COMP 5411: Advanced Computer Graphics

OpenGL Introduction

We have prepared a demo program for you to get hands on OpenGL. We use

- GLFW (http://www.glfw.org/) for window management,
- CMake (https://cmake.org/) for generating make files or IDE project files.

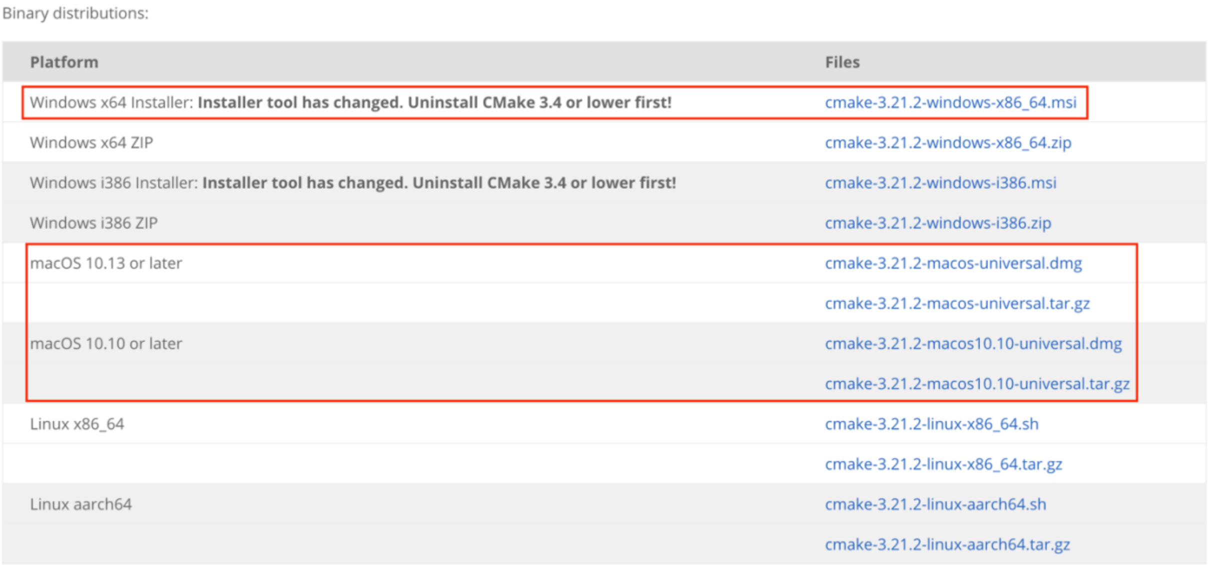
The code has been tested on Windows and macOS. This demo serves the following purposes:

- showing how to use GLFW for window management;
- showing how to use the callback mechanism in GLFW for user interactions;
- showing how to construct/manipulate self-maintained Modelview and Projection matrices;
- showing how to use shaders for rendering in modern programmable OpenGL pipeline.

Please follow the instructions to compile source files of the demo before obtaining the executable program.

1 Preparation

Please install CMake if it does not exist in your operating system. You can go to https://cmake.org/download/ for obtaining the installation file (see Fig. 1).



Binary distributions:

Platform	Files
Windows x64 Installer: **Installer tool has changed. Uninstall CMake 3.4 or lower first!**	cmake-3.21.2-windows-x86_64.msi
Windows x64 ZIP	cmake-3.21.2-windows-x86_64.zip
Windows i386 Installer: **Installer tool has changed. Uninstall CMake 3.4 or lower first!**	cmake-3.21.2-windows-i386.msi
Windows i386 ZIP	cmake-3.21.2-windows-i386.zip
macOS 10.13 or later	cmake-3.21.2-macos-universal.dmg
	cmake-3.21.2-macos-universal.tar.gz
macOS 10.10 or later	cmake-3.21.2-macos10.10-universal.dmg
	cmake-3.21.2-macos10.10-universal.tar.gz
Linux x86_64	cmake-3.21.2-linux-x86_64.sh
	cmake-3.21.2-linux-x86_64.tar.gz
Linux aarch64	cmake-3.21.2-linux-aarch64.sh
	cmake-3.21.2-linux-aarch64.tar.gz

Figure 1: Select appropriate CMake installer depending on your Operating System.

