


## Using Modern OpenGL

اعتماداً على توصيف OpenGL القياسي تم إنشاء مكتبات جديدة من أجل التعامل مع برمجة وحدة معالجة كرت الفيديو GPU لتسهيل العمل، سندر من هذه المكتبات glew، كما يمكن الإطلاع على المكتبات الأخيرة ك Glad

### إعداد Glew


يمكن تنزيل الملفات المطلوبة من الموقع <http://glew.sourceforge.net>



The screenshot shows the GLEW website. The header includes the URL 'glew.sourceforge.net'. The main content area is titled 'The OpenGL Extension Wrangler Library'. It describes GLEW as a cross-platform open-source C/C++ extension loading library. The 'Downloads' section indicates that GLEW is distributed as source and precompiled binaries, with the latest release being 2.1.0. A table lists download options: Source (ZIP, TGZ) and Binaries (Windows 32-bit and 64-bit). The 'Supported Extensions' section lists OpenGL 4.6, WGL, and GLX extensions. The 'News' section mentions updates for GLEW 2.1.0 and 2.0.0.

كما يمكن الوصول إلى تعليمات الاستخدام من خلال الرابط Usage

Latest Release: 2.1.0




[Download](#)  
[Usage](#)  
[Building](#)  
[Installation](#)  
[Source Generation](#)  
[Change Log](#)

[GitHub](#)  
[Issues](#)  
[Pull Requests](#)  
[Authors](#)  
[Licensing](#)

[SourceForge Page](#)

Last Update: 07-31-17



## The OpenGL Extension Wrangler Library

### Initializing GLEW

First you need to create a valid OpenGL rendering context and call `glewInit()` to initialize the extension entry points. If `glewInit()` returns `GLEW_OK`, the initialization succeeded and you can use the available extensions as well as core OpenGL functionality. For example:

```
#include <GL/glew.h>
#include <GL/glut.h>
...
glutInit(&argc, argv);
glutCreateWindow("GLEW Test");
GLenum err = glewInit();
if (GLEW_OK != err)
{
    /* Problem: glewInit failed, something is seriously wrong. */
    fprintf(stderr, "Error: %s\n", glewGetErrorString(err));
    ...
}
fprintf(stdout, "Status: Using GLEW %s\n", glewGetString(GLEW_VERSION));
```

### Checking for Extensions

Starting from GLEW 1.1.0, you can find out if a particular extension is available on your platform by querying globally defined variables of the form `GLEW_{extension_name}`:

```
if (GLEW_ARB_vertex_program)
{
    /* It is safe to use the ARB_vertex_program extension here. */
    glGenProgramsARB(...);
}
```

**In GLEW 1.0.x, a global structure was used for this task. To ensure binary compatibility between releases, the struct was replaced with a set of variables.**

