

## A.2 — Using libraries with Visual Studio

BY ALEX ON JUNE 30TH, 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.

Note: The examples in this lesson show screenshots from Visual Studio 2005 express, but the process hasn't changed too substantively since then.

### 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.

#### For Visual Studio users

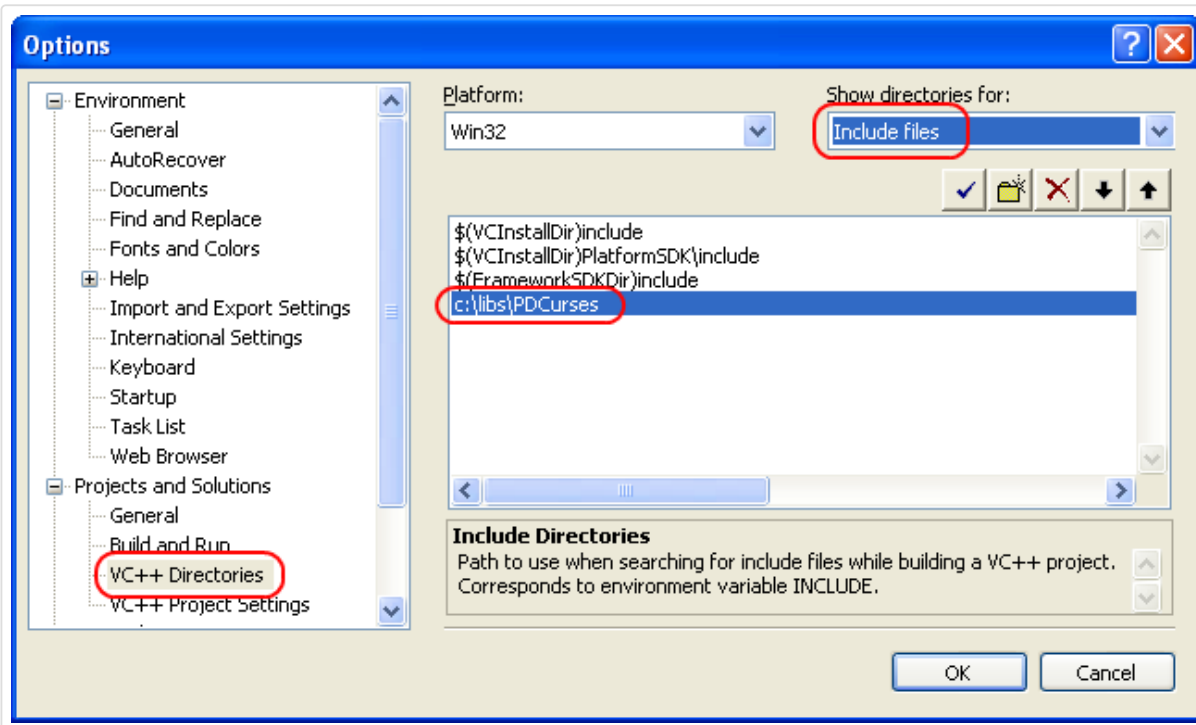
The ability to do this globally has been deprecated as of Visual Studio 2013. These options are now set per project via Project Menu -> Properties. Some of the names and locations have changed, but the general idea is still the same.

A) Go to the "Tools menu" and pick "Options".

