Lazy Foo' Productions

Setting up SDL 2 on Code::Blocks 12.11

Last Updated 9/17/13

1)First thing you need to do is download SDL headers, libary and binaries. You will find them on the SDL website, specifically on this page.

Since Code::Blocks comes with the MinGW compiler by default, odds are you'll want to download the MinGW development libraries.

SDL version 2.0.0 (stable)

Source Code:

SDL2-2.0.0.zip - GPG signed
SDL2-2.0.0.tar.gz - GPG signed

Runtime Binaries:

Windows:
SDL2-2.0.0-win32-x86.zip (32-bit Windows)
SDL2-2.0.0-win32-x64.zip (64-bit Windows)

Mac OS X:
SDL2-2.0.0.dmg (Intel 10.5+)

Linux:
Please contact your distribution maintainer for updates.

Development Libraries:

Windows:
SDL2-devel-2.0.0-VC.zip (Visual C++ 32/64-bit)
SDL2-devel-2.0.0-mingw.tar.gz (64-bit)

Open the gzip archive and there should be a tar archive. Open up the tar archive and the should be a folder called SDL2-2.*something.something*. In side of that folder there should be a bunch of folders and files, most importantly i686-w64-mingw32 which contains the 32bit library and x86_64-w64-mingw32 which contains the 64bit library.

This is important: most compilers still compile 32bit binaries by default to maximize compatibility. We will be using the 32bit binaries for this tutorial set. It doesn't matter if you have a 64bit operating system, since we are compiling 32bit binaries we will be using the 32bit library.

