

OpenGL Setup for Visual Studio

Because this setup is complex, please follow carefully. These instructions were developed for Visual Studio 2019; new releases of Visual Studio can bring minor changes to these steps.

The setup consists of copying files as well as specific Visual Studio operations; these operations are numbered 1-13. Order is important to the following steps.

For the Mac

Apple no longer supports modern OpenGL. To work with OpenGL on a Mac, programmers should run Boot Camp Assistant (free software) to install MS Windows; this will allow access to OpenGL drivers. For good performance, it may be advisable to allocate more than the minimum storage required for Boot Camp.

If your Mac is of M1 architecture (introduced in 2020) you will be unable to run Boot Camp and unable to use the OpenGL API. Most OpenGL exercises should translate to Apple's graphics API, Metal.

For Windows and Linux Machines and Macs running Bootcamp

Suppose your name is Robin, and your graphics work is in the directory C:/Users/Robin/Graphics.

Create Project

1. Open Visual Studio.

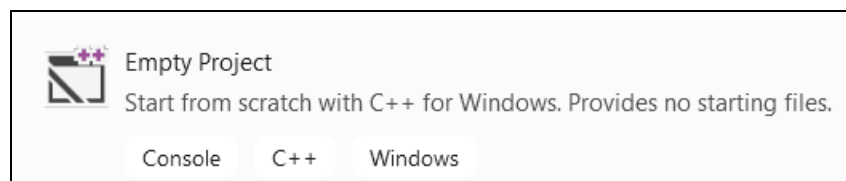
2. Ensure that "solution platform" is set to **x86**.



3. Under File tab, **New → Project**.

4. Select **Empty Project**.

There should be "**C++**" indicated next to the selection. If no C++ option appears, you may need to download a Visual Studio extension. Go to Tools → Get Tools and Features -> select the Windows 10 (or 11) Software Development Kit and download.



5. Browse to C:/Users/Robin/Code/Graphics (using the "..." button).

6. Name: **Apps** (or whatever you prefer); this will create a subdirectory Apps under Graphics.

7. With older Visual Studio, uncheck "Create directory for solution" and uncheck "Add to source control"; with newer versions, check "Place solution and project in same directory".

8. Older: click OK; newer: Create

9. (optional) Create desktop shortcut to C:/Users/Robin/Graphics/Apps/Apps.sln.

Copy Source Files

Copy Downloads.zip's Include directory to your Graphics directory, making C:/Users/Robin/Graphics/**Include**.

Under Graphics, create subdirectory **Lib** and, from Downloads.zip/Lib, copy GLXtras.cpp and glad.c to C:/Users/Robin/Graphics/Lib.

Creating the project (step 3-8, above) should have created C:/Users/Robin/Graphics/**Apps**. To it, copy 2-VersionGL.cpp and 2-ClearScreen.cpp from Downloads.zip/Apps.

You should now have:

C:/Users/Robin/Code/Graphics
 Apps (VS files Apps.sln, Apps.vcxproj, etc.), 2-VersionGL.cpp, and 2-ClearScreen.cpp)
 Include (.h files)
 Lib (GLXtras.cpp and glad.c)

Copy DLL Files

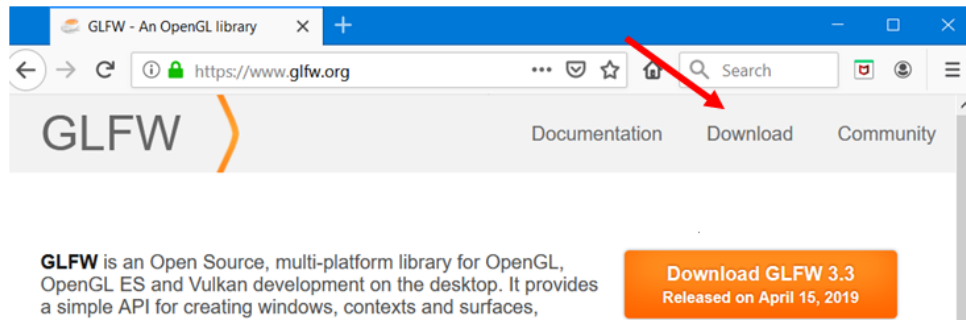
Determine the destination directory for DLL (dynamically-loaded library) files for your computer. For a 32-bit machine, it is **C:/Window/System32**; for a 64-bit machine it is **C:/Windows/SysWOW64**.