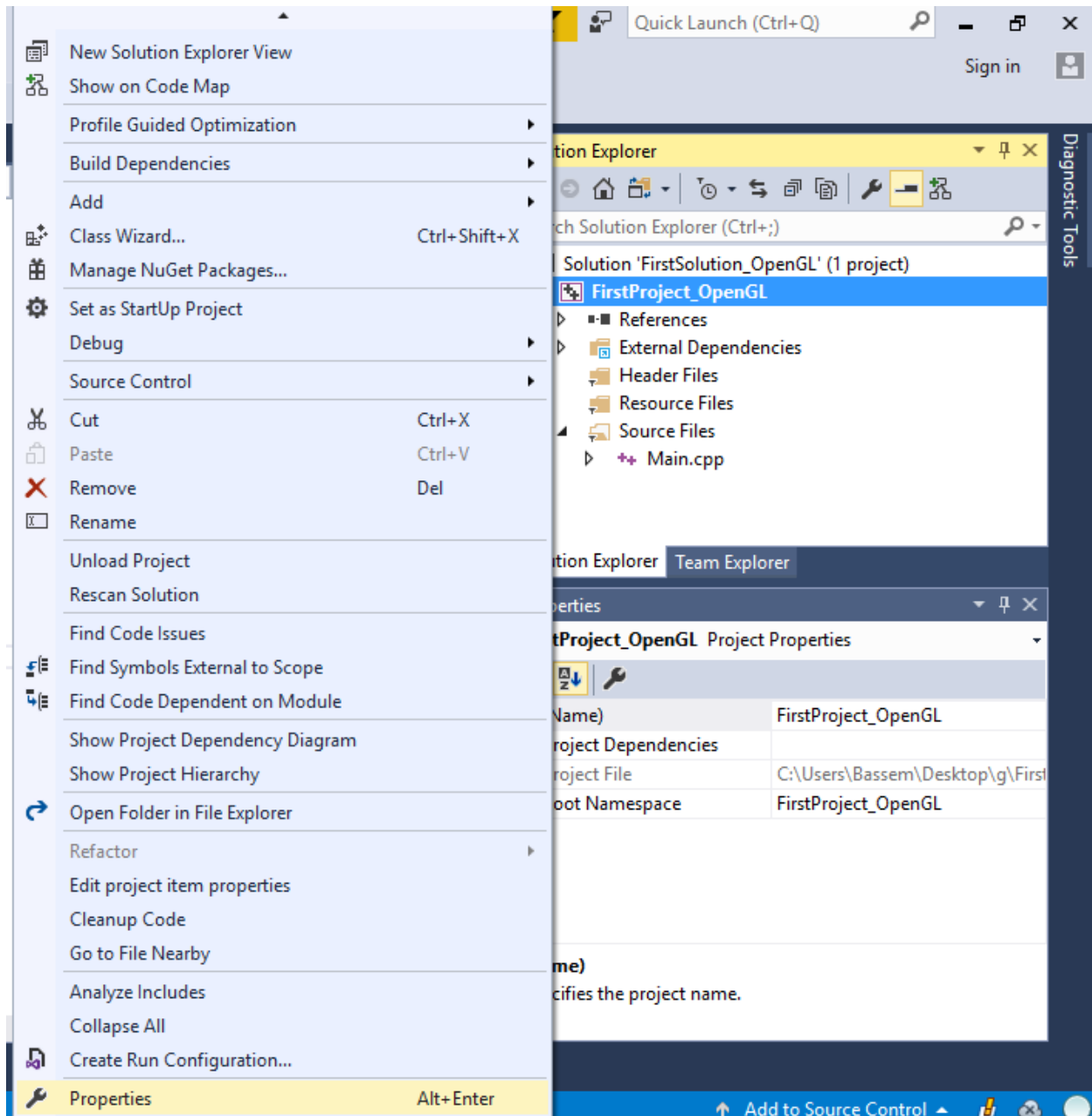
You can also check for core OpenGL functionality. For example, to see if OpenGL 1.3 is supported, do the following:

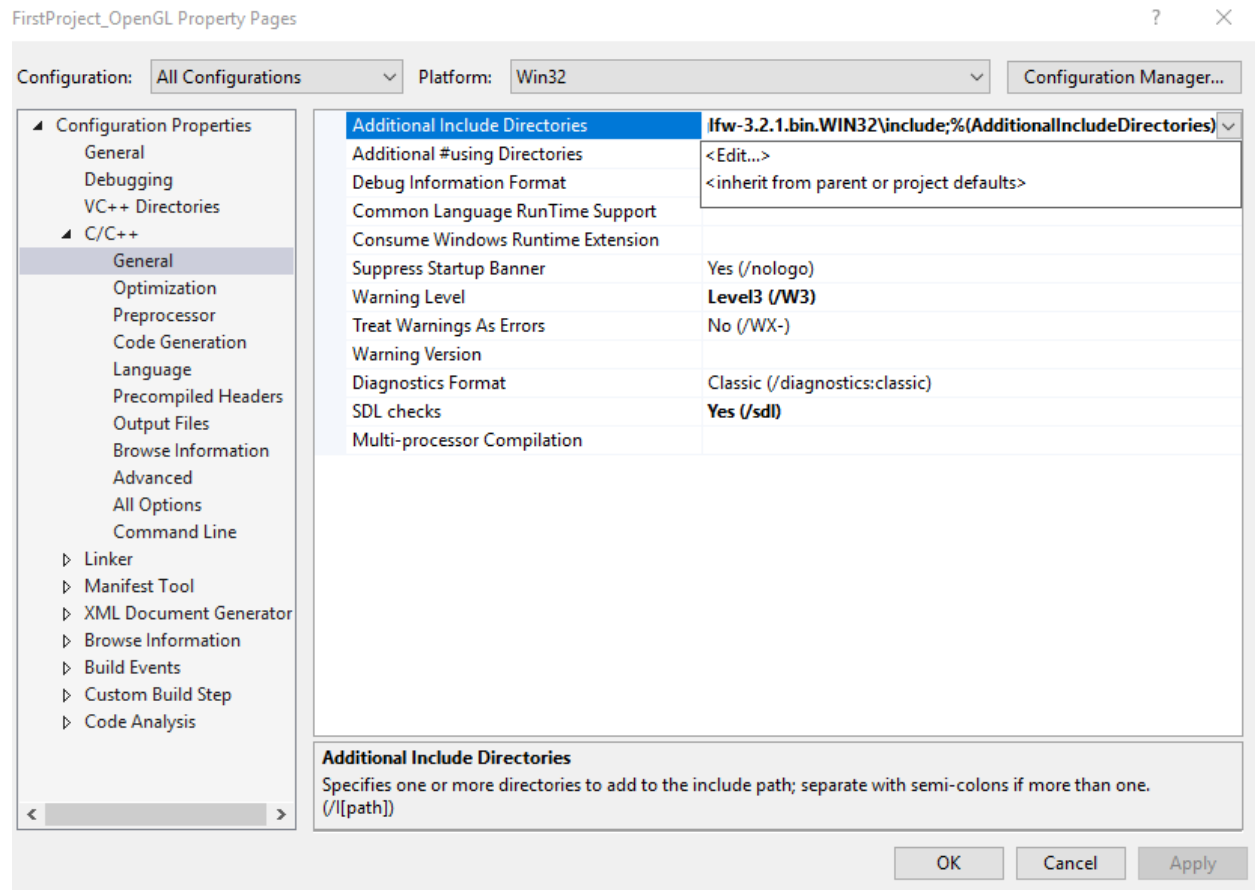
بعد تنزيل الملفات المطلوبة وبعد فك ضغط الملفات يمكن نقل المجلد بكامله إلى المجلد lib الذي أنشأناه في المحاضرة السابقة