Inside of i686-w64-mingw32 are the include, lib, and bin folders which contain everything we need compile and run SDL applications. Copy the contents of i686-w64-mingw32 to any directory you want. I recommend putting it in a folder that you dedicate to holding all your development libraries for MinGW. For these tutorials I'm putting it in a directory I created C:\mingw_dev_lib
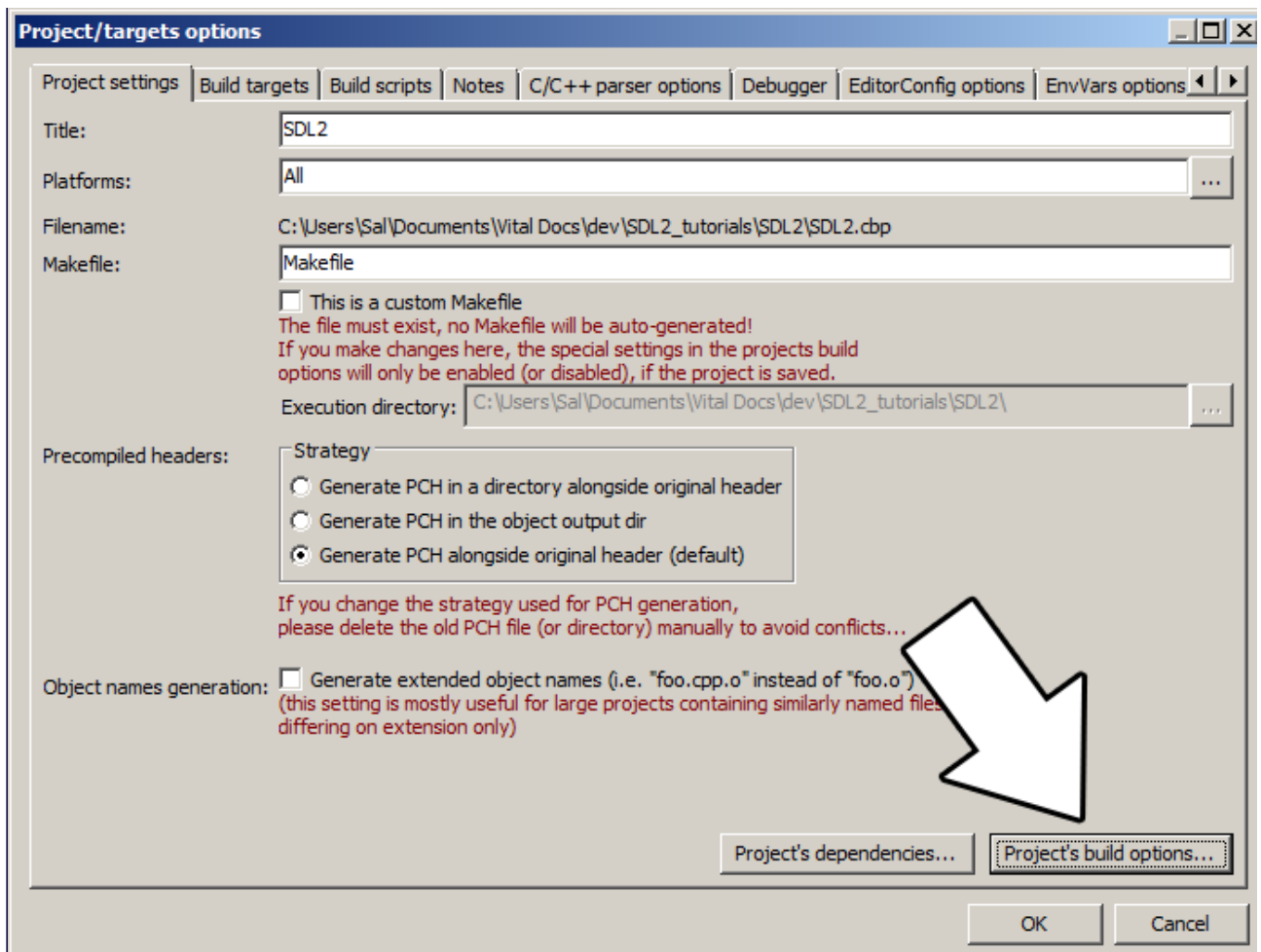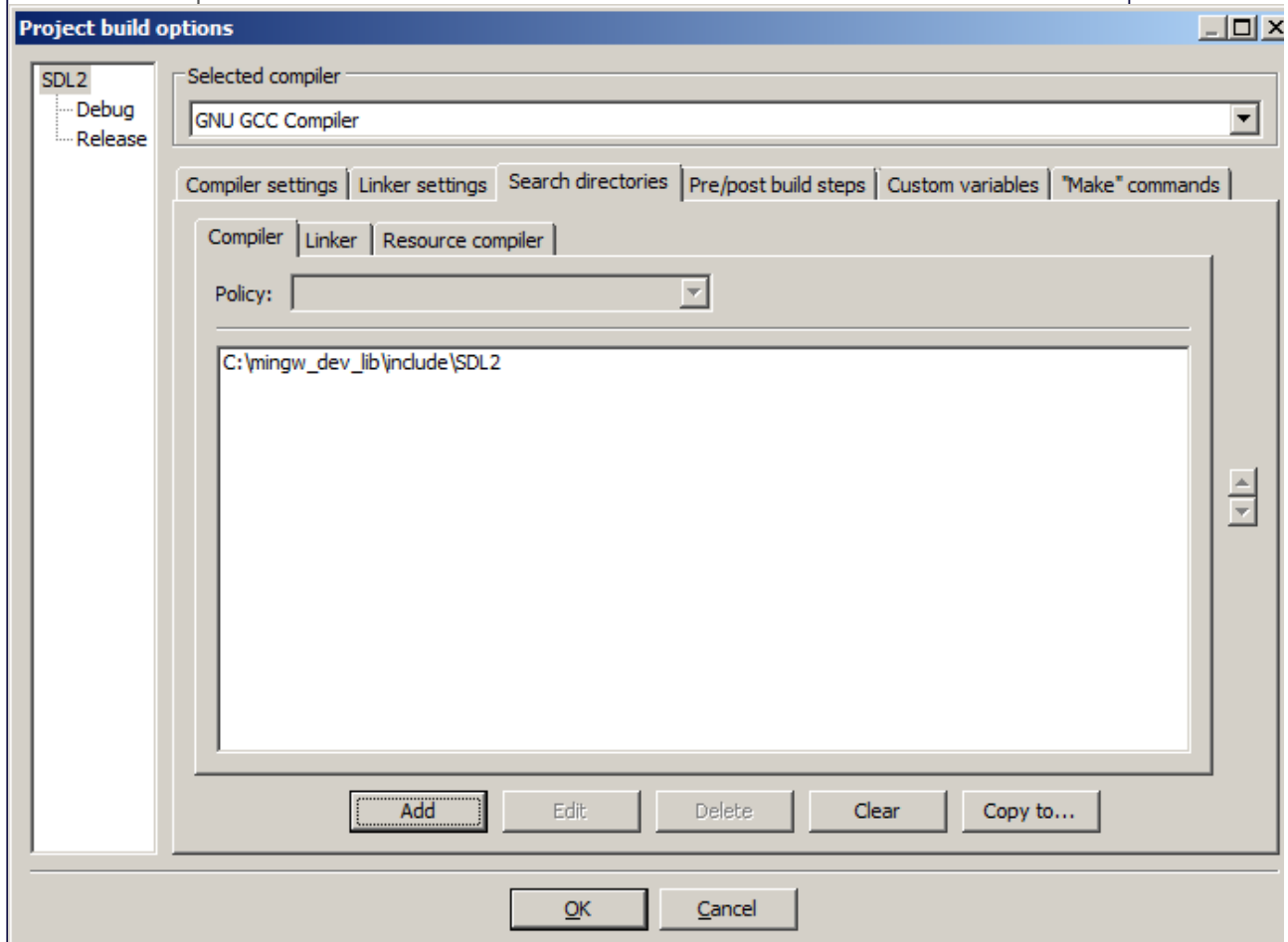
2)Start up Code::Blocks and create a new empty project.
