

A.3 — Using libraries with Code::Blocks

BY ALEX ON JULY 3RD, 2007 | LAST MODIFIED BY ALEX ON JANUARY 23RD, 2020

To recap the process needed to use a library:

Once per library:

- 1) Acquire the library. Download it from the website or via a package manager.
- 2) Install the library. Unzip it to a directory or install it via a package manager.
- 3) Tell the compiler where to look for the header file(s) for the library.
- 4) Tell the linker where to look for the library file(s) for the library.

Once per project:

- 5) Tell the linker which static or import library files to link.
- 6) #include the library's header file(s) in your program.
- 7) Make sure the program know where to find any dynamic libraries being used.

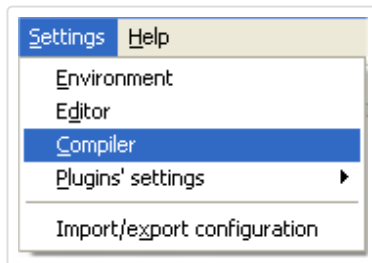
Steps 1 and 2 -- Acquire and install library

Download and install the library to your hard disk. See the tutorial on [static and dynamic libraries](#) for more information about this step.

Steps 3 and 4 -- Tell the compiler where to find headers and library files

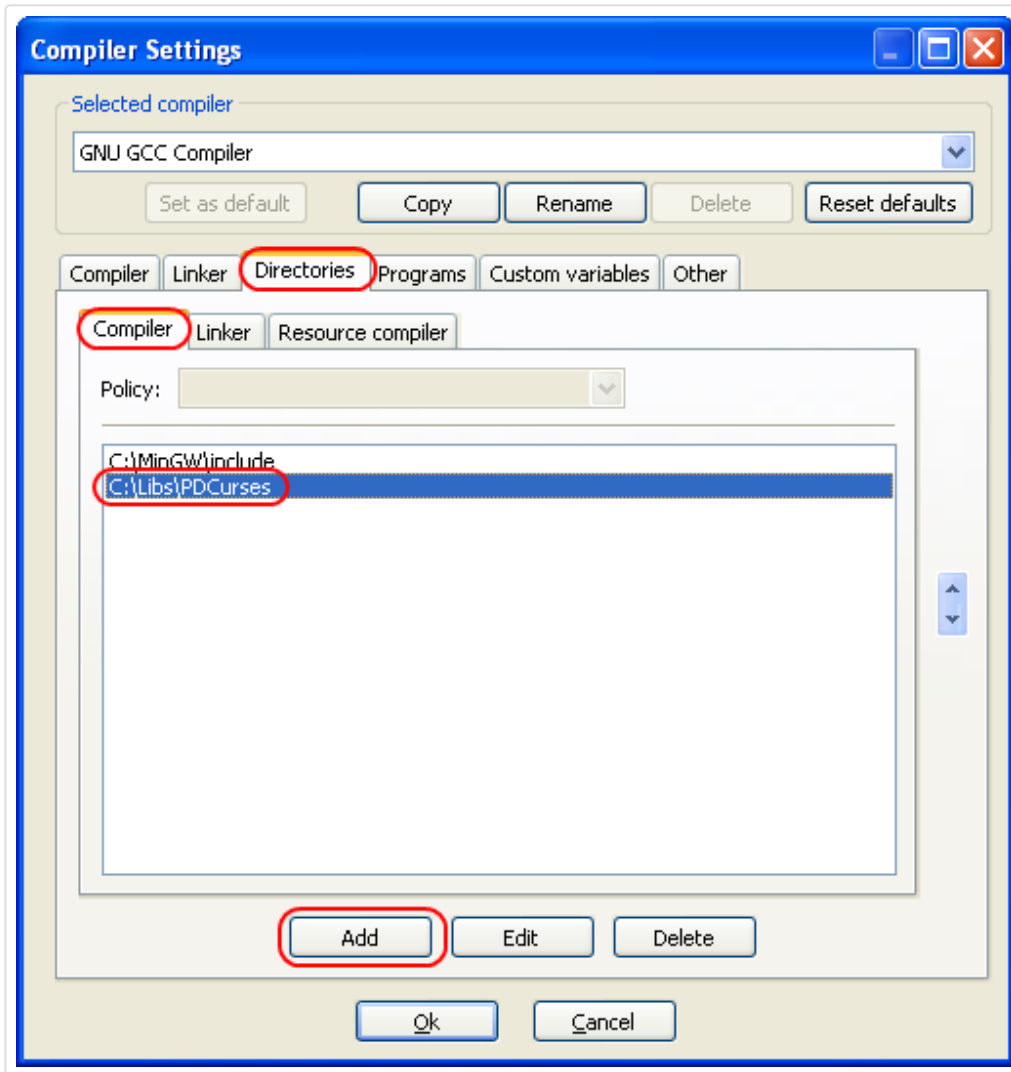
We are going to do this on a global basis so the library will be available to all of our projects. Consequently, the following steps only need to be done once per library.

