

# Cmake package를 이용한 OpenGL Setting Tutorial

- 문서의 목적:
  - MSVC 2019 x64 (Microsoft Visual Studio 2019 64bit Windows) 버전  
외의 컴파일러를 이용, OpenGL programming을 할 때 필요한 개발환경  
설정
    - 윈도우에서 다른 VS compiler 사용 (VS 2010/2012/2015/2017/2022...)
    - 윈도우에서 32 bit 프로그램으로 작성
    - 다른 운영체제 (MacOS, Linux)에서의 개발 환경 설정
- 주의점:
  - 프로그래밍 과제는 실습실 환경 (+ MSVC 2019 x64 Windows c++ 컴파일러)에서 채점할 예정. 다른 곳에서 작성한 파일들 (xxx.cpp, xxx.glsl)이 실습실 환경에서 제대로 동작하는지 제출 전에 확인할 것

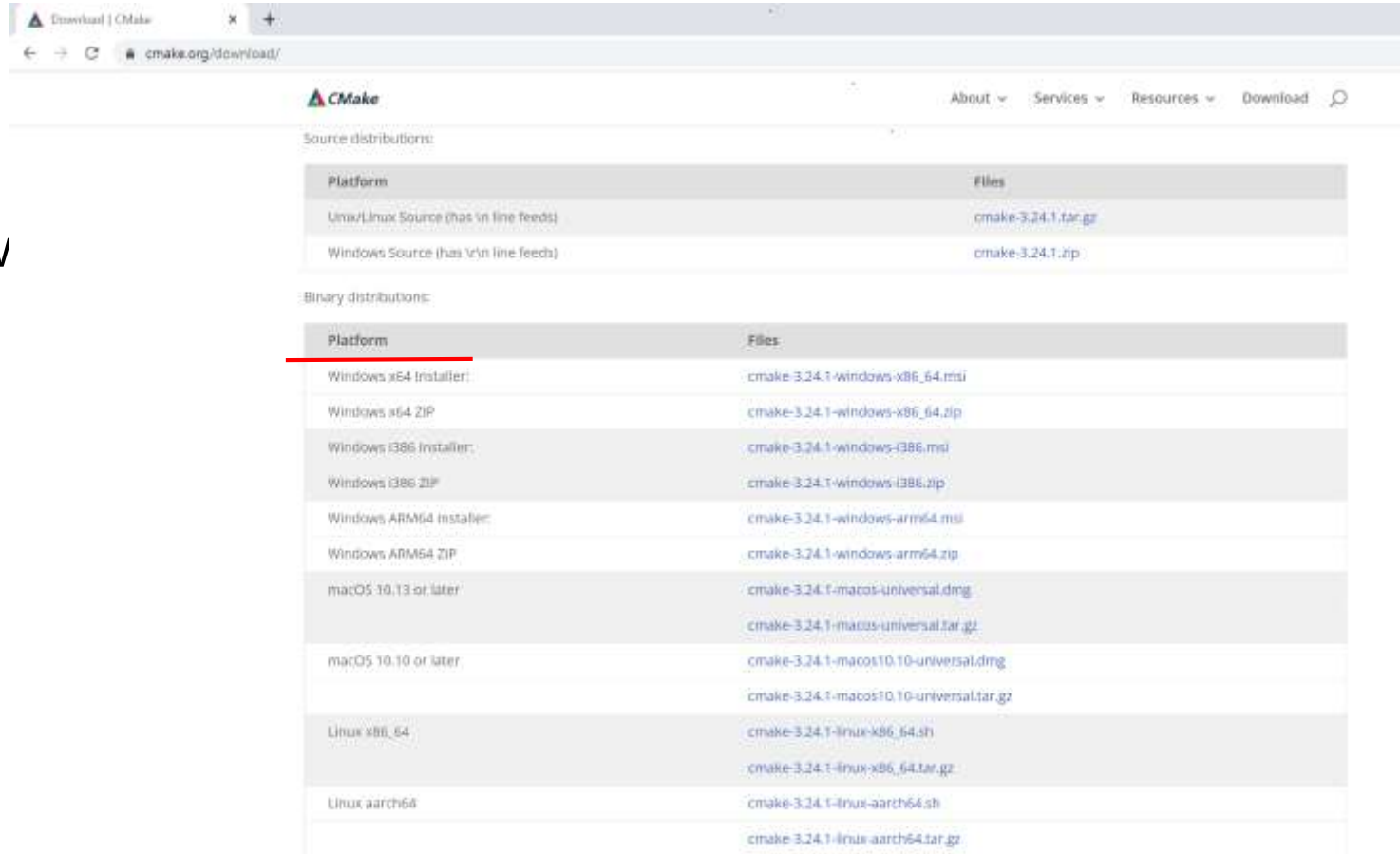
# 준비물

- C++ 컴파일러
  - Visual Studio C++를 설치했으면 이미 포함되어 있음
  - macOS/Linux의 경우 gcc/g++ 컴파일러
- CMake (<http://cmake.org>)
  - Open source, cross platform tool to compile a software packing with the user-specified compiler
  - CMake (command line) or CMake-GUI (graphical user interface)
- Packages to build
  - freeGLUT: freeGLUT-3.2.2.tar.gz (<http://freeglut.sourceforge.net/index.php#download>)
  - GLEW: glew-2.1.0.zip or glew-2.1.0.tgz (<http://glew.sourceforge.net/>)

# Step 0. CMake 설치

- Binaries already exists, no need to build CMake (<https://cmake.org/download/>)

- 원하는 운영체제와 platform 선택하여 install: 64bit freeglut,glew 라이브러리들을 빌드하기 위해서는 x64 Cmake를 설치해야함



The screenshot shows the CMake website's download page. It features a navigation bar with links for 'About', 'Services', 'Resources', and 'Download'. The main content is divided into two sections: 'Source distributions' and 'Binary distributions'. Each section contains a table with 'Platform' and 'Files' columns. In the 'Binary distributions' table, the 'Windows x64 Installer' row is highlighted with a red underline.

Platform	Files
Linux/Linux Source (has '\n' line feeds)	<a href="#">cmake-3.24.1.tar.gz</a>
Windows Source (has '\r\n' line feeds)	<a href="#">cmake-3.24.1.zip</a>