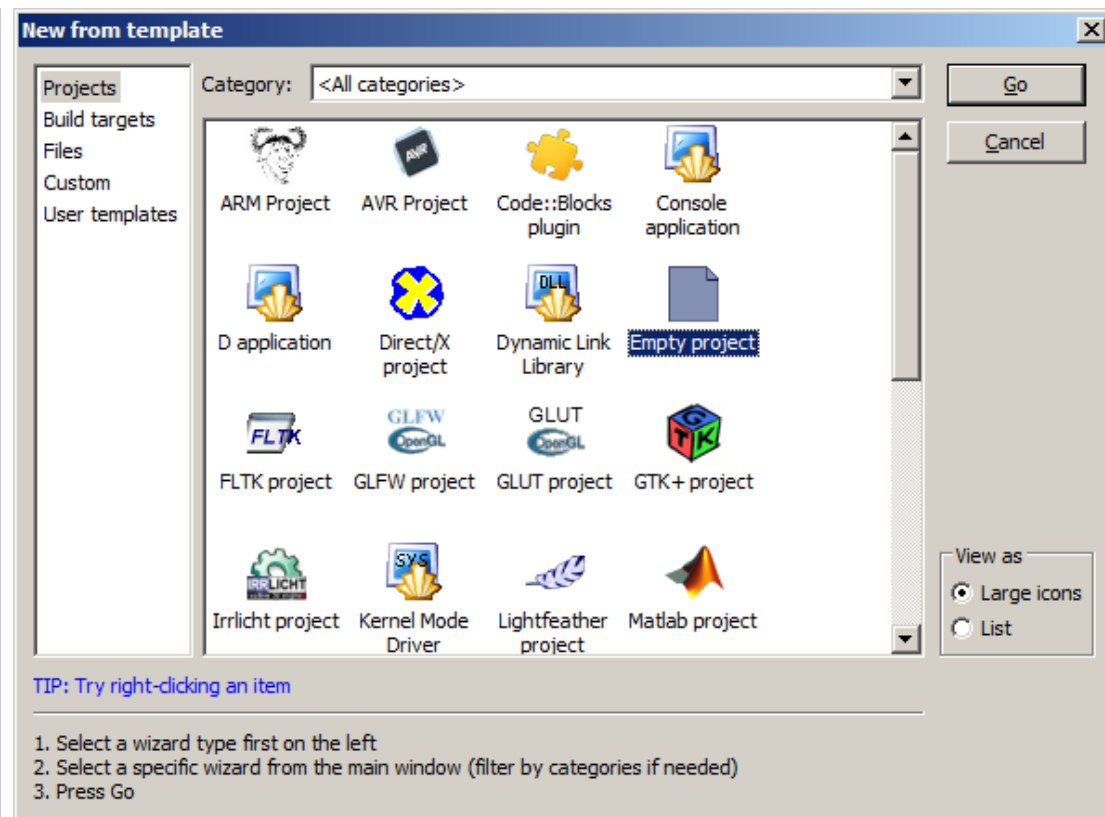
3)Go to project properties.
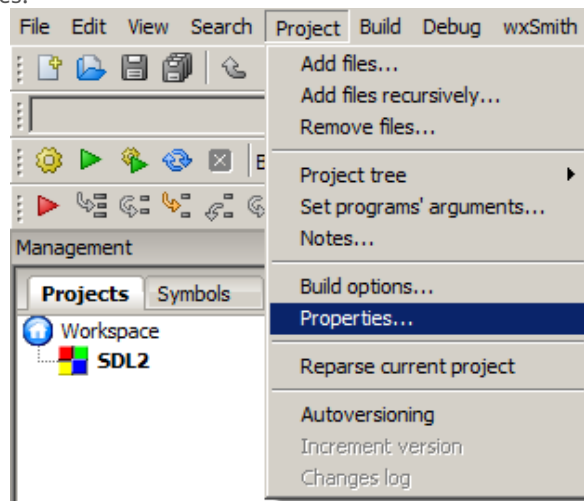
4)Now we have to tell Code::Blocks to search for header files in the library folder we just extracted. Go to build options.