- A) Go to the “Settings menu” and pick “Compiler”.

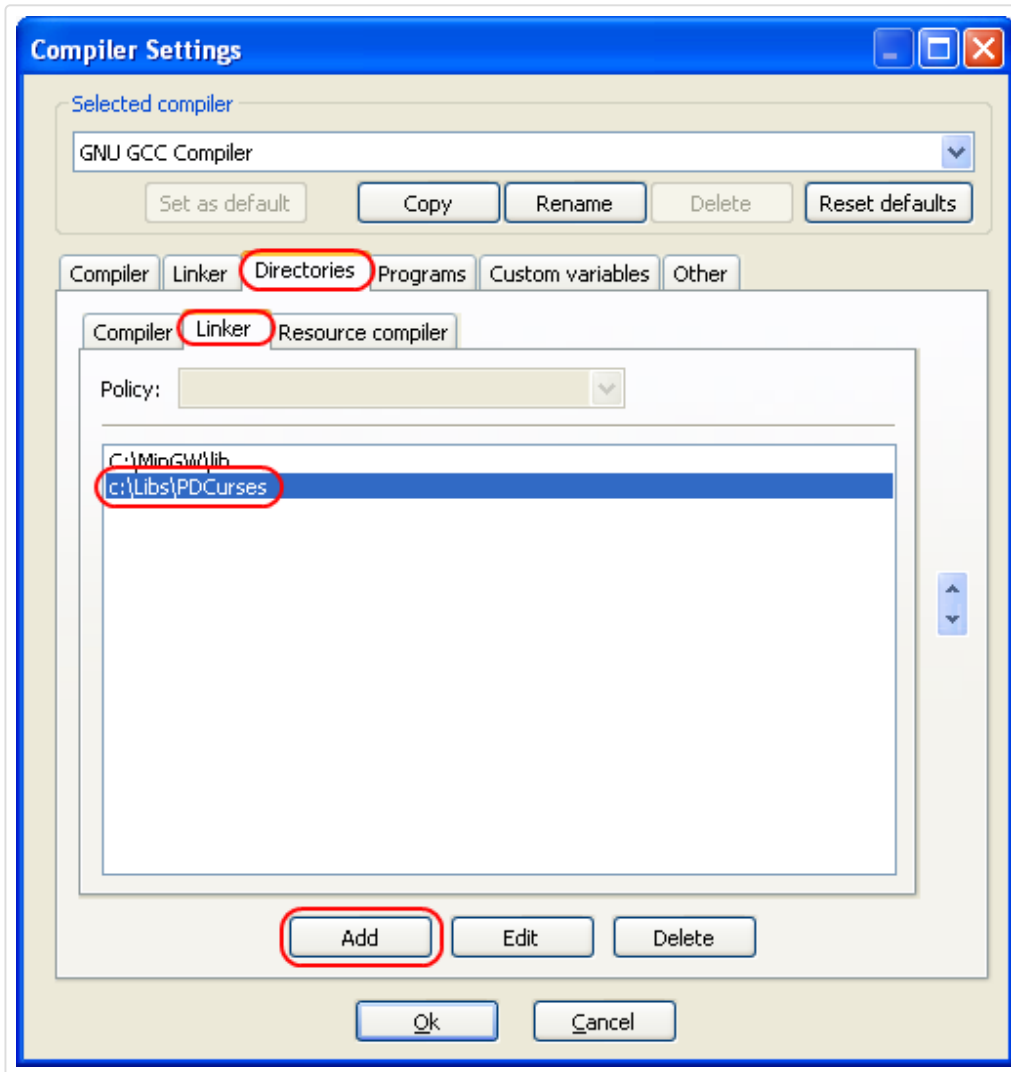


- B) Click the “Directories” tab. The compiler tab will already be selected for you.