Besides, make sure you also have a C++ compiler (and an IDE) installed. For example, on Windows, **Visual Studio** 2015 or above is preferable. (Note: please do select the C++ compiler option during the VS installation.)

If you prefer a more light-weight, easy-to-use, cross-platform supported IDE, you can also try **Visual Studio Code** (free to download at https://code.visualstudio.com/). If you choose this approach, you will need to install the C++ compiler separately, following instructions at https://code.visualstudio.com/docs/languages/cpp (look for section **Example: Install MinGW-x64**).

Instructions for both **Visual Studio** (2015 or above) and **Visual Studio Code** are provided below, whatever you prefer. Instructions for **Visual Studio Code** will be pretty much the same on Windows and MacOS.

2 Download

Source files of the OpenGL demo are located at https://course.cse.ust.hk/comp5411/ogl_beginner/OpenGLIntro.zip. Download the zipped file and extract its content into a folder, for example, named ~/demo.

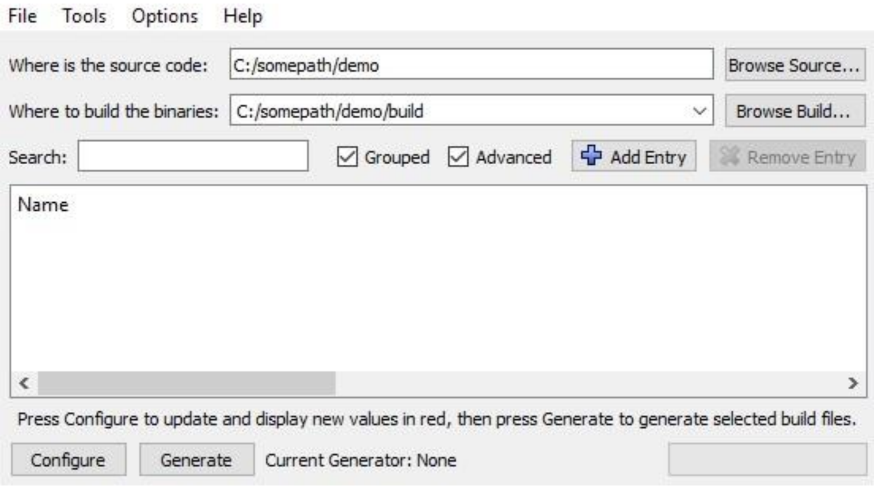
3 Compilation

Here are the instructions for compiling the source files on Windows and macOS.

3.1 Windows

If you use **Visual Studio** (2015 or above):

1. Open CMake and fill in the paths. We can use a folder named **~/demo/build** for storing make files or IDE project files (see Fig. 2).



File Tools Options Help

Where is the source code: C:/somepath/demo

Where to build the binaries: C:/somepath/demo/build

Search:

☑ Grouped ☑ Advanced

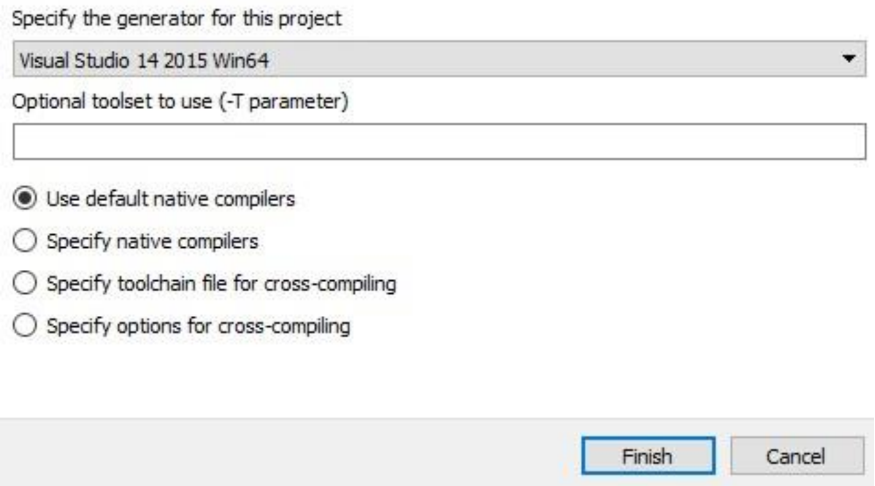
Name

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

Figure 2: Windows: fill CMake paths.

