



NGL (NCCA Graphics Library) Windows Installation Guide

by Jonathan Flynn - jonflynn designer@gmail.com

Setup for the NGL library for Windows can be a little fiddly but once you understand the process it is fairly easy to grasp. Since there is no detailed documentation on the setup of this, I've decided to write one. If you need any help then feel free to drop me an email at jonflynn designer@gmail.com and if you see any errors, I've left the document for people to be able to add comments so feel free to suggest any changes.

[Installing NGL](#)

[Install Microsoft Visual Studio 2013](#)

[Install Qt 5.4 \(5.4.1\)](#)

[Install Boost](#)

[Install GitHub and git clone NGL](#)

[Build NGL](#)

[Testing using SimpleNGL](#)

[Installing SDL](#)

[Download DirectX Software Development Kit](#)

[Building SDL](#)

[Testing using SDLNGL](#)

[Error Solutions](#)

[Common Errors with Example Projects](#)

[SDL Build Errors](#)

Installing NGL

Install Microsoft Visual Studio 2013

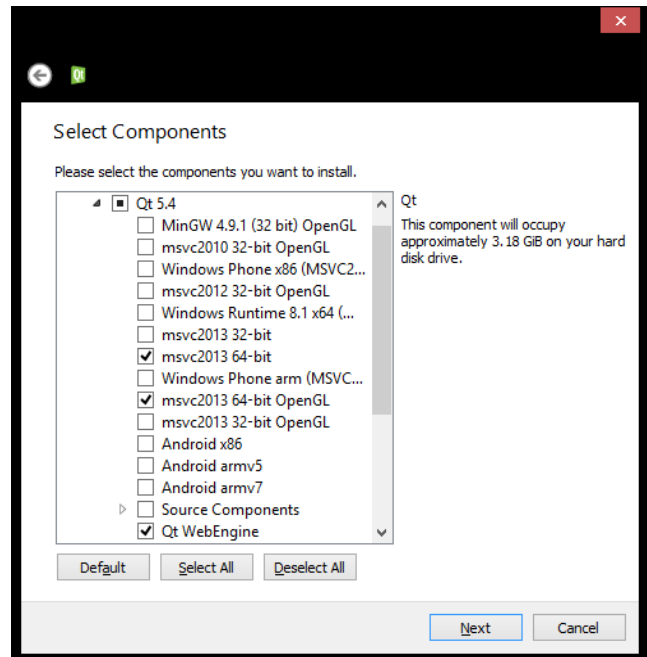
If you are a student, you can get Visual Studio and loads of other software free through Microsoft Dreamspark. You can find out more information [here](#), and all you need to do is sign up and link your student email to an account. If you're currently at the NCCA, then you can get the software free using the NCCA student account and password without needing to sign up [here](#). The specific version that I downloaded and installed was [Microsoft Visual Studio Professional 2013 with Update 2 32-Bit - DVD](#). Ask a computing lecturer for the account details to access the Dreamspark page. Complete the installation with the default setup and update the software to version 4 [here](#).

Install Qt 5.4 (5.4.1)

Download the Community build of Qt 5.4 [here](#). Install it in the default directory of `C:/Qt`. In the "Select Components" section, un-select "msvc2012 32-bit" and select "msvc2013 64-bit OpenGL". If that was confusing, just make sure that only the following are selected under the Qt 5.4 drop down option:

- msvc2013 64-bit
- msvc2013 64-bit OpenGL
- Qt WebEngine

Then continue the install operation as normal.



Install Boost

Download Boost 1.57.0 from [here](#). Extract the .zip file and rename the folder from `boost_1.57.0` to `boost` and move the folder to the location `C:/` so that you have the path of `C:/boost/bootstrap.bat` for the file `bootstrap.bat`.

To install boost, simply run the file `bootstrap.bat` either in the file explorer or using the Command Prompt followed by running `b2.exe` in the same folder which should have been created from `bootstrap.bat` if the setup was successful.