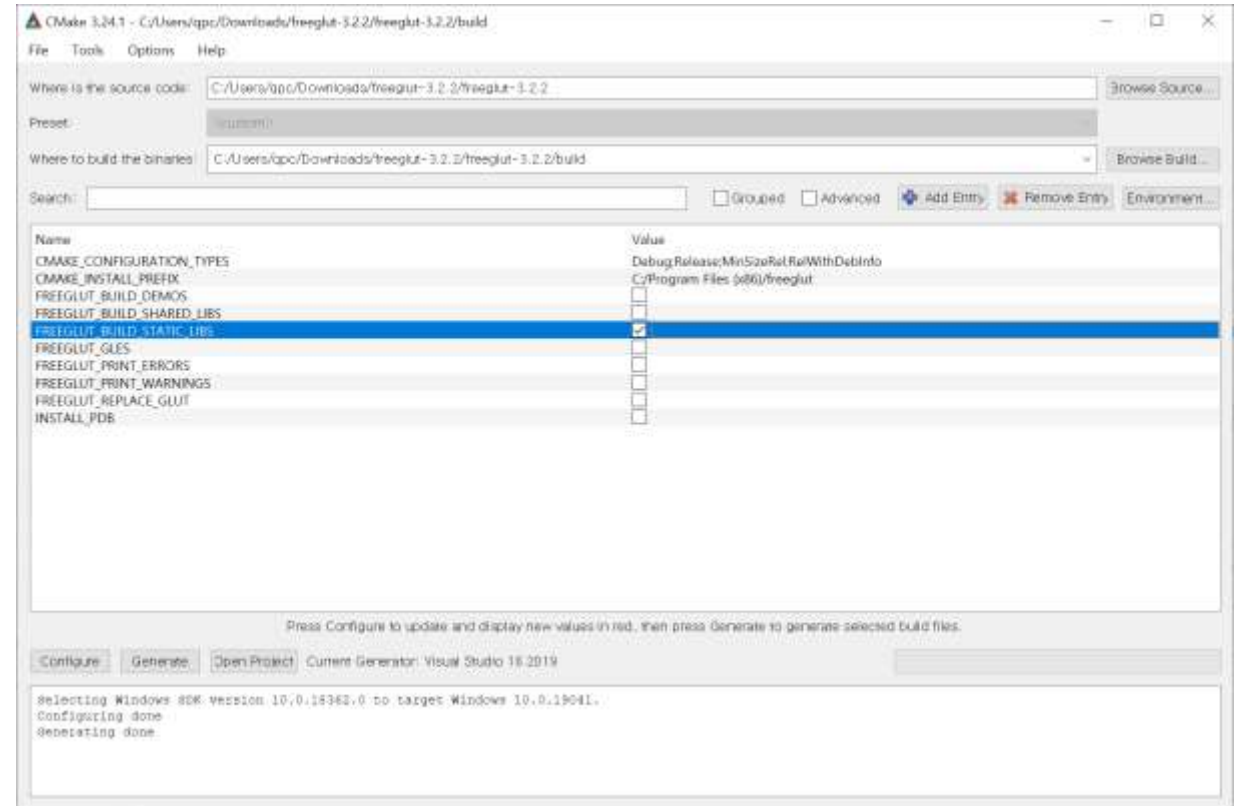
Platform	Files
Windows x64 Installer:	<a href="#">cmake-3.24.1-windows-x86_64.msi</a>
Windows x64 ZIP	<a href="#">cmake-3.24.1-windows-x86_64.zip</a>
Windows i386 Installer:	<a href="#">cmake-3.24.1-windows-i386.msi</a>
Windows i386 ZIP	<a href="#">cmake-3.24.1-windows-i386.zip</a>
Windows ARM64 Installer:	<a href="#">cmake-3.24.1-windows-arm64.msi</a>
Windows ARM64 ZIP	<a href="#">cmake-3.24.1-windows-arm64.zip</a>
macOS 10.13 or later	<a href="#">cmake-3.24.1-macos-universal.dmg</a> <a href="#">cmake-3.24.1-macos-universal.tar.gz</a>
macOS 10.10 or later	<a href="#">cmake-3.24.1-macos10.10-universal.dmg</a> <a href="#">cmake-3.24.1-macos10.10-universal.tar.gz</a>
Linux x86_64	<a href="#">cmake-3.24.1-linux-x86_64.sh</a> <a href="#">cmake-3.24.1-linux-x86_64.tar.gz</a>
Linux aarch64	<a href="#">cmake-3.24.1-linux-aarch64.sh</a> <a href="#">cmake-3.24.1-linux-aarch64.tar.gz</a>

# Step 1. freeGLUT 설치하기

- 다음의 방법 중 선택하여 freeGLUT 설치

**(방법1)** Cmake-GUI.exe이용 (windows)

1. "Where is the source code" 박스에 freeglut-3.2.2의 root directory(/src와 README.cmake가 포함된 directory)의 위치를 입력
2. "Where to build the binaries" 박스에 visual studio 솔루션 파일을 저장할 위치 지정 (예: /build)
3. 하단의 "Configure" 버튼 누르기
4. 새창에서 C++ 컴파일러 버전 선택 (예: Visual Studio 16 2019)
5. OK 누르고 configuration 끝날 때까지 기다림
6. Configuration 결과의 체크박스를 통해 생성할 라이브러리를 선택할 수 있음 (우측 그림 참조)
7. 종료 후 "Generate" 버튼 누르기