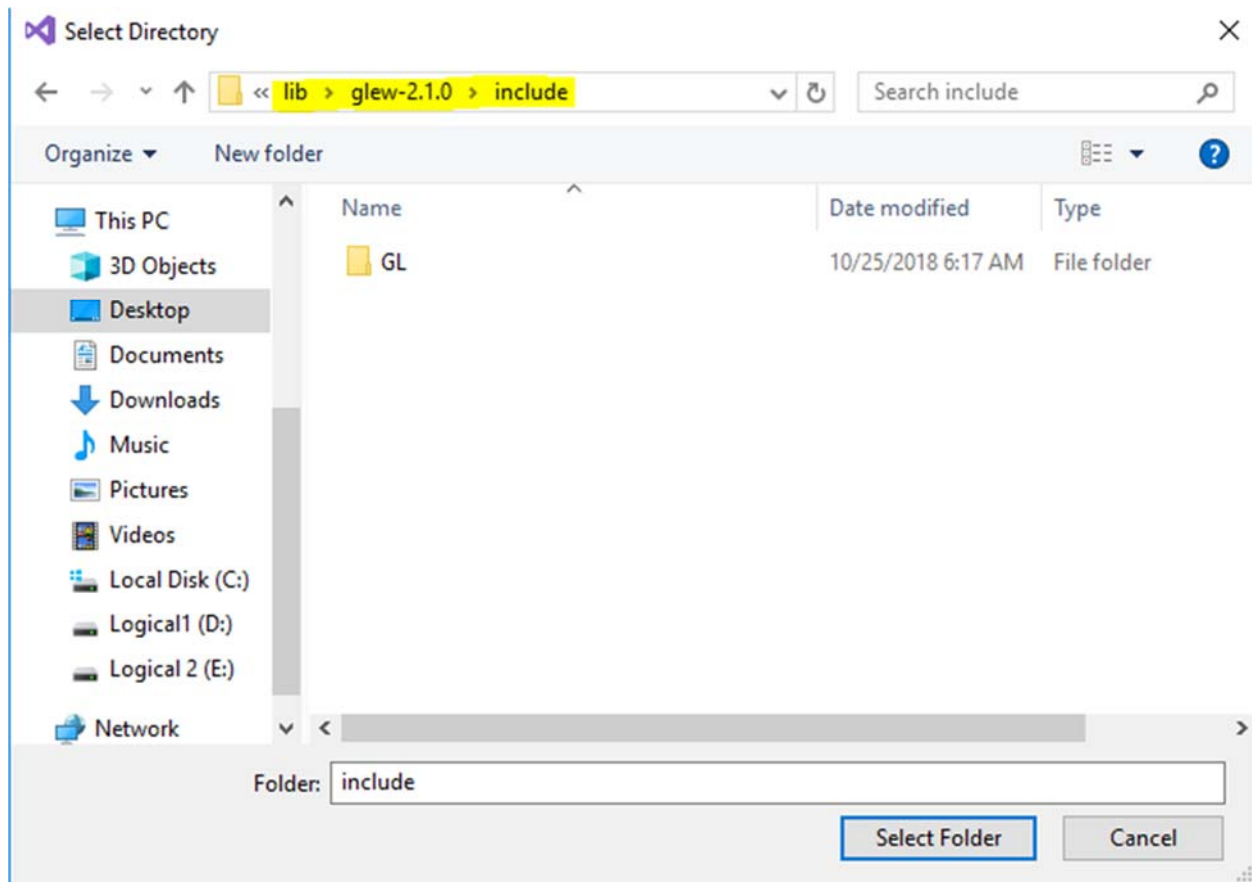
The screenshot shows a Windows File Explorer window. The address bar displays the path: `↑ > g > FirstSolution_OpenGL > FirstProject_OpenGL > lib`. The left sidebar shows the 'access' tab selected, with 'top', 'nloads', and 'uments' visible below it. The main pane displays a list of files and folders:

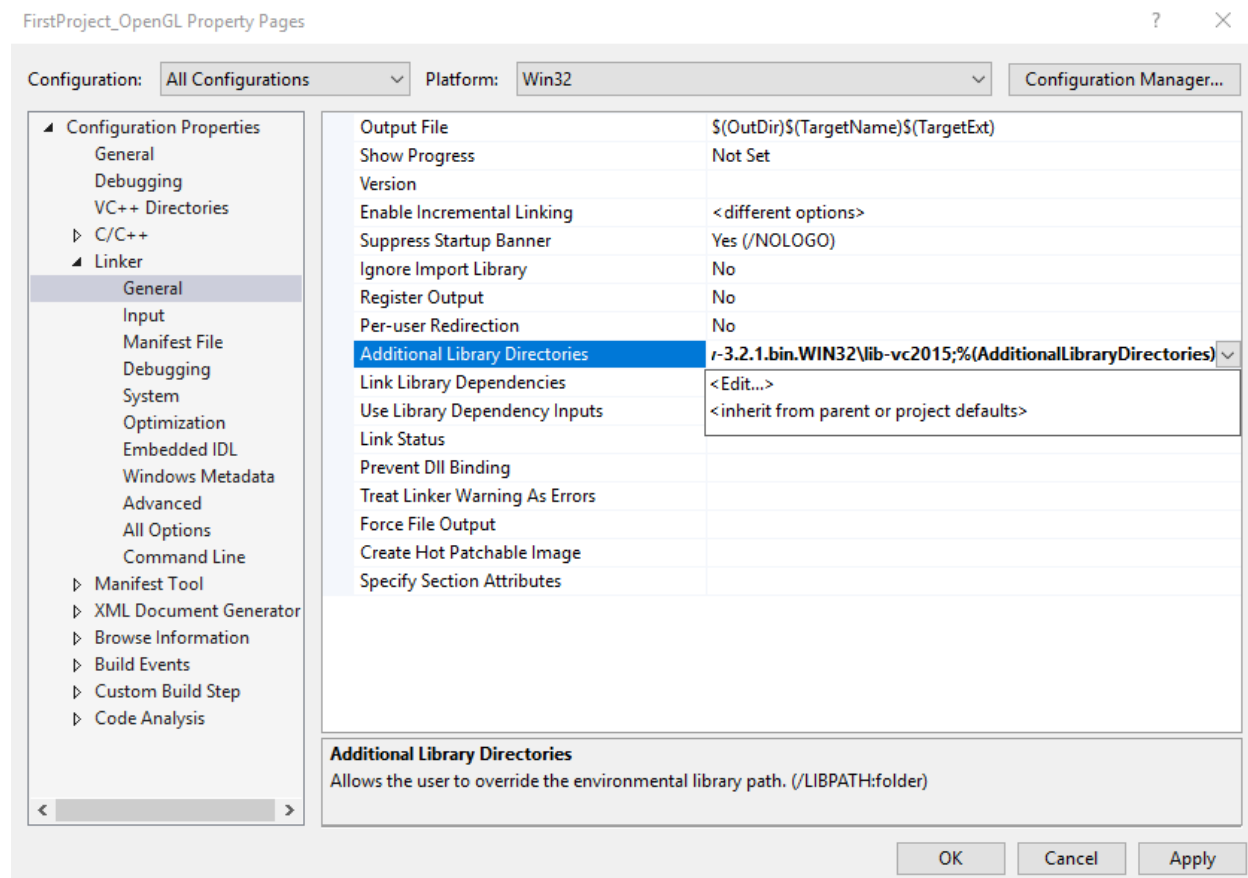
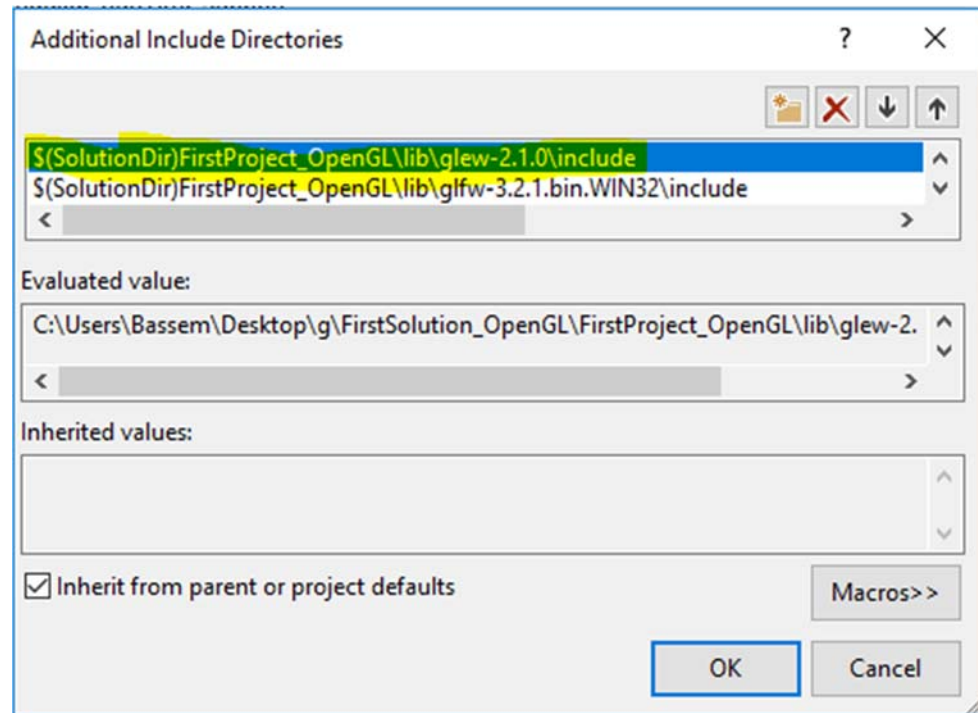
Name	Date modified	Type	Size
glew-2.1.0	10/25/2018 6:17 AM	File folder	
glfw-3.2.1.bin.WIN32	8/18/2016 2:38 PM	File folder	