Install GitHub and git clone NGL

Download GitHub from [here](#). Go through the install process. Once complete you need to clone the NGL library. You can do this by either going through the interface or opening the Git Shell and running the command:

[git clone git@github.com:NCCA/NGL.git](https://github.com/NCCA/NGL.git)

If you do it through the interface you just need to login, click the plus symbol in the top left and clone under the NCCA/NGL on the left hand side. Either make sure you download the library into the location **C:** or move the entire library for it afterwards to **C:/NGL** so that you have the path **C:/NGL/NGL.pro** for the **NGL.pro** file

Build NGL

Open the NGL.pro file in QtCreator and configure the NGL project for **Desktop Qt 5.4.1 MSVC2013 OpenGL 64bit** and then disable **Shadow build** under **Projects**. Finally, build the project, from which you should only get warnings and no errors. After the build you should have the file **NGL.lib** in the folder **C:/NGL/lib/**.

Testing using SimpleNGL

Git clone the SimpleNGL project [git@github.com:NCCA/SimpleNGL.git](https://github.com/NCCA/SimpleNGL.git) and open the .pro file in QtCreator. Configure the project like before and disable shadow build. In the .pro file, you need to change the line 71 which is the following:

```
INCLUDEPATH += $$$(HOME)/NGL/include/
```

and change the above line to be:

```
!win32:INCLUDEPATH += $$$(HOME)/NGL/include/
```

After that, you then you need to add to the win32 section from line 73 to 83 by inserting a line below 75 with:

```
INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/
```

So it should all look like this:

```
!win32:INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/

win32: {
    PRE_TARGETDEPS += C:/NGL/lib/NGL.lib
    INCLUDEPATH += -I c:/boost
    INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/
    DEFINES += GL42
    DEFINES += WIN32
    DEFINES += _WIN32
    DEFINES += _USE_MATH_DEFINES
    LIBS += -LC:/NGL/lib/ -INGL
    DEFINES += NO_DLL
}
```

Save the file, run qmake again (Build>Run qmake) and then build and run the project. All done! You should only get warnings but no errors should occur.

If you get any errors, check the Errors section at the end of this document.

Installing SDL

The following steps are optional only for people doing projects using SDL.

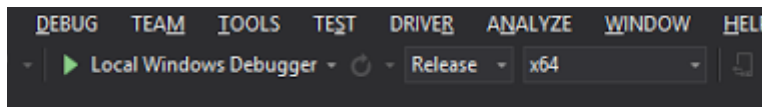
Download DirectX Software Development Kit

Before we can compile and build SDL, we need to download the DirectX Software Development kit. You can download this from [here](#). Just run through the installation and it should all be fine. You should be able to ignore the Error Code S1023 if it happens to you.

Building SDL

Download the SDL 2.0.3 source code from [here](#). Once you have downloaded it, extract the .zip and rename the folder from **SDL2-2.0.3** to **SDL2** and move it to the **C:/** drive so that the path for the file **README.txt** is **C:/SDL2/README.txt**.

Now, open the file **SDL_VS2013.sln** inside the **VisualC** folder. It will ask if you trust the solution, just untick "Ask me for every project in the solution" and click OK. Set the project to **Release** and **x64** in the top of the window in the two drop down boxes shown below.



You should just be able to **BUILD>Build solution** and get only warnings and no errors. Once complete, you need to navigate to the **C:/SDL2/VisualC/SDL/x64/Release** folder and find the files called **SDL2.lib** and

SDL2.dll. Move these files to a new folder in the root **SDL2** directory called **lib** so the files should be in a folder called **C:/SDL2/lib**. You will also need to move **SDL2main.lib** from the folder **VisualC/SDLmain/x64/Release** and place it in the same lib folder.

Testing using SDLNGL

Git clone the SDLNGL project [git@github.com:NCCA/SDINGL.git](https://github.com/NCCA/SDINGL) and open the .pro file in QtCreator. Configure the project like before and disable shadow build. You must edit the .pro file so that the project has the necessary includes and libs for SDL. To do this, you need to replace the existing commands used for unix and linux (lines 15-19) which is the following:

```
QMAKE_CXXFLAGS += $$system(sdl2-config -cflags)
message(output from sdl2-config -cflags added to CXXFLAGS= $$QMAKE_CXXFLAGS)

LIBS += $$system(sdl2-config -libs)
message(output from sdl2-config -libs added to LIB=$$LIBS)
```