2. Press the button **Configure** and choose the compiler you have installed, then press the button **Finish** (see Fig. 3).



Specify the generator for this project

Visual Studio 14 2015 Win64

Optional toolset to use (-T parameter)

● Use default native compilers

○ Specify native compilers

○ Specify toolchain file for cross-compiling

○ Specify options for cross-compiling

Figure 3: Windows: choose the compiler.

3. After the configuration is done, press the button **Generate**.

4. Go to the folder **~/demo/build** and double-click the file OpenGLDemo.sln.

5. In the opened Visual Studio, you may want to change the build type to Release for better performance. Then click the project **OpenGLDemo** in the Solution Explorer using the right mouse button and select Build (see Fig. 4).

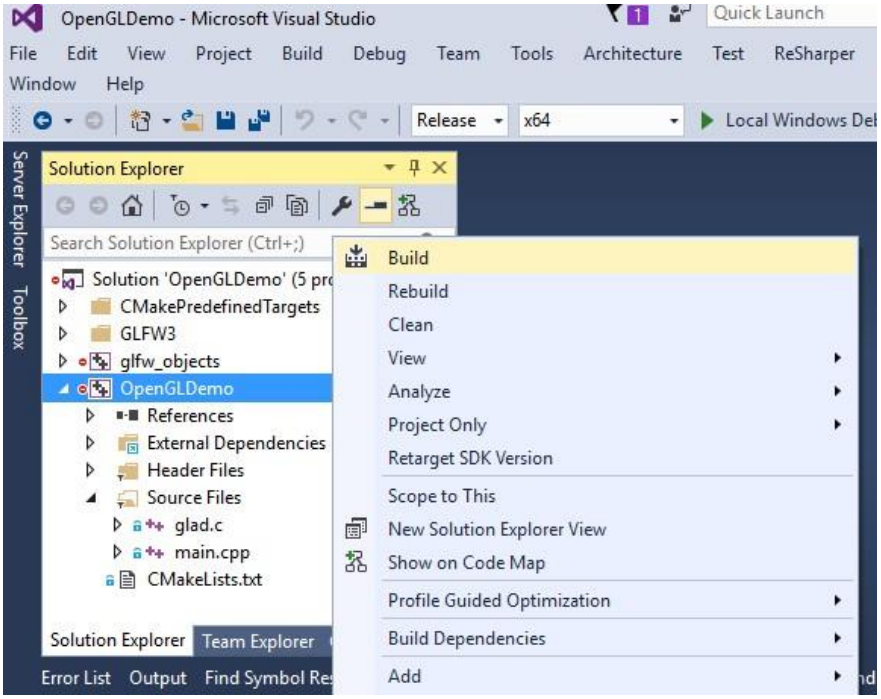


Figure 4: Windows: build the project.

6. After the build is done, go to the folder ~/demo/build/Release for obtaining the executable program. Open the program and follow the instructions printed in the console for interactions (see Fig. 5).

