IN4089 - Data Visualization - Project 2: Volume Visualization

Install Guide macOS (Visual Studio Code)

Note: This install guide has been tested with macOS 12 and Visual Studio Code. Variations might work, but you will be responsible for adjusting the setup yourself.

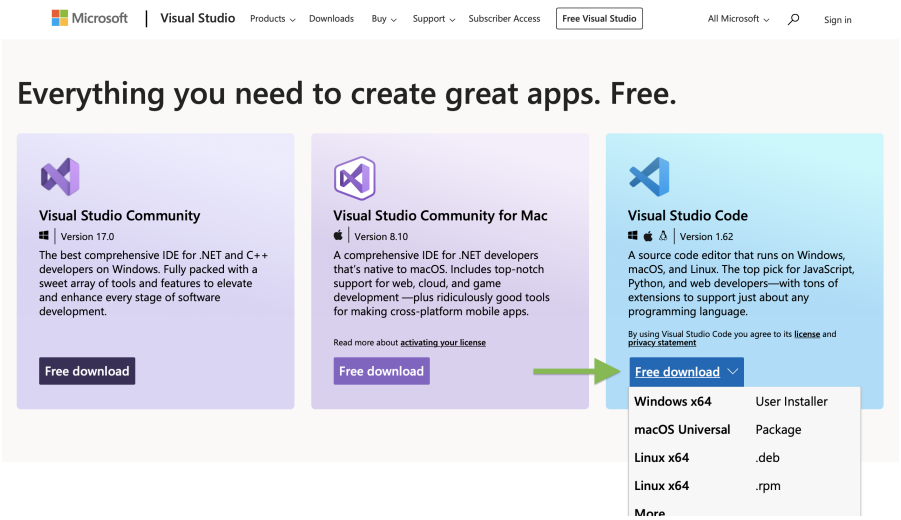
⚠️ **Please use pathes/folders for the framework and vcpkg without spaces or special characters.** ⚠️

Third party software

C++ compiler. Install the clang C++ compiler, in case you do not already have it (i.e. you already have XCode installed) To check if you already have clang installed open a macOS Terminal and run
`clang --version`
If clang is installed you will get an output starting similar to
`Apple clang version 13.0.0 (clang-1300.0.29.3)`
If clang is not installed, you can install the developer tools via
`xcode-select --install`

CMake. Install CMake 3.22.0 or newer from `https://cmake.org/download/` Drag the GUI executable into your applications folder. If you do so, you command line executable will be in the following folder (you will need this later to configure VS Code) `/Applications/CMake.app/Contents/bin/`

Visual Studio Code. Download Microsoft Visual Studio Code from:
`https://visualstudio.microsoft.com/free-developer-offers/`.



vcpkg

Next, we will install vcpkg, a package manager that provides 3rd party libraries needed for the provided framework. Open a Terminal.app. 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.sh -disableMetrics`
- Now we enable vcpkg integration in Visual Studio Code. Run
  `./vcpkg integrate install`
  and note down the output after CMake projects should use: (similar to the red box below)

```
mathijs@ubuntu:~/vcpkg$ ./vcpkg integrate install
Applied user-wide integration for this vcpkg root.

CMake projects should use: "-DCMAKE_TOOLCHAIN_FILE=/home/mathijs/vcpkg/scripts/buildsystems/vcpkg.cmake"
```

- If with the new command you get an error like
  `vcpgk cannot create /Users/<your_username>/.vcpkg/vcpkg.path.txt`
  create the folder manually by typing
  `mkdir /Users/<your_username>/.vcpkg/` (replace <your_username> with your actual username as in the error message)
  and re-run
  `./vcpkg integrate install`