Change the above lines to this:

```
# different options for not windows and windows since sdl-config is not available
# for msvc compiler
!win32 {
    QMAKE_CXXFLAGS += $$system(sdl2-config -cflags)
    message(output from sdl2-config -cflags added to CXXFLAGS= $$QMAKE_CXXFLAGS)

    LIBS += $$system(sdl2-config -libs)
    message(output from sdl2-config -libs added to LIB=$$LIBS)
}
win32 {
    INCLUDEPATH += $(HOMEDRIVE)/SDL2/include/
    LIBS += -L$(HOMEDRIVE)/SDL2/lib/ -ISDL2 -ISDL2main
}
```

Another edit to be made to the .pro file is to change a warning flag for boost which raises an error. This is on line 65 (may vary due to all the lines that have been added) that looks like the following:

```
QMAKE_CXXFLAGS_WARN_ON += "-Wno-unused-parameter"
```

and change the above line to be:

```
unix*:QMAKE_CXXFLAGS_WARN_ON += "-Wno-unused-parameter"
```

You also need to change the .pro in the same way as SimpleNGL by changing the include path for the NGL include. Here is the same process repeated but with adjusted line numbers: in the .pro file you need to change the line 88 (again, may differ) which is the following:

```
INCLUDEPATH += $$$(HOME)/NGL/include/
```

and change the above line to be:

```
!win32:INCLUDEPATH += $$$(HOME)/NGL/include/
```

After that, you then you need to add to the win32 section of the .pro file from line 73 to 83 by inserting a line below 75 with:

```
INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/
```

So it should all look like this:

```
!win32:INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/

win32: {
    PRE_TARGETDEPS += C:/NGL/lib/NGL.lib
    INCLUDEPATH += -I c:/boost
    INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/
    DEFINES += GL42
    DEFINES += WIN32
    DEFINES += _WIN32
    DEFINES += _USE_MATH_DEFINES
    LIBS += -LC:/NGL/lib/ -INGL
    DEFINES += NO_DLL
}
```

In the “main.cpp” you will also need to insert a new line under `#include <SDL.h>` which is on line 1. This line is `#undef main` to give this result:

```
#include <SDL.h>
#undef main
#include <cstdlib>
```

Save the file, run qmake again (Build>Run qmake) and then build and run the project. All done! You should only get warnings but no errors should occur.

MSVS & Qt Debugger

I haven't finished this bit yet but here is it briefly:

1. Download Windows Driver Kit 8.1 from [here](#).
2. Restart Qt and it should pick up the debugger

Error Solutions

Common Errors with Example Projects

error: D8021 : invalid numeric argument '/Wno-unused-parameter'

Explanation:

This boost error gets turned into a warning on unix based machines but isn't necessary for the MSVC compiler and doesn't exist and is an invalid argument.

Solution:

In the .pro file, find the line:

```
QMAKE_CXXFLAGS_WARN_ON += "-Wno-unused-parameter"
```

and change the above line to be:

```
unix*:QMAKE_CXXFLAGS_WARN_ON += "-Wno-unused-parameter"
```

D:\Users\Jon\Documents\GitHub\SDLNGL\include\NGLDraw.h:4: error: C1083: Cannot open include file: 'ngl/Camera.h': No such file or directory

Explanation:

There is a problem with the include path which doesn't point to the include directory for NGL. The problem I found was that the examples used `$(HOME)` for the NGL include path, however this is different on the Windows build we have created, with `$(HOME)` being `C:/Users/<username>`.

Solution:

Search for the following line:

```
INCLUDEPATH += $(HOME)/NGL/include/
```

Change the above line to be:

```
!win32:INCLUDEPATH += $$$(HOME)/NGL/include/
```

After that, you then you need to add to the win32 section of the .pro file from line towards the end by inserting a line below the pre-existing INCLUDEPATH command for boost with:

```
INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/
```

So it should all look like this:

```
!win32:INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/

win32: {
    PRE_TARGETDEPS += C:/NGL/lib/NGL.lib
    INCLUDEPATH += -I c:/boost
    INCLUDEPATH += $$$(HOMEDRIVE)/NGL/include/
    DEFINES += GL42
    DEFINES += WIN32
    DEFINES += _WIN32
    DEFINES += _USE_MATH_DEFINES
    LIBS += -LC:/NGL/lib/ -INGL
    DEFINES += NO_DLL
}
```

D:\Users\Jon\Documents\GitHub\SDLNGL\src\main.cpp:125: error: C4716: 'SDL_main' : must return a value

Explanation:

The SDL.h include hasn't had its main undefined, preventing the int main function from being unable to compile without a return statement.

