



How to Set Up SDL with Visual Studio

Author Info

Whether you're a novice C++ programmer or a veteran software developer, graphics programming is an educational and rewarding experience. Simple DirectMedia Layer is a C library (compatible with C++) that allows simple, low-level access, to the graphics functionality of a variety of platforms. This guide will help you get over the first challenge of using SDL: installing and setting it up. It is assumed you work with Windows (7, 8, or 10) and Visual Studio 2019.

After setting up SDL on a project, create an Export Template (See last part "Creating Project with SDL Template") to create SDL projects without setting up again.

Explore this Article

[Configuring Visual Studio](#)
[Downloading SDL](#)
[Creating Visual Studio Project](#)
[Show 3 more...](#)

Related Articles

Part 1

Configuring Visual Studio

- 1 Download Visual Studio 2019.** If you have not done so you can download it from <https://www.visualstudio.com/downloads>.
- 2 Check the *Desktop development with C++ workload* box as you download.** If necessary scroll down the page.

Part 2

Downloading SDL

- 1 Create folder *SDL*.** In directory (disk) C:, create (by right click > New > Folder) new folder called **SDL**.
- 2 Download the *SDL2-devel-2.0.9-VC.zip (Visual C++ 32/64-bit)*.** It can be found at <https://www.libsdl.org/download-2.0.php>. Scroll to the bottom of the page and find *Development Libraries*. Download the version for Visual C++. It is *SDL2-devel-2.0.9-VC.zip (Visual C++ 32/64-bit)*.
 - In downloading window you have the zip folder **SDL2-devel-2.0.9-VC.zip**, and inside the folder **SDL2-2.0.9**. Click it > right click > select *Copy*.
 - Navigate to C:\SDL. Inside SDL, right click > select *Paste*. Click on name SDL2-2.0.9 and rename to **SDL2**. Now in folder **SDL** you have folder **SDL2**.

Part 3

Creating Visual Studio Project

- 1 Create an empty project.** Open Visual Studio 2019. Click > > .
 - In "Project name" text field type: **SDLproject**.
 - Next to "Location" text field click > Navigate to C: > SDL > click . Now "Location" is **C:\SDL**.
 - Check box next to "Place solution and project in the same directory" > click . Wait till *Visual Studio 2019* instance appears.
- 2 Add your Source file to the project.**
 - In "Solution Explorer" window right click the "Source Files" folder (the last one).
 - Click "Add > *New Item...*"
 - In the "Add New Item - SDLproject" window, click "C++ File (.cpp)", the first one. "Name:" **Source.cpp** is okay.
 - The "Location" should be C:\SDL\SDLproject\. If it's not so, click to the right of the field and navigate to C: > SDL > SDLproject. Click "Select folder". Now "Location" is **C:\SDL\SDLproject**.

**Part
4**

Installing **SDL2** on Project

1 Configure the "Additional Include Directories".

- In "Solution Explorer" right click on the name of your project, that is **SDLproject**, and select "Properties".
- Open the C/C++ drop-down menu. Click "General" > "Additional Include Directories" > down arrow at the right of the field > "Edit" in the drop-down menu.
- At the top of the "Additional Include Directories" window, click the brown folder icon (the first icon). Click next to the text box.
- In the "Select Directory" window, navigate (by double click): C: > SDL > SDL2, and click "include" folder. Make sure you have selected the "include" folder. The "Folder" is "include". Click on the "Select Directory" window and in the "Additional Include Directories" window.

2 Configure the linker "Additional Library Directories"

- Open the "Linker" drop-down menu, and click "General". Click "Additional Library Directories" entry, and click the down arrow at the right of the field. Click "Edit" in the drop-down menu.
- At the top of the "Additional Library Directories" window, click the first icon. Then click next to the text box.
- In the "Select Directory" window, navigate C: > SDL > SDL2, and double click "lib" folder, followed by clicking "x86" folder. The "Folder" is "x86". Click on "Select Directory" window and on "Additional Library Directories" window.

3 Configure the linker "Additional Dependencies"

- In the "Linker" drop-down menu, click "Input". Click the "Additional Dependencies" entry > the down arrow at the right of the field > "Edit" in the drop-down menu.
- In "Additional Dependencies" window copy and paste **SDL2.lib; SDL2main.lib** in the top-most text box. Click in the "Additional Dependencies" window.

4 Set linker "SubSystem" to "Console"

- In the "Linker" drop-down menu, click "System" > "SubSystem". Click the down arrow and select "Console(/SUBSYSTEM:CONSOLE)" from the dropdown menu. Click , then on the "SDLproject Property Pages" window.

5 Copy "SDL2.dll" file and paste into project-folder

- in "File Explorer" navigate C: > SDL > SDL2 > lib > x86. In "x86" folder click "SDL2.dll" file > right-click > "Copy".
- Navigate C: > SDL > SDLproject. Right-click on empty area in "SDLproject" folder, and select "Paste".
- The "SDL2.dll" file is now in project-folder "SDLproject" along with "Source.cpp" and other 4 files created by Visual Studio.

**Part
5**

Testing and Debugging your Project

- 1 **Test your project.** Copy following code and paste in *Source.cpp* code area. Click the "Local Windows Debugger" button next to the green arrow, above the code area. If the project builds and the console prints "SDL initialization succeeded!" then the project was set up correctly. You are now ready to program with SDL! If your project did not compile or you got an error message see the debugging section

```
#include <iostream>
#include <SDL.h>
using namespace std;

int main(int argc, char * argv[])
{
    if (SDL_Init(SDL_INIT_EVERYTHING) < 0)
    {
        cout << "SDL initialization failed. SDL Error: " << SDL_GetError();
    }
    else
    {
        cout << "SDL initialization succeeded!";
    }

    cin.get();
    return 0;
}
```

2 Correct any errors. If you fail, repeat above steps.

Part 6

Creating Project with SDL Template

1 Create a project template. Go to Visual Studio and, while "SDLproject" is open, click "Project" > "Export Template...". On "Export template Wizard" check "Project Template", if it's not checked. Click . On "Select Template Options", for "Template name:" type **SDL**. Click . Template has been created.

2 Create SDL project with created template

- On V.S. click "File" > "New" > "Project...". > click template: "SDL". In "Name:" text field, type: **SDLproject-1** > Check box next to "Place solution and project in the same directory" > click .
- Navigate C: > SDL > SDLproject > click file "SDL2.dll" > right click > click "Copy".
- Navigate C: > SDL > SDLproject-1 > click on empty area > right click > click "Paste".
- Now a copy of the file "SDL2.dll" is in folder "SDLproject-1" among "Source.cpp" and other 4 files.
- In "Solution Explorer" double click "Source.cpp". Delete code and paste a new one. Click "Local Windows Debugger", to the right of green arrow, just above V.S. editor's area. If everything is done well, eventually the black window appears saying: "SDL initialization succeeded!".

3 Use your template. Creating a project with an SDL-project-template is just like creating an ordinary C++ project, but with one more step:

- The file "SDL2.dll" should be copied from a previous SDL project and pasted into new project.

Community Q&A

Question

What does unresolved external symbol "referenced in" mean?

Bryan Hadland
Community Answer

An unresolved external symbol means that you have not setup the project fully. By this I mean make sure you have linked against the library that your using.

Tips

- After you have finished this lesson, you might want to learn how to [Set Up an OpenGL SDL GLEW Template Project in Visual Studio 2017](#).

- Also to learn how to [Set Up an OpenGL-GFW-GLAD Template Project in Visual Studio 2017](#).
- Also [Set Up SFML in a Project on Visual Studio 2017](#)