Building/Running the Framework

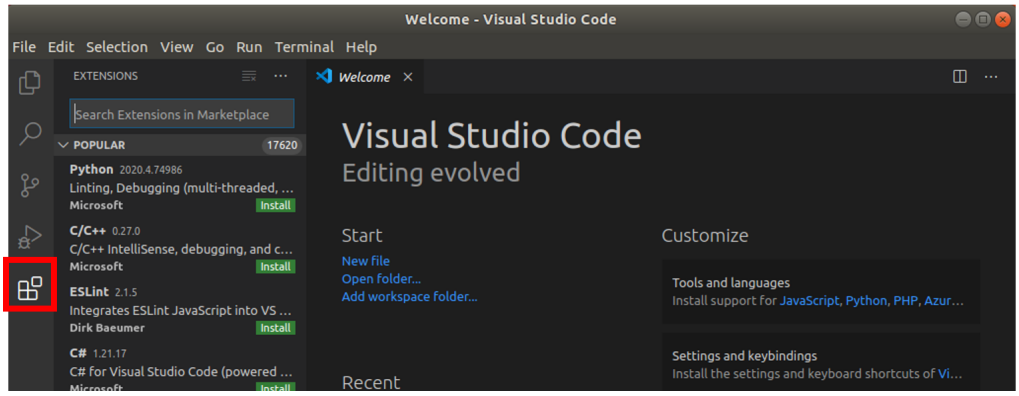


Figure 1: Open Visual Studio Code and click the framed icon to install the "C/C++" and "CMake Tools" extensions by Microsoft

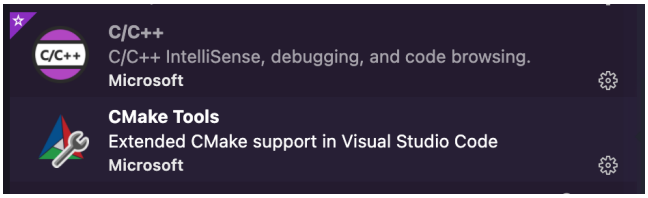


Figure 2: The "C/C++" and "CMake Tools" extensions by Microsoft

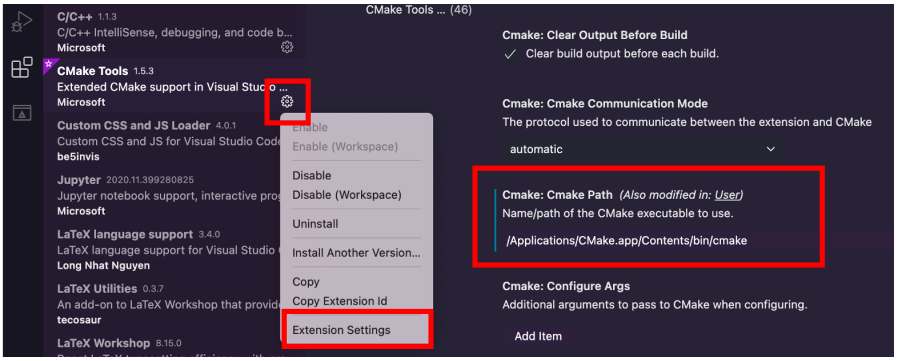


Figure 3: You need to set the folder of the cmake command line folder (mentioned above) in the CMake Tools settings

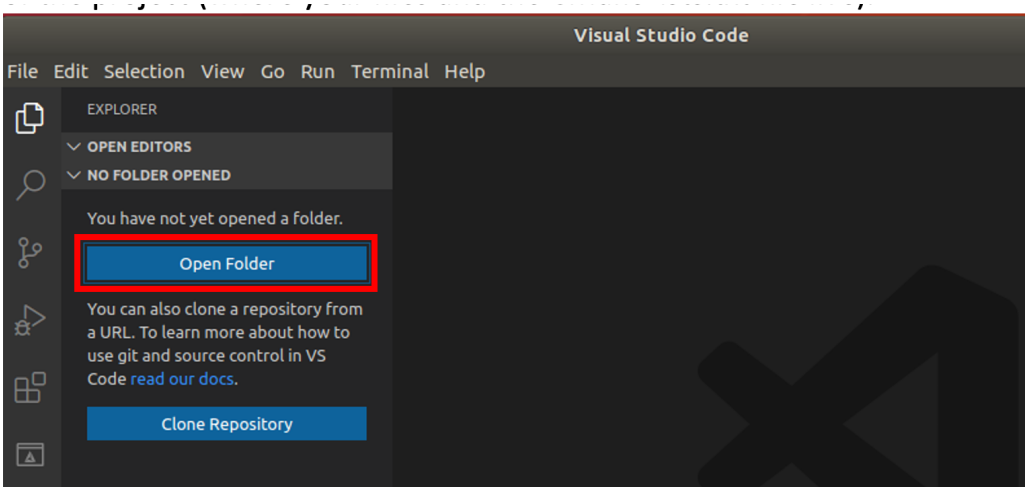


Figure 4: In the main view click on *Open Folder* (or *File* ⇒ *Open Folder*) and select the root of the project framework (where your files and the *CMakeLists.txt* file live). (See below for what the content of the folder should look like.)