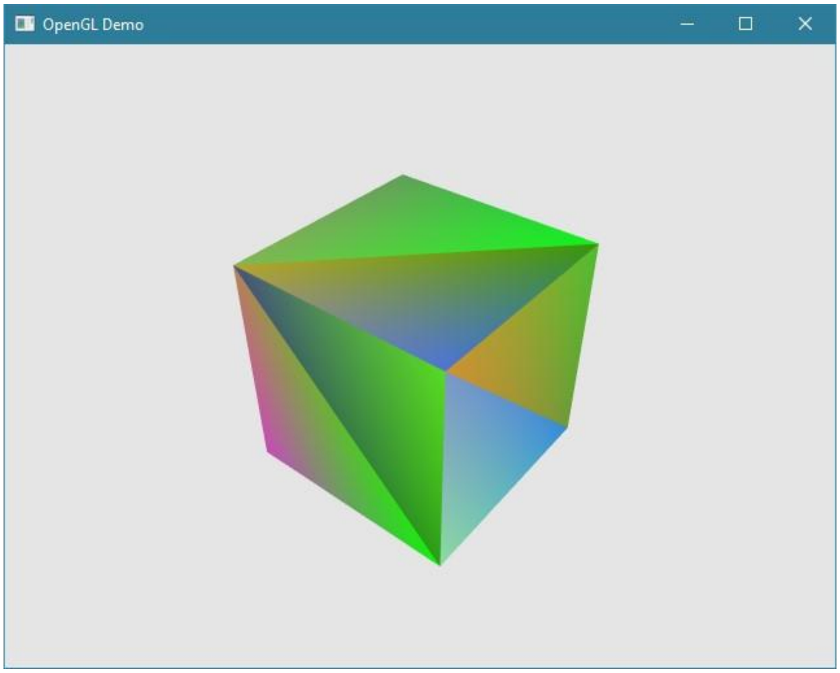
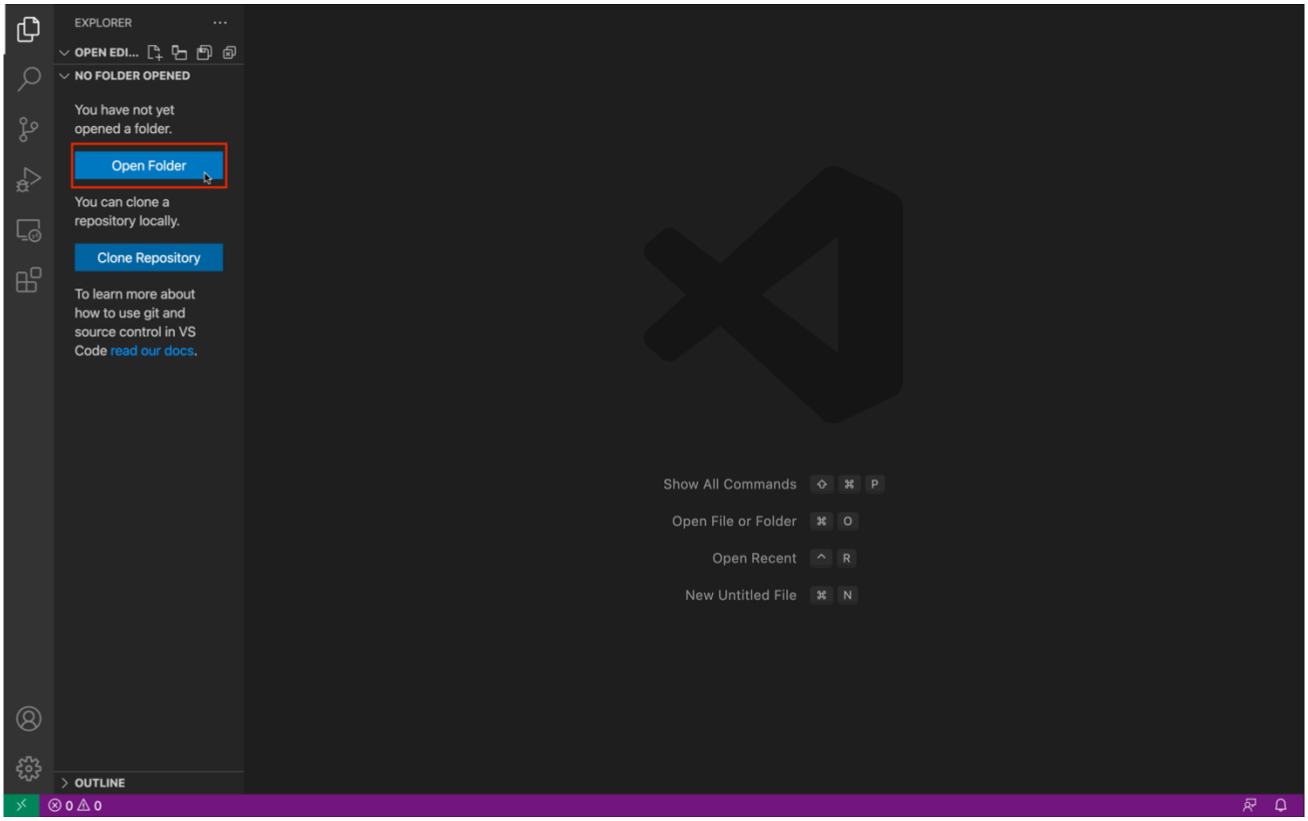


Figure 5: A rotating colored cube.