Project/targets options   _ □ ×

| Project settings | Build targets | Build scripts | Notes | C/C++ parser options | Debugger | EditorConfig options | EnvVars options |

Title: SDL2

Platforms: All   ...

Filename: C:\Users\Sal\Documents\Vital Docs\dev\SDL2_tutorials\SDL2\SDL2.cbp

Makefile: Makefile

☐ This is a custom Makefile
The file must exist, no Makefile will be auto-generated!
If you make changes here, the special settings in the projects build
options will only be enabled (or disabled), if the project is saved.

Execution directory: C:\Users\Sal\Documents\Vital Docs\dev\SDL2_tutorials\SDL2\   ...

Precompiled headers: ┌ Strategy ───┐
                        │ ○ Generate PCH in a directory alongside original header │
                        │ ○ Generate PCH in the object output dir   │
                        │ ⦿ Generate PCH alongside original header (default)   │
                        └──┘

If you change the strategy used for PCH generation,
please delete the old PCH file (or directory) manually to avoid conflicts...

Object names generation: ☐ Generate extended object names (i.e. "foo.cpp.o" instead of "foo.o")
(this setting is mostly useful for large projects containing similarly named files
differing on extension only)

                                      Project's dependencies...   Project's build options...

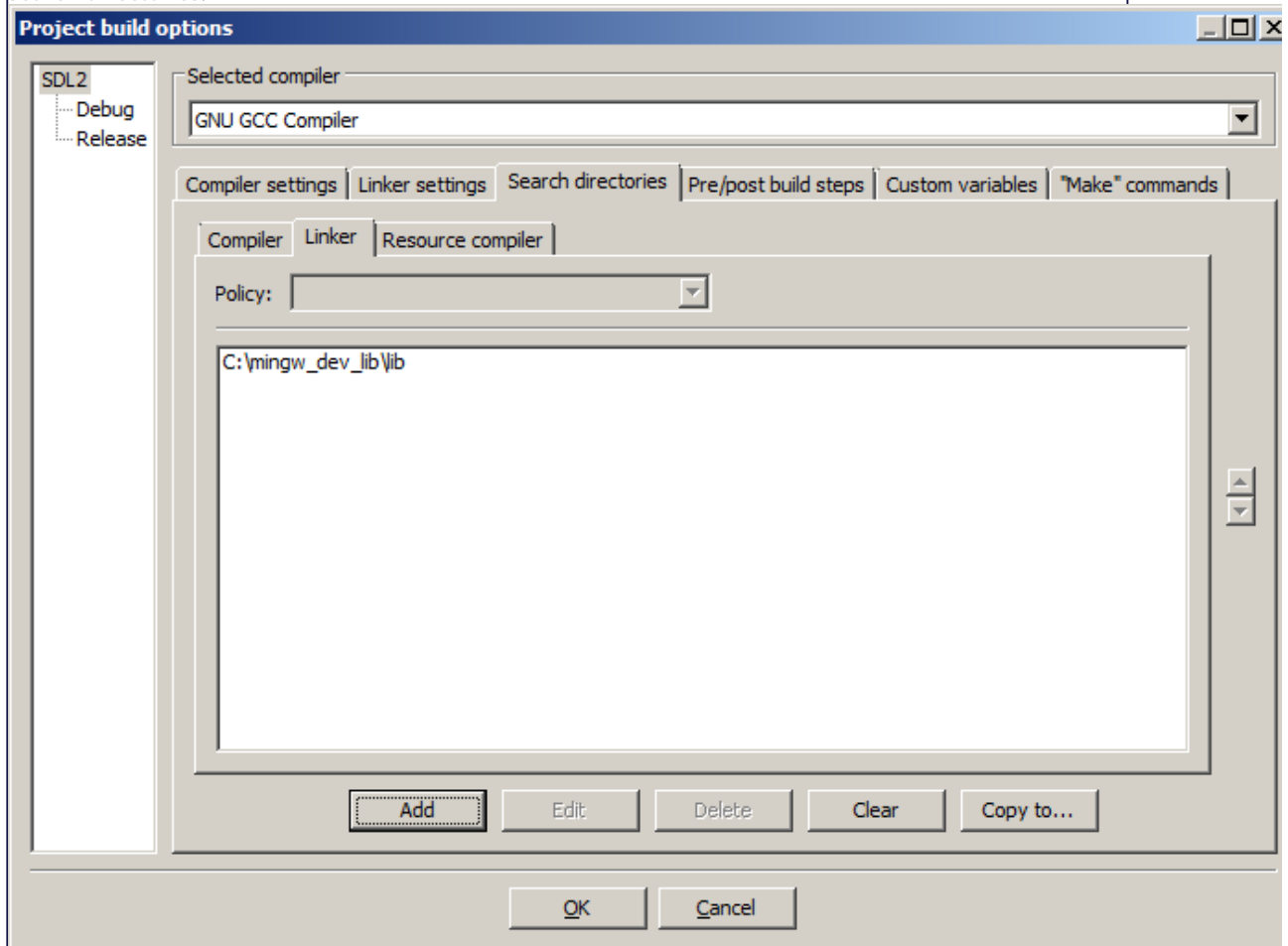   OK   Cancel

In the Search Directories, we need to add a new compiler directory. Click add, Select the SDL2 folder inside of the include directory from the folder we extracted. Say no when it asks you whether you want it to be a relative path. Now Code::Blocks knows where to find the SDL 2 header files.

Project build options   _ □ ×

SDL2   ┌ Selected compiler ───┐
  Debug   │ GNU GCC Compiler   ▼ │
  Release   └──┘

| Compiler settings | Linker settings | Search directories | Pre/post build steps | Custom variables | "Make" commands |

| Compiler | Linker | Resource compiler |

Policy: [▼]

C:\mingw_dev_lib\include\SDL2

                                                                                    ▲
                                                                                    ▼

        Add   Edit   Delete   Clear   Copy to...

                              OK   Cancel

If you get an error where the compiler says it can't find SDL.h, it means you messed up this step.

