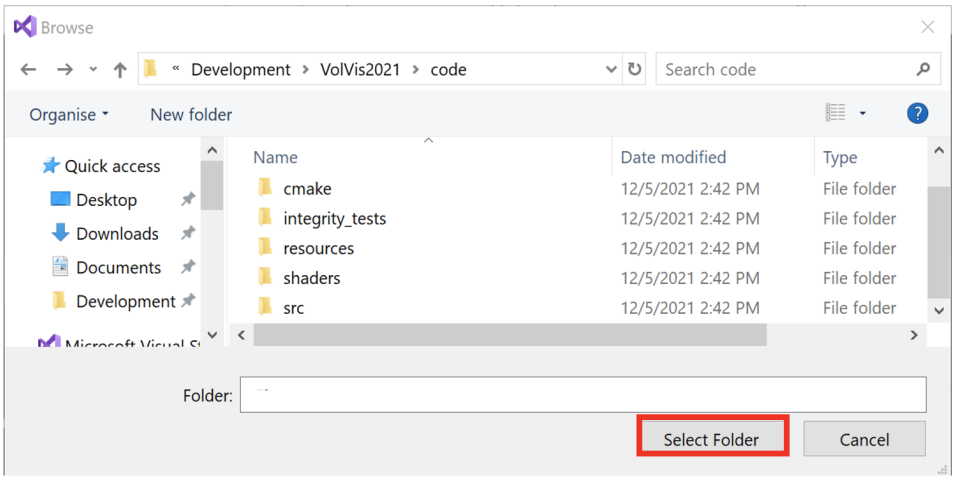

Figure 5: Navigate to the root of the project and click *Select Folder*

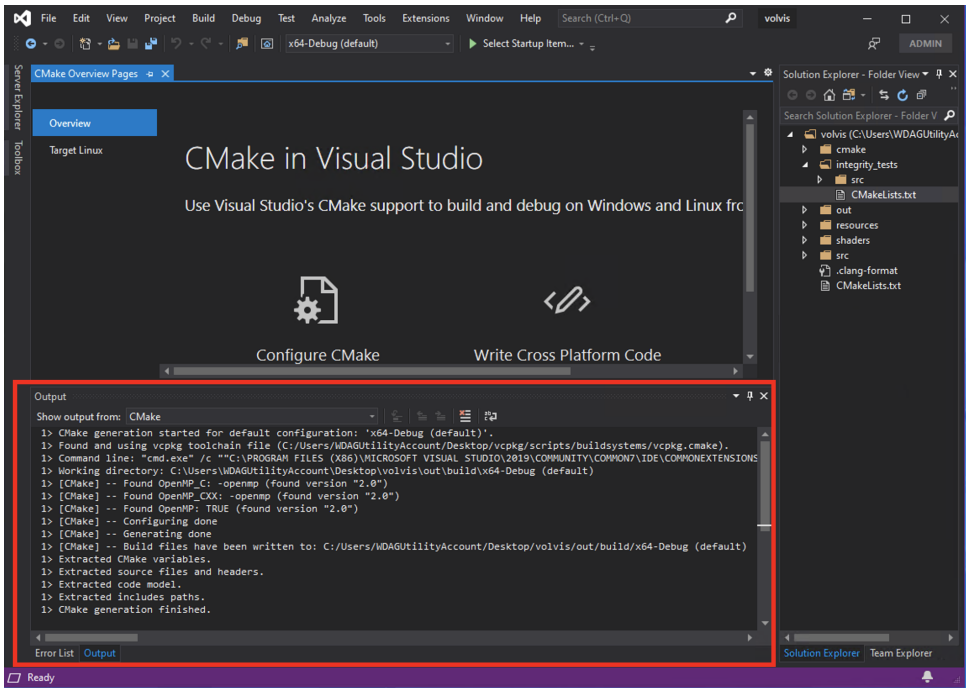

Figure 6: When you open the project for the first time, the CMake build system will configure the compiler and look for the third-party libraries that we installed with vcpkg. Wait for the output window to show CMake generation finished.

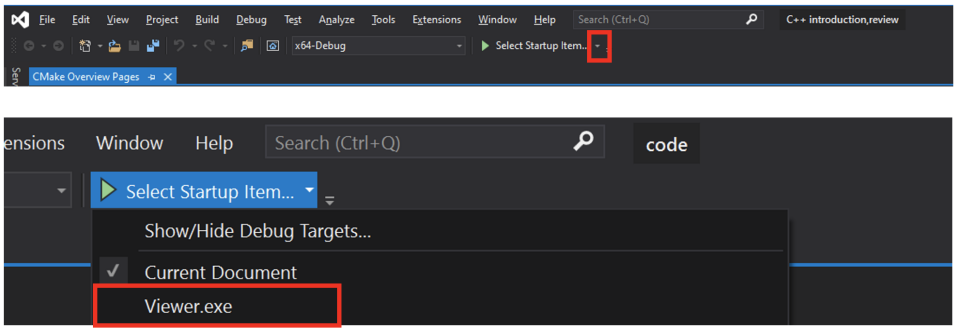

Figure 7: Select the executable that you want to compile and run. Click on the little down arrow next to *Select Startup Item...* and click on [NameOfExecutable].exe, in our case *Viewer.exe*.

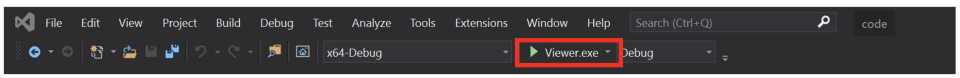

Figure 8: Press the green button to compile and run the program. This will run the application in *Debug Mode* allowing you to set breakpoints and analyze your code and issues. However, this is also a bit slower than it could be. Check the *VolVis Enable Release Mode.pdf* to see how to change to a faster runtime.