- C) Press the “Add” button, and add the path to the .h files for the library. If you are running Linux and installed the library via a package manager, make sure `/usr/include` is listed here.



D) Click the “Linker” tab. Press the “Add” button, and add the path to the .lib files for the library. If you are running Linux and installed the library via a package manager, make sure */usr/lib* is listed here.

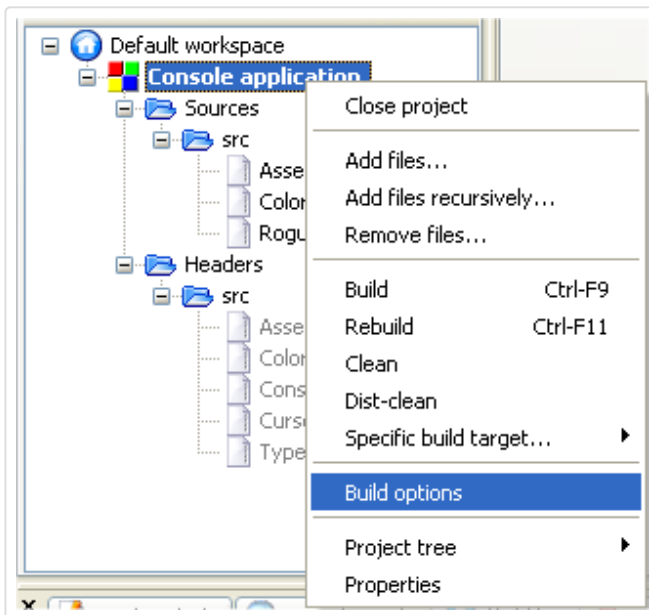


E) Press the "OK" button.