5)Next we are going to tell Code::Blocks to search for library files in the SDL folder we just extracted. All you have to is go to the linker tab and add the lib directory from the folder we extacted to the linker search directories.

Project build options

Selected compiler
GNU GCC Compiler

| SDL2 |
| Debug |
| Release |

Compiler settings | Linker settings | Search directories | Pre/post build steps | Custom variables | "Make" commands

Compiler | Linker | Resource compiler

Policy:

C:\mingw_dev_lib\lib

Add | Edit | Delete | Clear | Copy to...

OK | Cancel

If you get an error where the linker complains it can't find -lSDL2 or -lSDL2main, it means you messed up this step.

6)In order to compile SDL 2 code, we have to tell the compiler to link against the libraries. Go under Linker Settings and paste

-lmingw32 -lSDL2main -lSDL2

into the other linker options field and click OK.
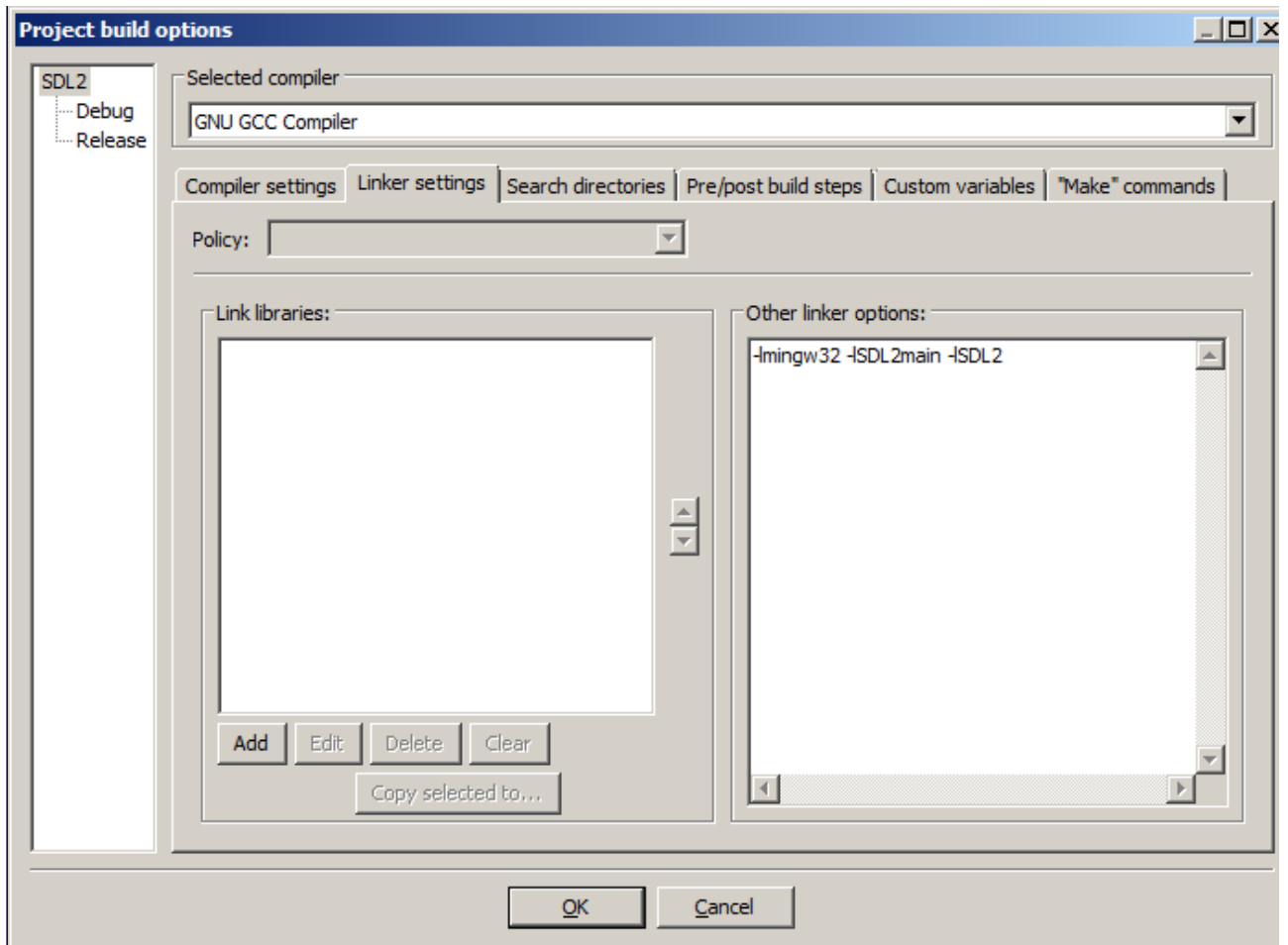
If you get an error where the linker complains about a bunch of undefined references, it means you messed up this step.