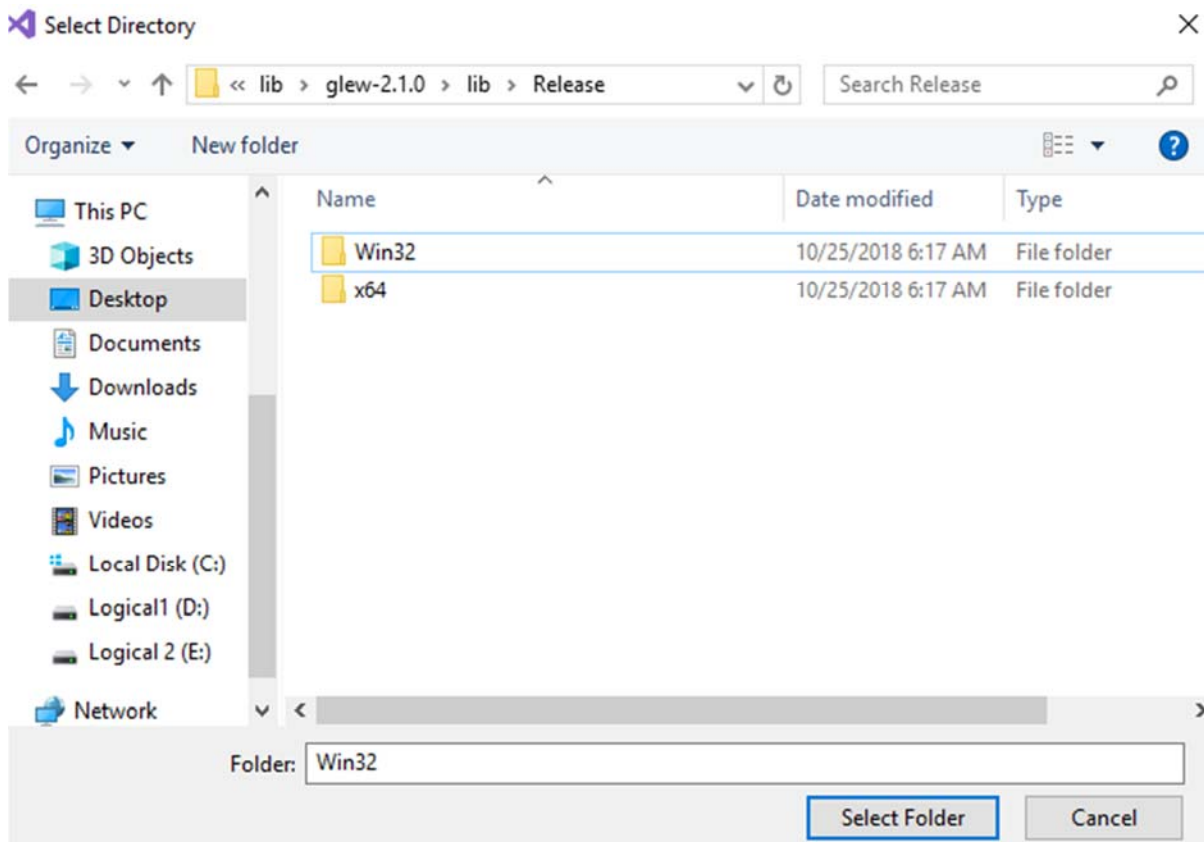
عملية إعداد GLEW تشبه GLFW ولكن هناك مرحلة إضافية سيتم شرحها عند الوصول إليها (إضافة ملفات dll).