Name	Date Modified	Size	Kind
cmake	Yesterday at 13:29	--	Folder
CMakeLists.txt	Yesterday at 13:30	3 KB	Plain Text
integrity_tests	Yesterday at 13:29	--	Folder
resources	Yesterday at 13:29	--	Folder
shaders	Yesterday at 13:29	--	Folder
src	Yesterday at 13:30	--	Folder
vcpkg.json	Yesterday at 13:30	286 bytes	JSON Document

Figure 5: The content of the folder you should open in Visual Studio Code.

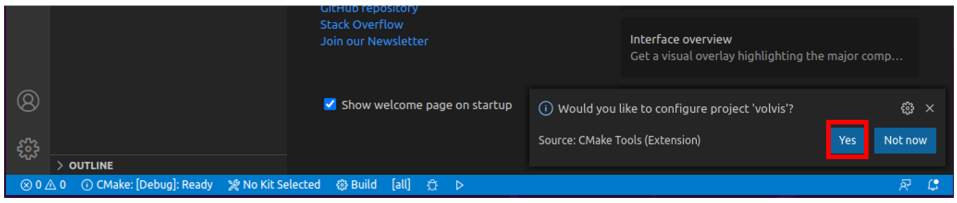


Figure 6: After opening the folder, a popup should show up asking to configure the project. Click *Yes*.

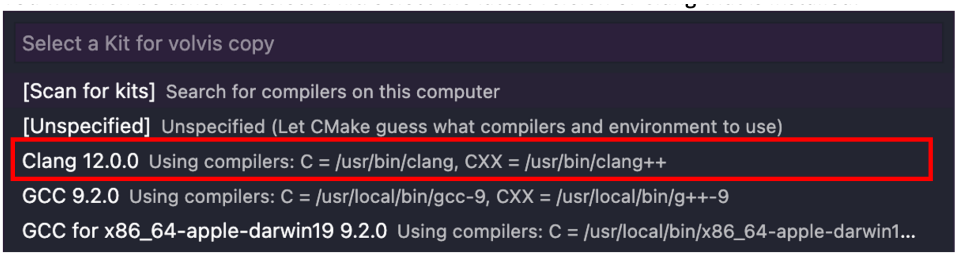


Figure 7: You will then be asked to select a kit. Select the latest version of Clang that is installed.

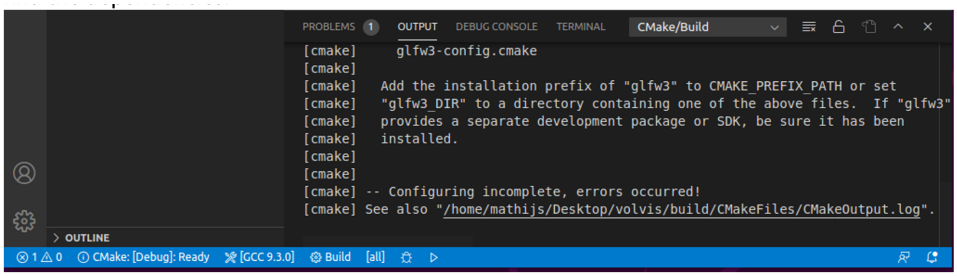


Figure 8: CMake will now start configuring your project. Initially, this will fail because it cannot find the dependencies.

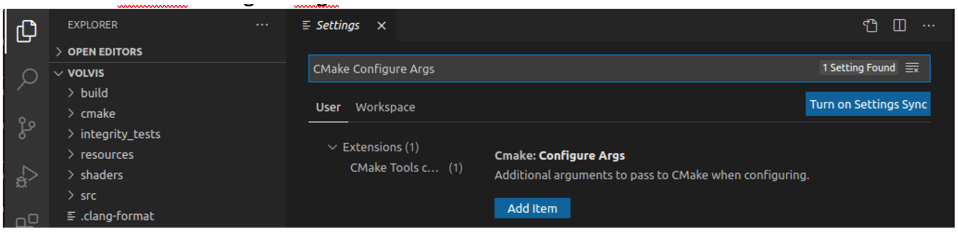


Figure 9: Open the settings menu by going to Code ⇒ Preferences ⇒ Settings, and search for *CMake Configure Args*.