7)Go back to the project properites and under Build Targets select the build type.
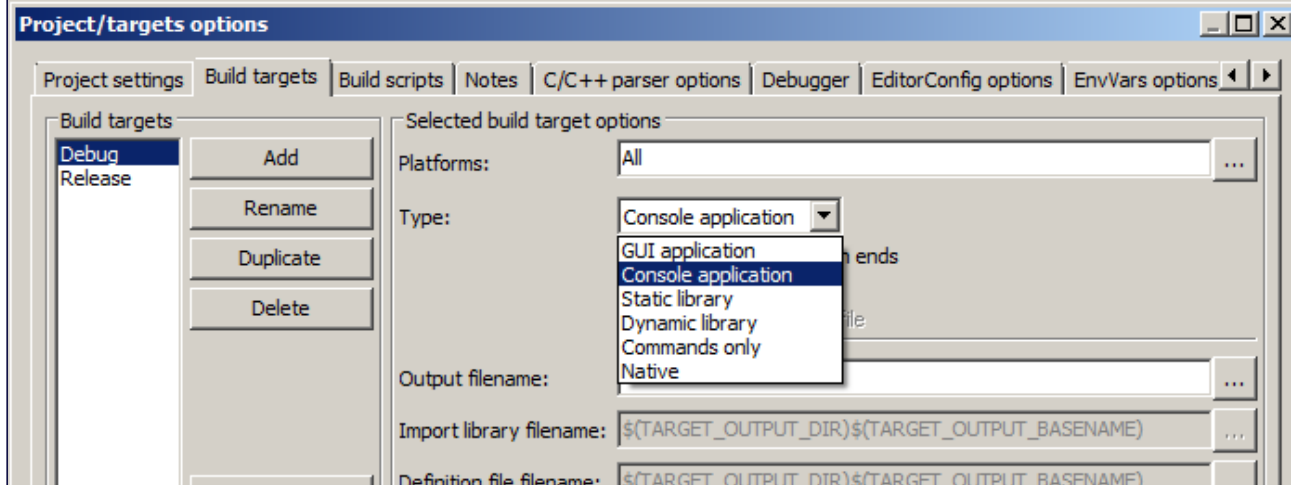
You can set it to GUI Application if you don't want console output, and Console Application if you do want console output. For this tutorial set, errors will be printed on the console, so I recommend leaving it as a console application as you are still learning and prototyping.

8)When our SDL 2 application runs, the operating system needs to be able to find the dll file.

Go find the SDL 2 folder you extracted and from the bin folder inside copy SDL2.dll and put it either where your executable will run, or inside of the system directory. C:\WINDOWS\SYSTEM32 is the 32bit windows system directory and C:\Windows\SysWOW64 is the 64bit system directory of 32bit applications. For these tutorials, I'm assuming we're making 32bit applications. If you get an error when you run the program where it complains that it can't find SDL2.dll, it means you messed up this step.

9)Now go download the source for lesson 01. Add the source files inside to your project.

Now build. If there are any errors, make sure you didn't skip a step.

Now that you have SDL 2 compiling, it time to go onto part 2 of the tutorial.

Hello SDL Part 2: Your First Graphics Window

Copyright Lazy Foo' Productions 2004-2015