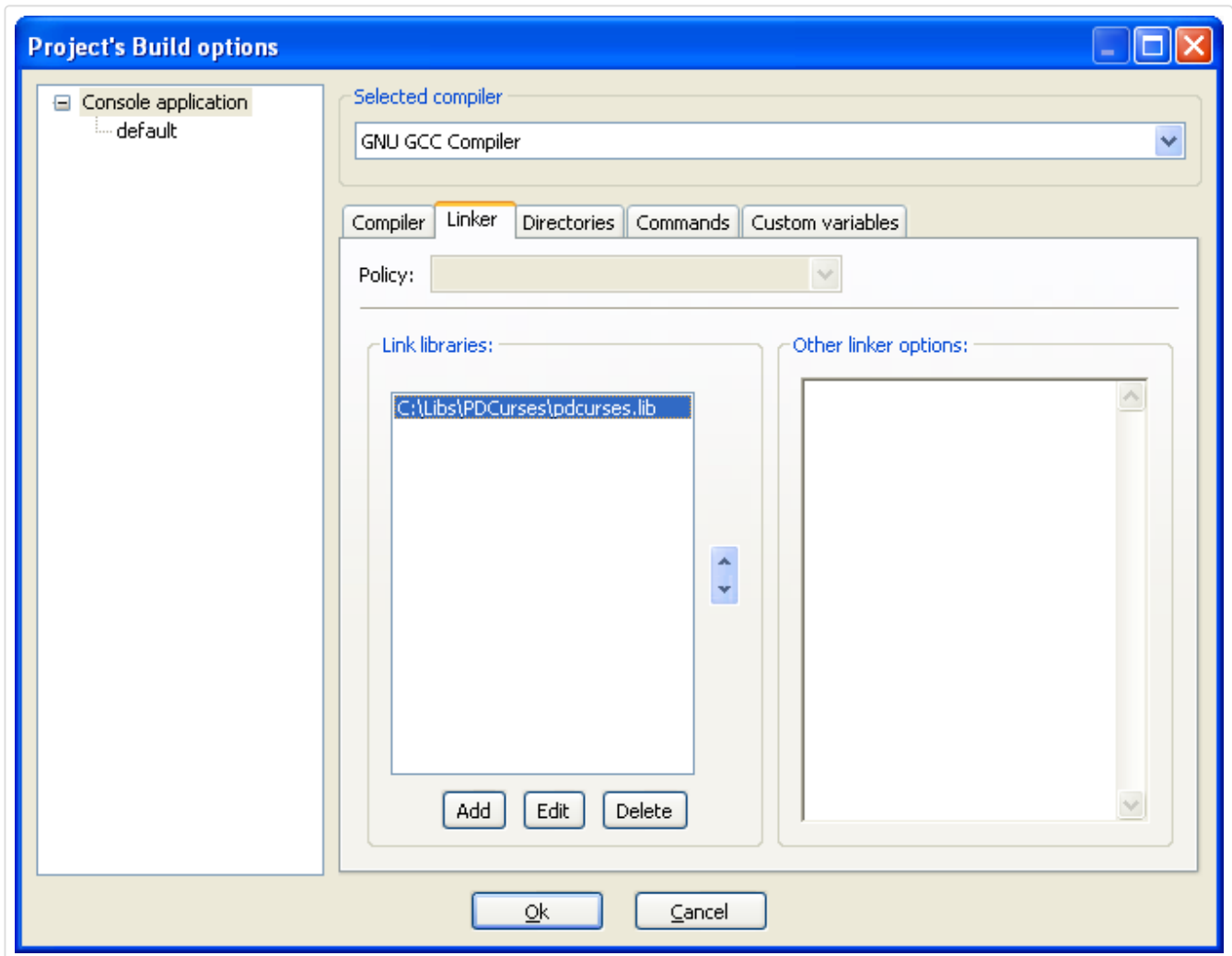
Step 5 -- Tell the linker which libraries your program is using

For step 5, we need to add the library files from the library to our project. We do this on an individual project basis.

A) Right click on the bolded project name under the default workspace (probably "Console application", unless you changed it). Choose "Build options" from the menu.



B) Click the linker tab. Under the “Link libraries” window, press the “Add” button and add the library you wish your project to use.



C) Press the “OK” button

Steps 6 and 7 -- #include header files and make sure project can find DLLs

Simply #include the header file(s) from the library in your project.

See the tutorial [A.1 -- static and dynamic libraries](#) for more information step 7.



[B.1 -- Introduction to C++11](#)



[Index](#)



[A.2 -- Using libraries with Visual Studio](#)

[C++ TUTORIAL](#) | [PRINT THIS POST](#)

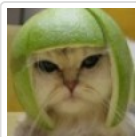
51 comments to A.3 — Using libraries with Code::Blocks



George Pitchurov

[January 28, 2020 at 10:27 am](#) · [Reply](#)

Why do we need step 5? Why is it not enough to #include the header file in our program and let the IDE link it? The IDE already knows perfectly well where to look for both library headers and code (steps 3 and 4). How is it we don't care about this step when using the standard libraries?



Alex

[February 2, 2020 at 12:35 pm](#) · [Reply](#)

specified.

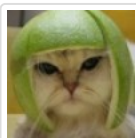
The standard libraries will automatically link in if you use anything from them. Unfortunately, this doesn't extend to user-defined libraries, so they need to be explicitly



masterOfNothing

[December 5, 2019 at 9:21 am](#) · [Reply](#)

I'm completely blank on this. I went to SDL website. Just for testing downloaded a .zip file for Windows. It only has one SDL2.dll file. What do I do with it? For some reason adding packages just didn't click. It is dynamic link library extension.



Alex

[December 6, 2019 at 5:54 pm](#) · [Reply](#)

Download the development libraries for your IDE, not the runtime binaries.

masterOfNothing

[December 22, 2019 at 10:27 am](#) · [Reply](#)

Hello, Alex.