# Step 1. freeGLUT 빌드하기

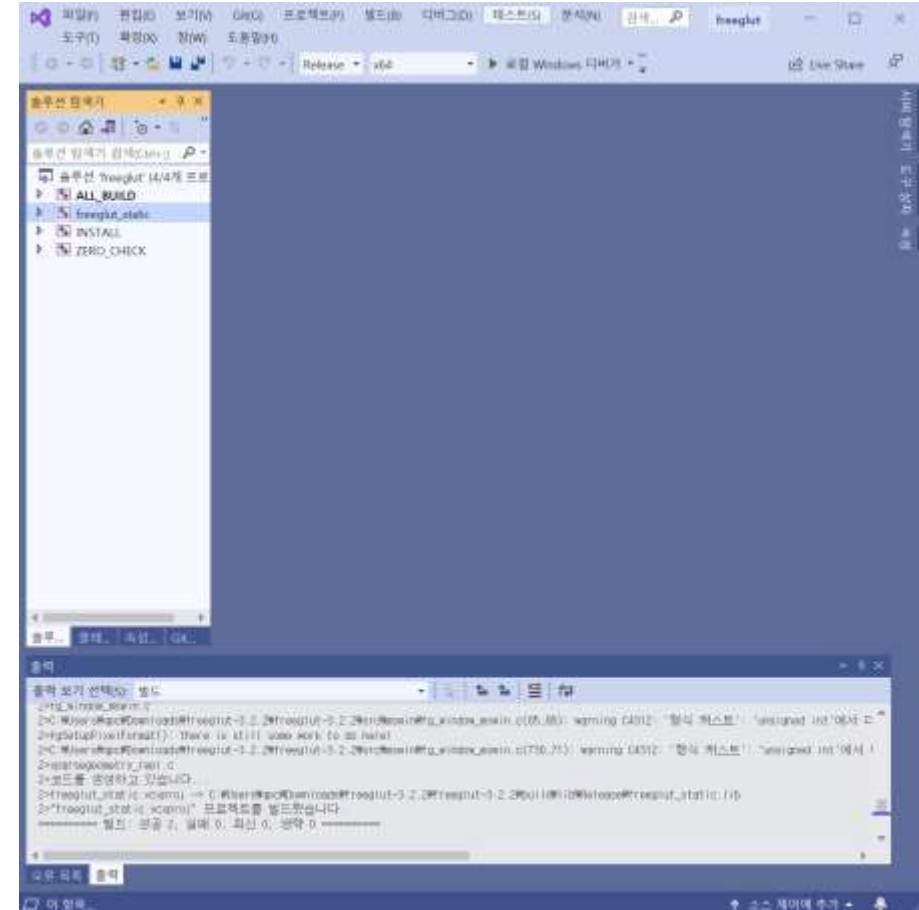
8. /build/freeglut.sln 열기

9. freeglut.sln에서 freeglut\_static 프로젝트 build하기 (release, debug)

10.

/build/lib/Release/freeglut\_static.lib  
(/build/lib/Debug/freeglut\_staticd.lib)  
b) 생성 확인 => c:\w\cg\w\lib 로 이동

11. /freeglut/include/GL/\*.h =>  
c:\w\cg\w\include/GL/\*.h로 이동



## (Step 2. GLEW 빌드하기)

- freeGLUT과 같은 방식으로 Cmake-GUI를 통해 빌드, 설치할 수 있음. 이때,
  - Source의 위치: /glew-2.1.0/build/cmake
  - Static library의 경우 64bit 버전도 libglew32.lib (libglew32d.lib for debug)의 이름으로 생성됨 (by default)

# macOS/Linux 에서 freeGLUT 설치하기

**(방법 2)** macOS에서 homebrew를 이용, freeGLUT 빌드/설치하기

- Homebrew는 macOS/Linux에서 소프트웨어 패키지의 설치를 간편화시키는 오픈 소스 프로그램
- freeGLUT과 glew의 경우, Homebrew를 통해서 설치할 수 있음
  - <https://formulae.brew.sh/formula/freeglut>
  - <https://formulae.brew.sh/formula/glew>

# macOS/Linux에서 OpenGL 프로그래밍하기

- 주의점

1. Homebrew를 통해 설치된 freeGLUT와 glew는 별도의 설정이 없으면 macOS/Linux의 native c++ compiler를 사용함

⇒ Xcode등 다른 IDE 사용

2. In vgl.h: #pragma macro is not supported in macOS/Linux

⇒ You have to link freeGLUT and glew libraries to your application!

3. Visual Studio for mac: not guaranteed to work with CMake