Bin-MSVC-x86 contains three dll files (gdi32, glu32, and opengl32) that are needed in your DLL directory. They are probably already there; copy those that don't exist, but *do not overwrite* those that do.

Copying to the dll directory requires administrator permission. You probably are the administrator of your computer, so simply click 'continue' when copying (if this fails, you can become administrator: ControlPanel → UserAccounts → Change Account Type).

Copy GLFW files

GLFW library files are obtained from www.glfw.org. Click on the Download button at the upper right.



This should lead to:

Windows pre-compiled binaries

These packages contain the GLFW header files, [documentation](#) and release mode static libraries, DLLs and import libraries for Visual C++ 2010-2019, MinGW-w64 and plain MinGW.

Binaries for Visual C++ 2010 and plain MinGW are only available in the 32-bit package.

64-bit Windows binaries

32-bit Windows binaries

Select the 32-bit version (it'll work on a 64-bit machine) because the 64-bit version is known to be problematic.

After download, open the .zip file and go to the appropriate sub-directory (e.g., lib-vc2019 for Visual Studio 2019).

Copy **glfw3.dll** to your DLL directory (C:/Windows/SysWOW64 or C:/Windows/System32).

Copy **glfw3.lib** and **glfw3dll.lib** to a common lib folder. This folder depends on your version of Visual Studio, but should look something like:

C:\ProgramFiles(x86)\MicrosoftVisualStudio\201?**Community**\VC\Tools\MSVC\???.???.???\lib\x86/
or, possibly, C:\ProgramFiles (i.e., without the x86).

If you can't find the above directory, you can use **your own Lib/ directory** (this is easier but a "common" directory is better).

Check Glad Version

Glad is an alternative to the now defunct GLEW (GL Extension Wrangler), allowing OpenGL to utilize whichever version is supported by your computer's GPU.

If your laptop is less than five years old, it likely supports OpenGL version 4.5. (To double-check, right-click on desktop, select NVidia control panel (if your GPU has an NVidia chip - if not, search for an equivalent option) and then (lower-left) click on System Information; the model number for the chip should be listed. Search the web for technical specifications, which should list the OpenGL version supported.)

The files **Include/glad.h** and **Lib/glad.c** support OpenGL version 4.5. If your GPU is less than 4.5, almost certainly glad.3.2.h and glad.3.2.cpp will work (copy them from Downloads.zip and rename them to glad.h and glad.c).

Set VS Source Files

10. View → Solution Explorer, right-click “Source Files” then Add → Existing Item, browse to and add **Apps/2-VersionGL.cpp**. Also add **Lib/GLXtras.cpp** and **Lib/glad.c**.

Project Settings

These settings must be manually made for each “solution configuration”, *i.e.*, Debug and Release modes. For this course there should be no advantage in Release mode; Debug mode is recommended.

11. In Visual Studio, select **Project** and then **Properties**. At left, under Configuration Properties (click to expand if necessary), find **Linker**.

Click ‘>’ to left of Linker, select “**Input**”