Solution:

In the "main.cpp" you will need to insert a new line under `#include <SDL.h>` which is on line 1. This line is `#undef main` to give this result:

```
#include <SDL.h>
#undef main
#include <cstdlib>
...
```

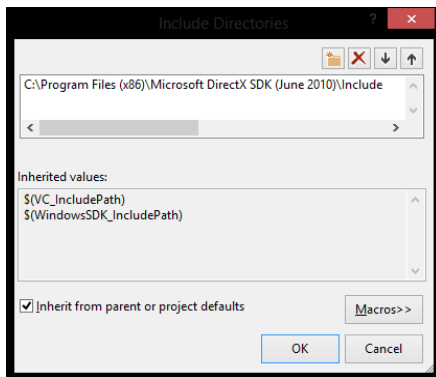

Now the file should build and run without errors.

SDL Build Errors

**error C1083: Cannot open include file: 'dxsdkver.h': No such file or directory
D:\Users\Jon\Downloads\NEW_NGL\SDL2-2.0.3\src\audio\audio2\SDL_audio2.c**

Explanation:

The DirectX include file “dxsdker.h” can’t be found when building. Either it’s not in the right directory or for some reason Visual Studio doesn’t know where to look.



Solution:

I'm not 100% on this solution since I wasn't able to recreate this error without deleting all of the DirectX SDK installation folder.

Firstly, try restarting your computer and re-opening the VisualC solution. If trying to build after this doesn't work, we need to have Visual Studio point to the exact location of the Include and Lib files we built with the DirectX SDK. To do this, right click on the **SDL2** project on the right in the Solution Explorer and click **Properties**. Now on the left go to **Configuration Properties>VC++ Directories** and click on the **Include Directories** value on the right and then click the

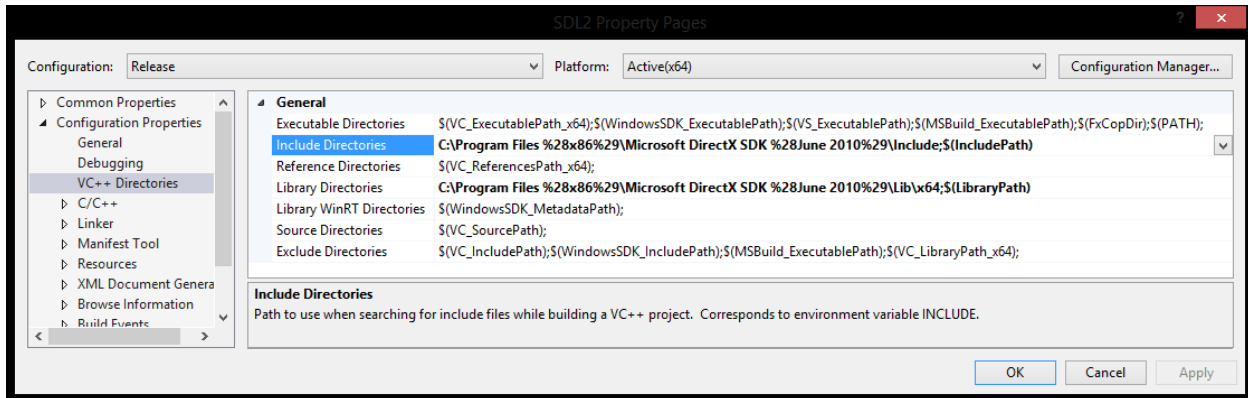
drop down on the right of the box. Follow this with pressing <Edit...> and a new dialog will appear. Press the folder with the * in it to create a new line, and then either paste the directory in the entry box or navigate to the folder:

C:\Program Files (x86)\Microsoft DirectX SDK (June 2010)\Include

Do the same for the **Library Directories** and add the directory:

C:\Program Files (x86)\Microsoft DirectX SDK (June 2010)\Lib\x64

It should look something like this when you have finished adding these directories:



Once you have done this, in the menu at the top just press **BUILD>Build Solution**. This should fix the error and create only warnings.