If you use **Visual Studio Code (also works on MacOS)**:

1. Open Visual Studio Code IDE and import the project folder (see Fig. 6):



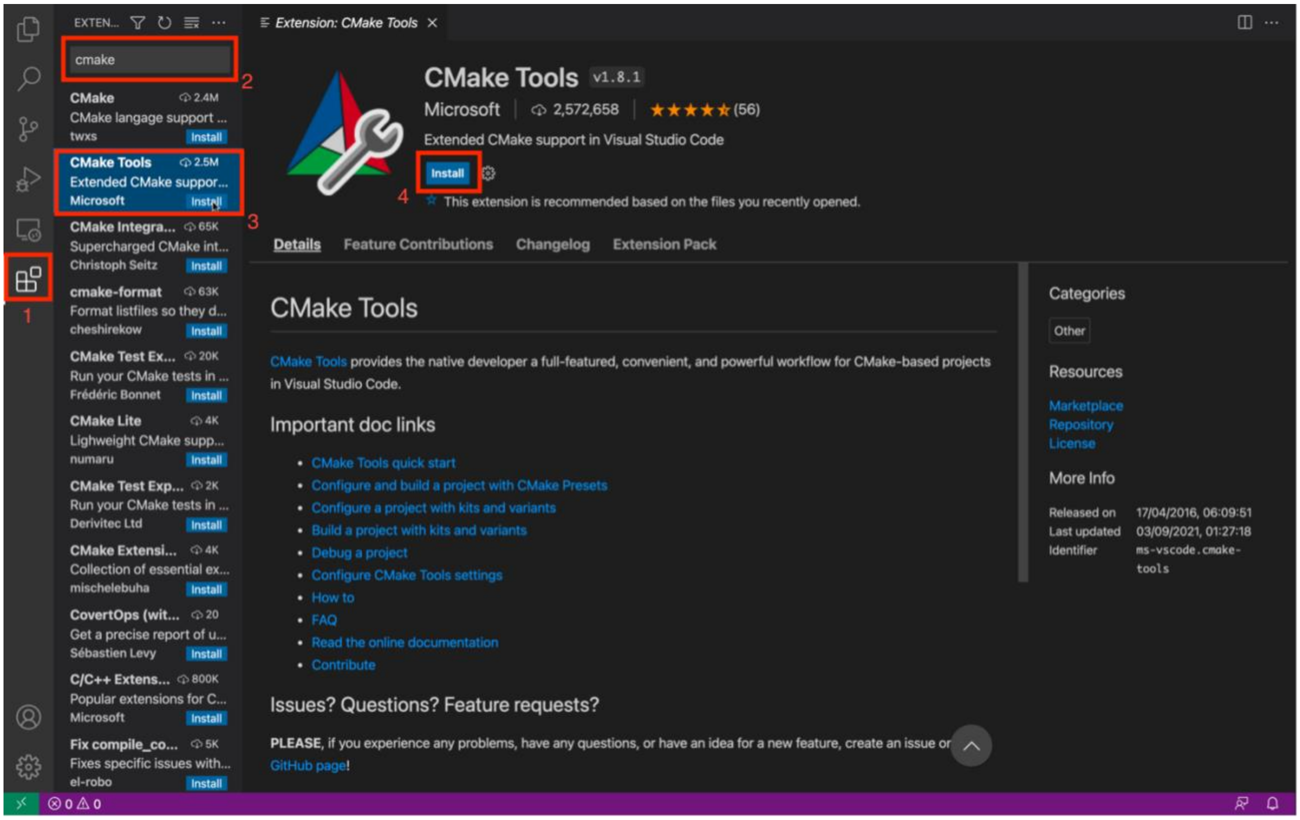
You have not yet opened a folder.

You can clone a repository locally.

To learn more about how to use git and source control in VS Code read our docs.

Figure 6: Import project folder into Visual Studio Code IDE.

2. In the Extensions panel on the left, search for 'cmake', then select and install the **Cmake Tools** (see Fig. 7). After installation is done, you will need to re-open the IDE in order to use the extension.