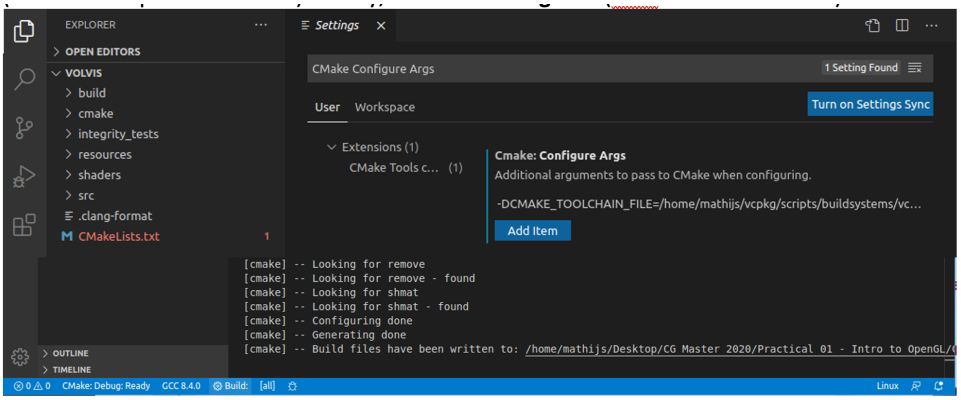


Figure 10: Click the *Add Item* and paste the text (without the quotation marks) that you after running `vcpkg integrate install` during the installation of vcpkg. **Then save the settings file** (File ⇒ Safe).

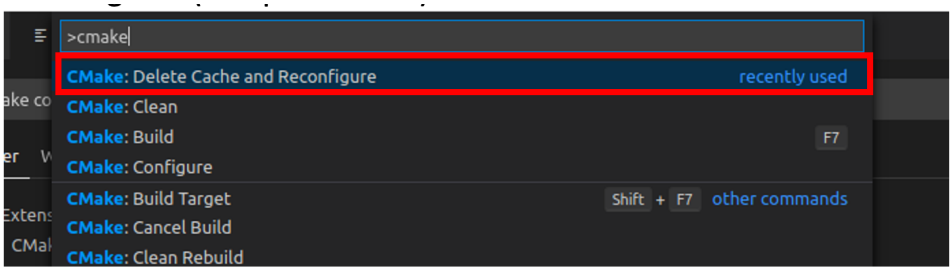


Figure 11: Now we will reconfigure the project with the changes we just made. Open the Command Palette (`cmd + shift + p`) and search for *CMake: Delete Cache and Reconfigure* (and press enter).

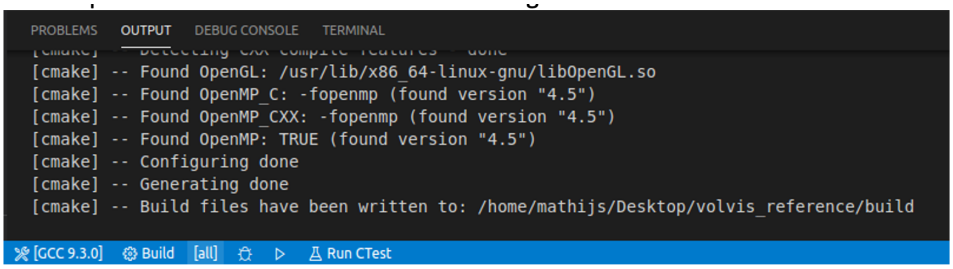


Figure 12: The output window should now look something like this.

Figure 13: The project has two targets: *IntegrityTests* and *Viewer* (the application). To build and run Viewer, we first need to select it. In the blue bottom bar, click on *[all]* and then select *Viewer* in the menu that pops up at the top.

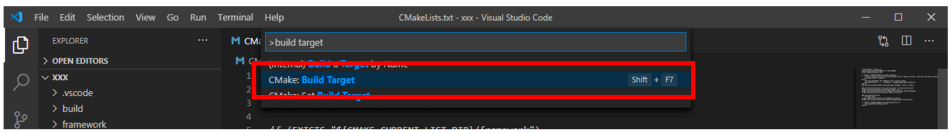


Figure 14: To compile the project, open the Command Palette (`cmd + shift + p`) and search for *CMake: Build Target* or use the shortcut `shift + F7`.