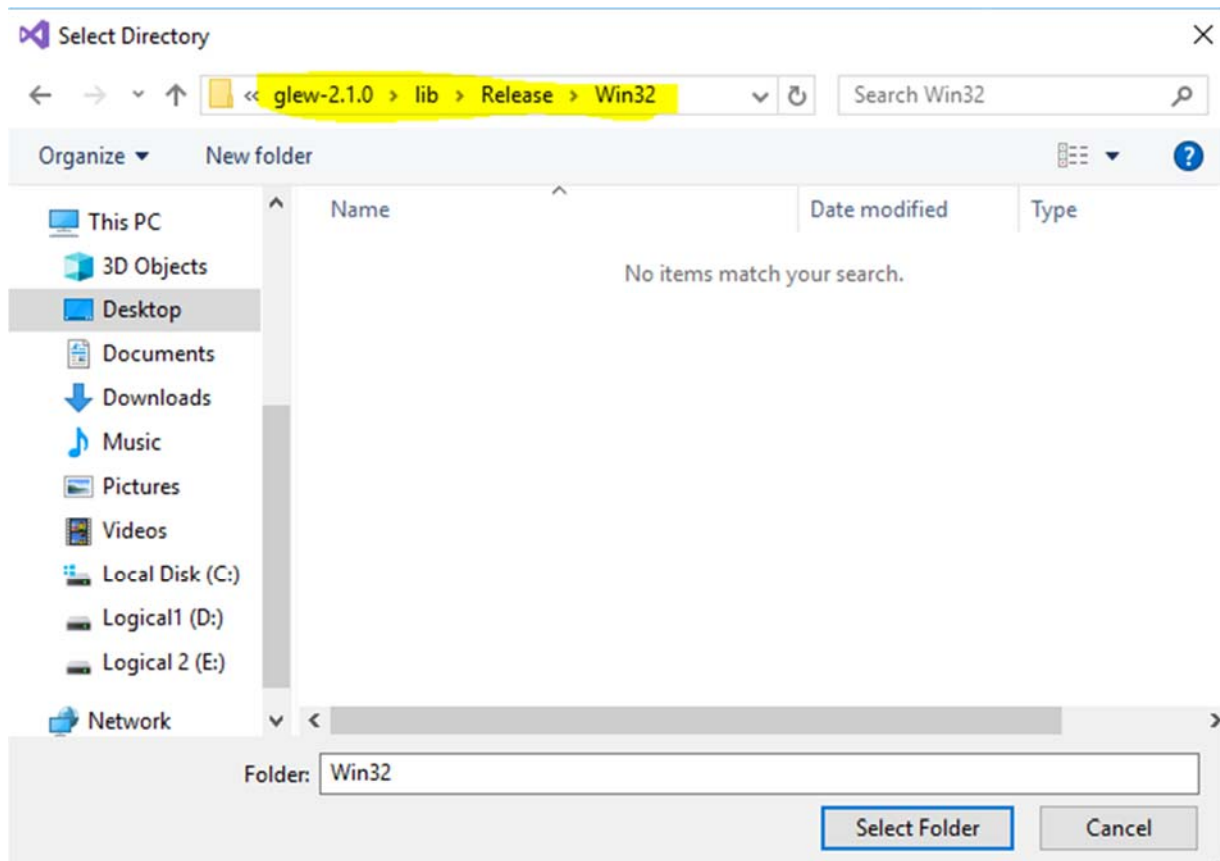




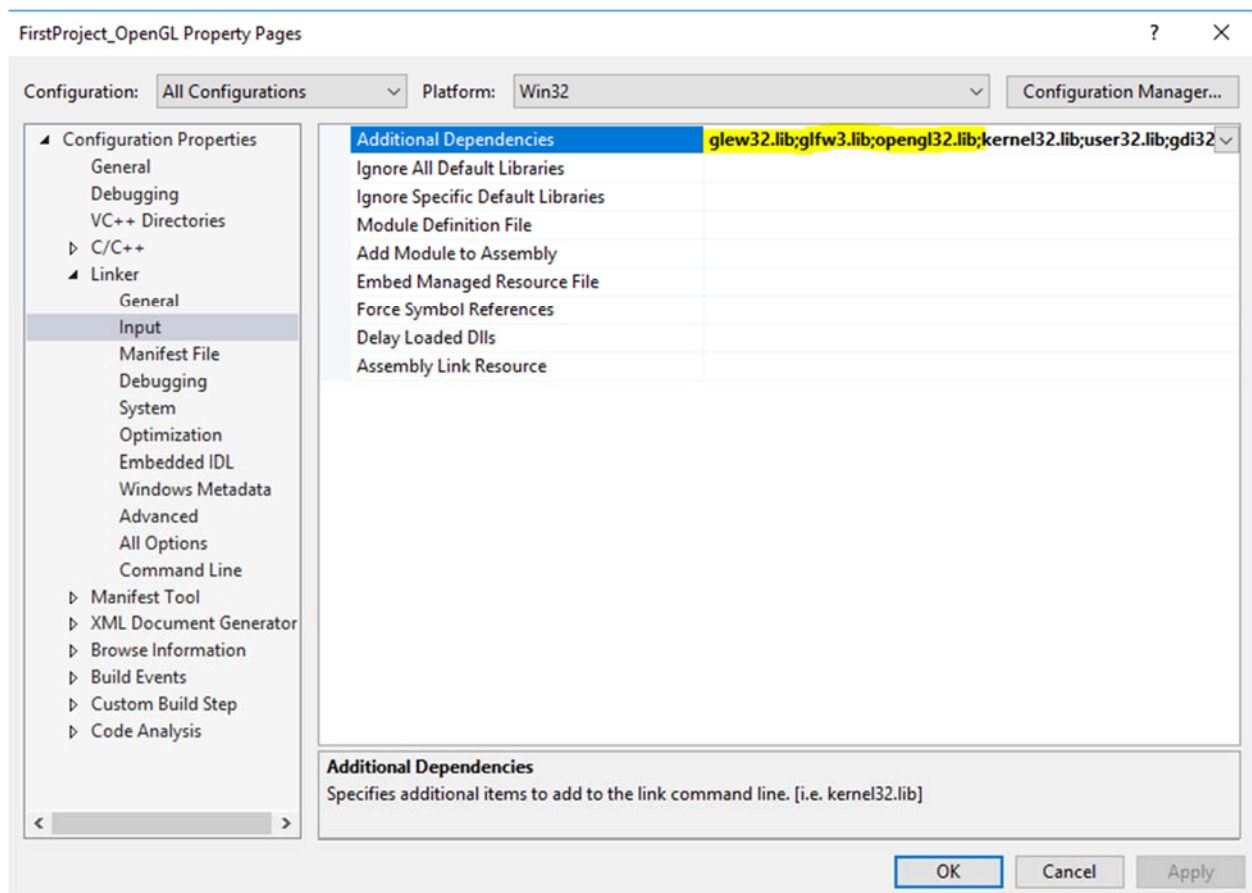