CMake ⤓ 2.4M
CMake langage support ...
twxs Install

CMake Tools ⤓ 2.5M
Extended CMake suppor...
Microsoft Install

CMake Integra... ⤓ 65K
Supercharged CMake int...
Christoph Seitz Install

cmake-format ⤓ 63K
Format listfiles so they d...
cheshirekow Install

CMake Test Ex... ⤓ 20K
Run your CMake tests in ...
Frédéric Bonnet Install

CMake Lite ⤓ 4K
Lighweight CMake supp...
numaru Install

CMake Test Exp... ⤓ 2K
Run your CMake tests in ...
Derivitec Ltd Install

CMake Extensi... ⤓ 4K
Collection of essential ex...
mischelebuha Install

CovertOps (wit... ⤓ 20
Get a precise report of u...
Sébastien Levy Install

C/C++ Extens... ⤓ 800K
Popular extensions for C...
Microsoft Install

Fix compile_co... ⤓ 5K
Fixes specific issues with...
el-robo Install

CMake Tools v1.8.1
Microsoft | ⤓ 2,572,658 | ★★★★★ (56)
Extended CMake support in Visual Studio Code

This extension is recommended based on the files you recently opened.

CMake Tools

CMake Tools provides the native developer a full-featured, convenient, and powerful workflow for CMake-based projects in Visual Studio Code.

Important doc links

- CMake Tools quick start
- Configure and build a project with CMake Presets
- Configure a project with kits and variants
- Build a project with kits and variants
- Debug a project
- Configure CMake Tools settings
- How to
- FAQ
- Read the online documentation
- Contribute

Issues? Questions? Feature requests?

PLEASE, if you experience any problems, have any questions, or have an idea for a new feature, create an issue on GitHub page!