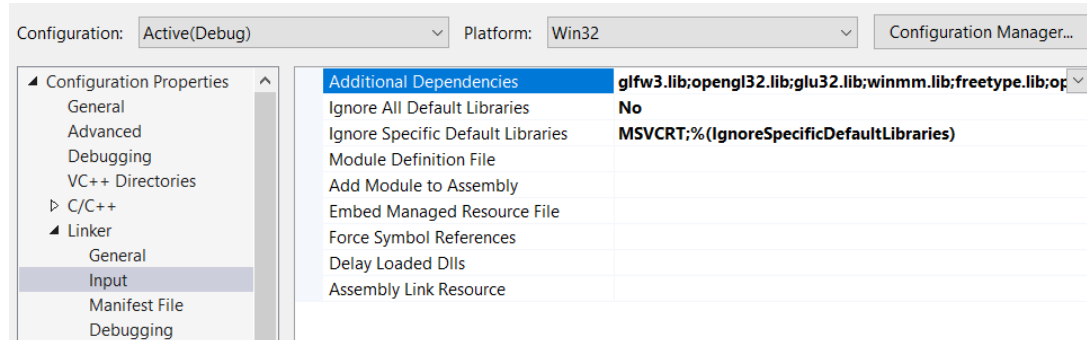
Add to “Additional Dependencies” - click on the window and then on the down arrow at the extreme right, select “<Edit>” and copy/paste:

glfw3.lib; opengl32.lib; glu32.lib

Be sure the “Inherit from parent or project defaults” is checked.

To suppress unnecessary linker warning, add to “Ignore Specific Default Libraries”

MSVCRT



12. To suppress unnecessary compiler warnings

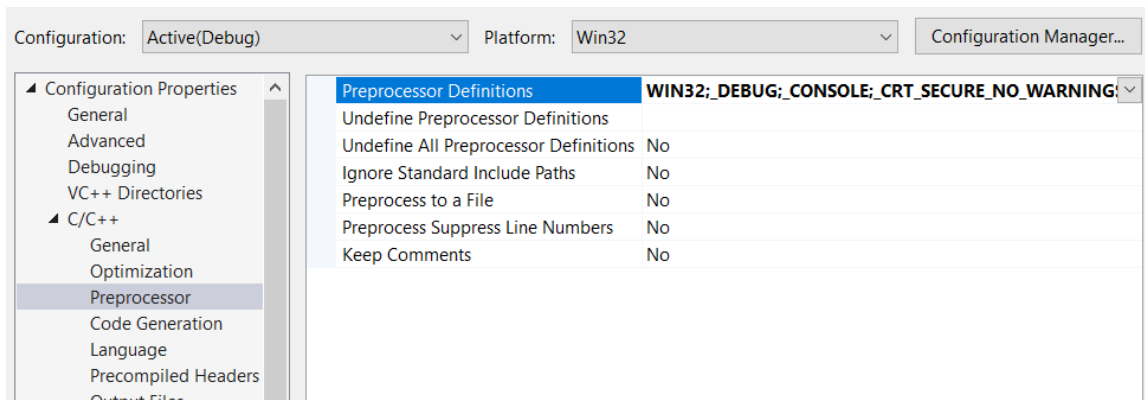
Click ‘>’ to left of **C/C++**, select “Preprocessor” (note: the C/C++ option does not appear until at least one source file has been added to the project).

click on the window and then on the down arrow at the extreme right of Preprocessor Definitions, select “<Edit>” and add:

_CRT_SECURE_NO_WARNINGS

CRT_SECURE_NO_DEPRECATED

(Spurious Microsoft compilation warnings might still be produced.)



13. Establish default directories for source (.h and .cpp) files

Click “VC++ Directories” under Configuration Properties in the same window as previous step.

Add your Include directory to “**Include Directories**”

You can add the directory by browsing to it: click on the down arrow at the extreme right, select <Edit>, click on the file-folder button with the ‘+’ symbol, then click on ‘...’

Add your Lib directory to “**Library Directories**”.

Test

Build and run 2-VersionGL.cpp.

If it does not run, under “VC++ Directories” click the arrow on the right of “Library Directories” then “Edit”. Under “Evaluated Value”, the first line is the directory to be used in previous step. Put library files (like GLXtras.cpp) in that directory and try again.

Now test 2-ClearScreen.cpp. Easiest way to do this is, under the Solution Explorer, right-click 2-VersionGL.cpp and select ‘Exclude from Project’. Then add 2-ClearScreen.cpp to the project and rebuild.