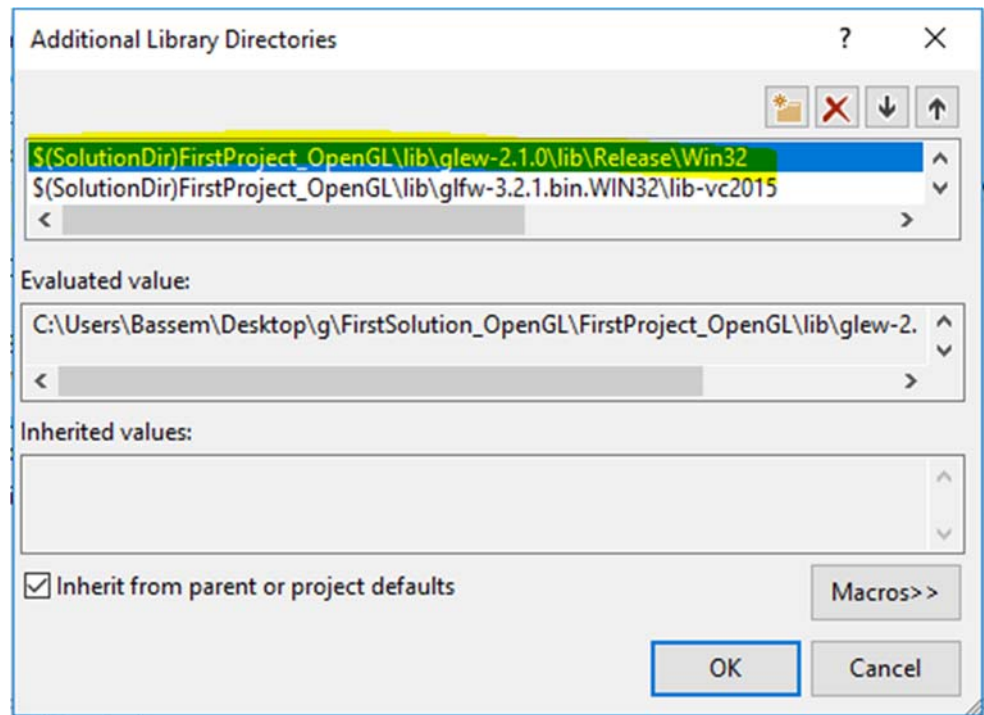