Categories

Other

Resources

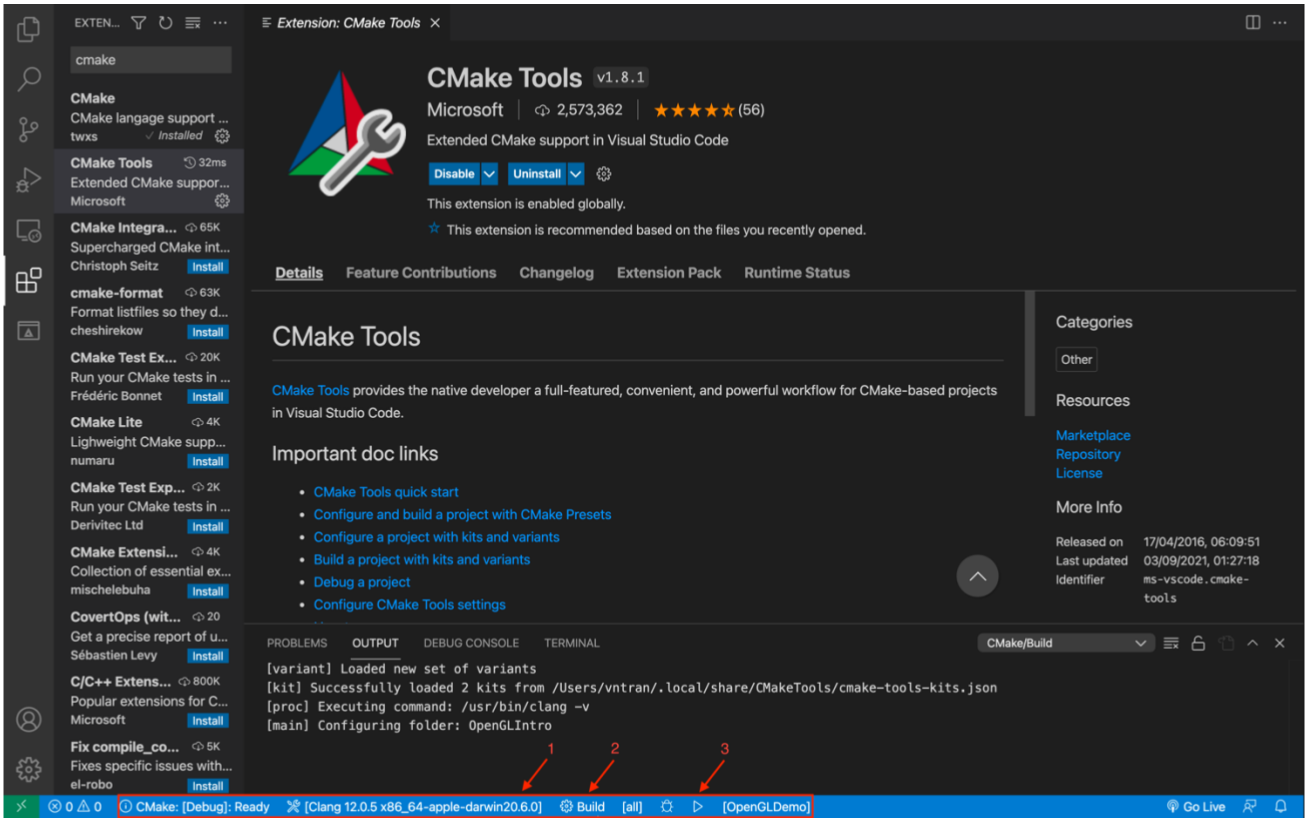
Marketplace
Repository
License

More Info

Released on	17/04/2016, 06:09:51
Last updated	03/09/2021, 01:27:18
Identifier	ms-vscode.cmake-tools

Figure 7: Install Cmake Tools extension within Visual Studio Code IDE.

2. You will now see a new toolbar at the bottom of the screen. Clicking on these icons will allow you to configure and build the project using the CMake extension that you've just installed (see Fig. 8). Click on these icons in that order (1->2->3), and you will have your application up and running.



Extended CMake support in Visual Studio Code

This extension is enabled globally.

This extension is recommended based on the files you recently opened.

CMake Tools

CMake Tools provides the native developer a full-featured, convenient, and powerful workflow for CMake-based projects in Visual Studio Code.

Important doc links

- CMake Tools quick start
- Configure and build a project with CMake Presets
- Configure a project with kits and variants
- Build a project with kits and variants
- Debug a project
- Configure CMake Tools settings

Released on 17/04/2016, 06:09:51
Last updated 03/09/2021, 01:27:18
Identifier ms-vscode.cmake-tools

```
[variant] Loaded new set of variants
[kit] Successfully loaded 2 kits from /Users/vntran/.local/share/CMakeTools/cmake-tools-kits.json
[proc] Executing command: /usr/bin/clang -v
[main] Configuring folder: OpenGLIntro
```

Figure 8. CMake extension toolbar. (1) Select the C++ compiler; (2) Build the project; (3) Run the application after building.

3.2 macOS (using command-line method)

1. Open a Terminal and go to the folder ~/demo
2. Create a folder named build: ’mkdir build’
3. Go to the created folder: ’cd build’
4. Run CMake command: ’cmake ..’
5. After the configuration is done, run Make command: ’make’
6. After the compilation is done, open the executable program named OpenGLDemo and follow the instructions printed in the console for interactions.

4 Other Resources

For more detailed OpenGL programming tutorials, you can visit this website http://www.opengl-tutorial.org/ or google other websites.

5 Q & A

If you encounter any problems during the above compilation process, please read this section first before reaching out for help (and Google is always your best friend). :)

Q1: I use Windows and I have installed Visual Studio. There is an error saying that ”No CMAKE CXX COMPILER could be found” after I press the configure button in CMake.

A1: You did not install the C++ compiler during the installation of Visual Studio, which is an option that you need to select and check.

Q2: I use Windows. There is an error related to ”MSBuild.exe” after I press the configure button in CMake.

A2: One possible reason is that you did not install the C++ compiler during the installation of Visual Studio. Please re-install Visual Studio with the C++ compiler option checked. Another possible reason is you selected the wrong generator that does not match your Visual Studio version. Please select the generator in CMake carefully.

Q3: I use Windows. Why are there items marked as red in the middle panel of CMake window, even though no errors are reported in the log panel of the window?

A3: The items being marked as red mean you can custom them, not errors. Of course, you do not need to change them at all.

Q4: I use macOS. Why is OpenGLDemo.sln not generated after I run CMake?

A4: If you use macOS, please strictly follow the instructions in Section 3.2. OpenGLDemo.sln is generated for Visual Studio on Windows platform. So please do NOT follow the instructions that do NOT match your OS.

Q5: I use macOS, but CMake command is not found when I run it in a Terminal.

A5: Please install the Terminal version of CMake via Brew (https://brew.sh/), which is a fantastic package manager on macOS. After Brew is installed, open a Terminal window and run the following command to install CMake: `brew install cmake`