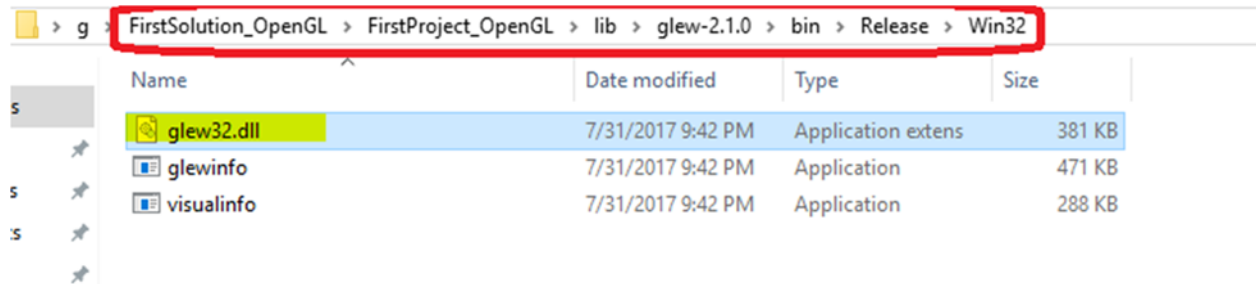




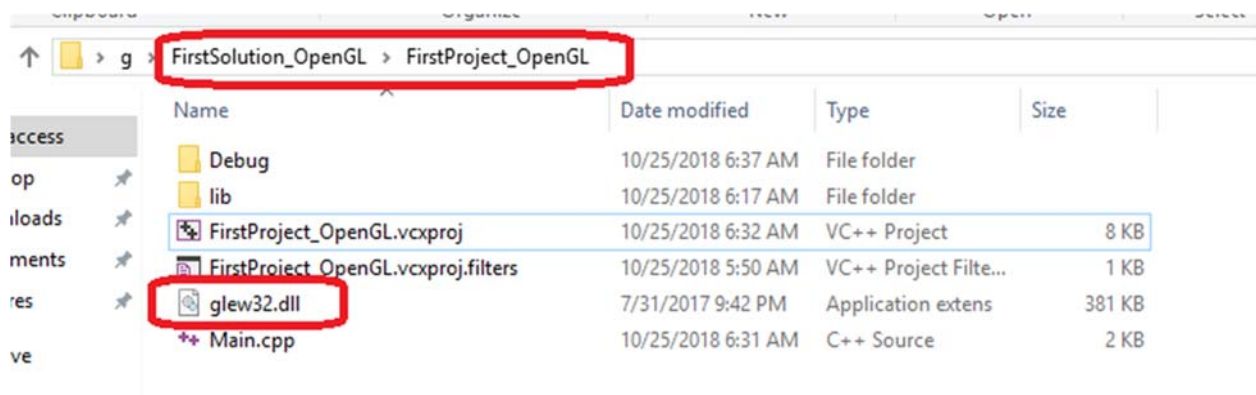




بعد الانتهاء من الخطوات السابقة يمكن الوصول إلى المسار (حسب المعالج 64 أو 32):



ثم نسخ الملف glew32.dll إلى مجلد المشروع



بعد ذلك تضمين ملف الترويسة للمكتبة في الرماز المصدري (يجب أن يكون قبل أي عملية include لأي مكتبة OpenGL أخرى، في حالتنا قبل glfw).

```
#include <GL/glew.h>
#include <GLFW/glfw3.h>

#include <iostream>
using namespace std;
```

```
    return -1;
}

/* Make the window's context current */
glfwMakeContextCurrent(window);

GLenum err = glewInit();
if (GLEW_OK != err)
{
    /* Problem: glewInit failed, something is seriously wrong. */
    cout << "glew error";
}

/* Loop until the user closes the window */
while (!glfwWindowShouldClose(window))
{
    /* Render here */
    glClear(GL_COLOR_BUFFER_BIT);
```

بالطبع يمكنك الوصول إلى مساعدة لتوابع المكتبة من خلال المسار

FirstSolution\_OpenGL\FirstProject\_OpenGL\lib\glew-2.1.0\doc

والبحث عن ملف Index