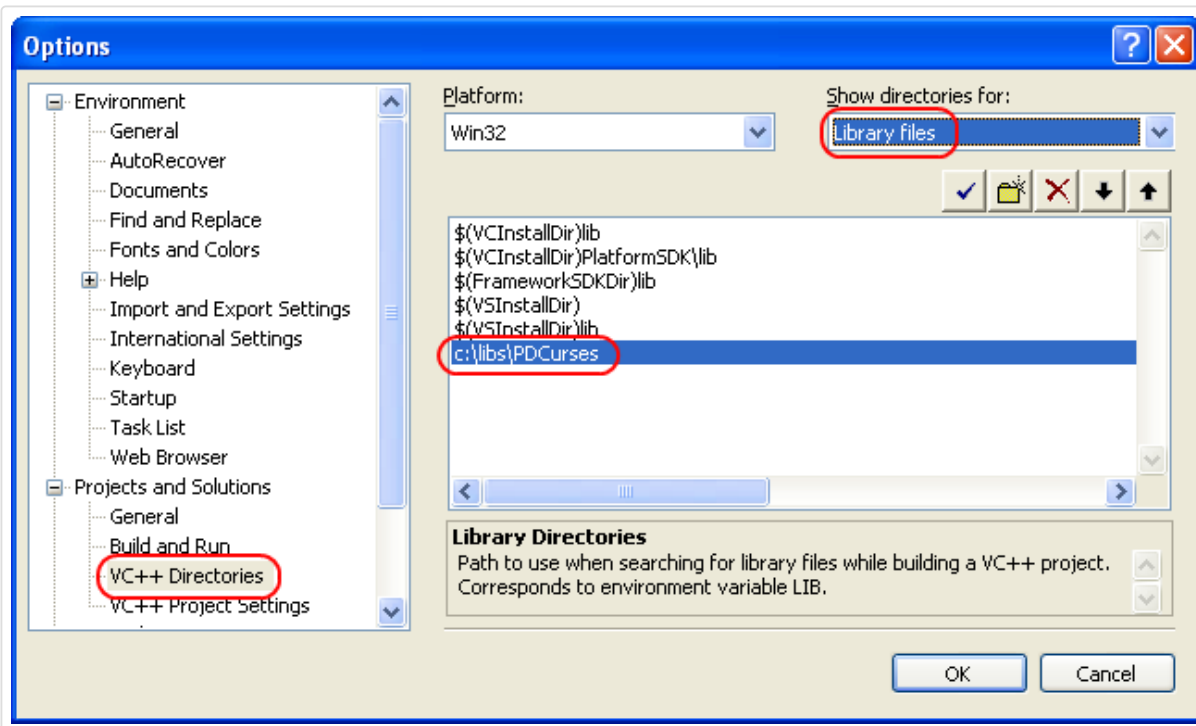


B) Open the "Projects and Solutions" node, and click on "VC++ Directories".

C) Under "Include Files"/"Include Directories", add the path to the .h files for the library.



D) Under "Library Files"/"Library Directories", add the path to the .lib files for the library.



E) Click "OK".

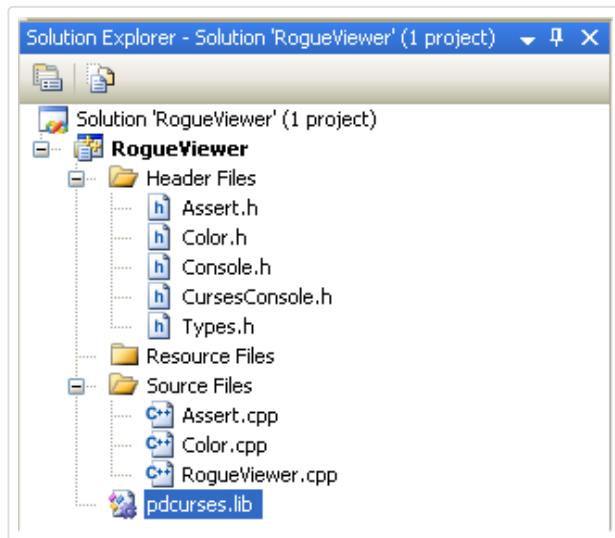
### Step 5 -- Tell the linker which libraries your program is using

For step 5, we need to add .lib files from the library to our project. We do this on an individual project basis. Visual Studio offers us 3 different methods for adding .lib files to our project.

A) Use a #pragma preprocessor directive to your primary .cpp file. This solution only works with Visual Studio and is non-portable. Other compilers will ignore this line.

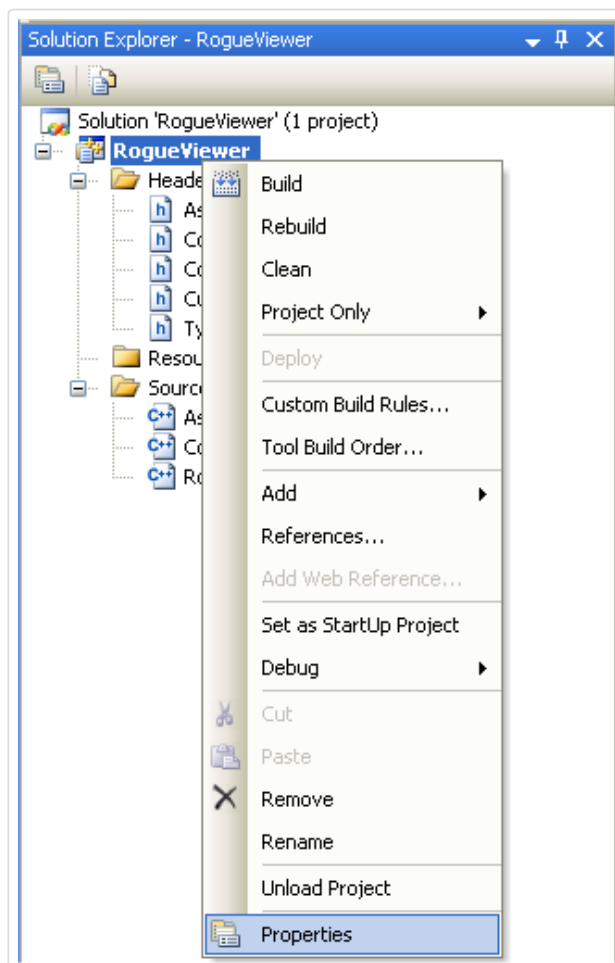
```
#include "curses.h"
#pragma comment(lib, "PDCurses.lib")
```

B) Add the .lib file to your project as if it were a .cpp or .h file. This solution works with Visual Studio, but not with many other compilers. This is the solution we recommend.



C) Add the library to the linker input. This is the most “portable” solution in the sense that every IDE will provide a similar mechanism. If you ever move to another compiler or IDE, this is the solution you will have to use. This solution requires 5 steps:

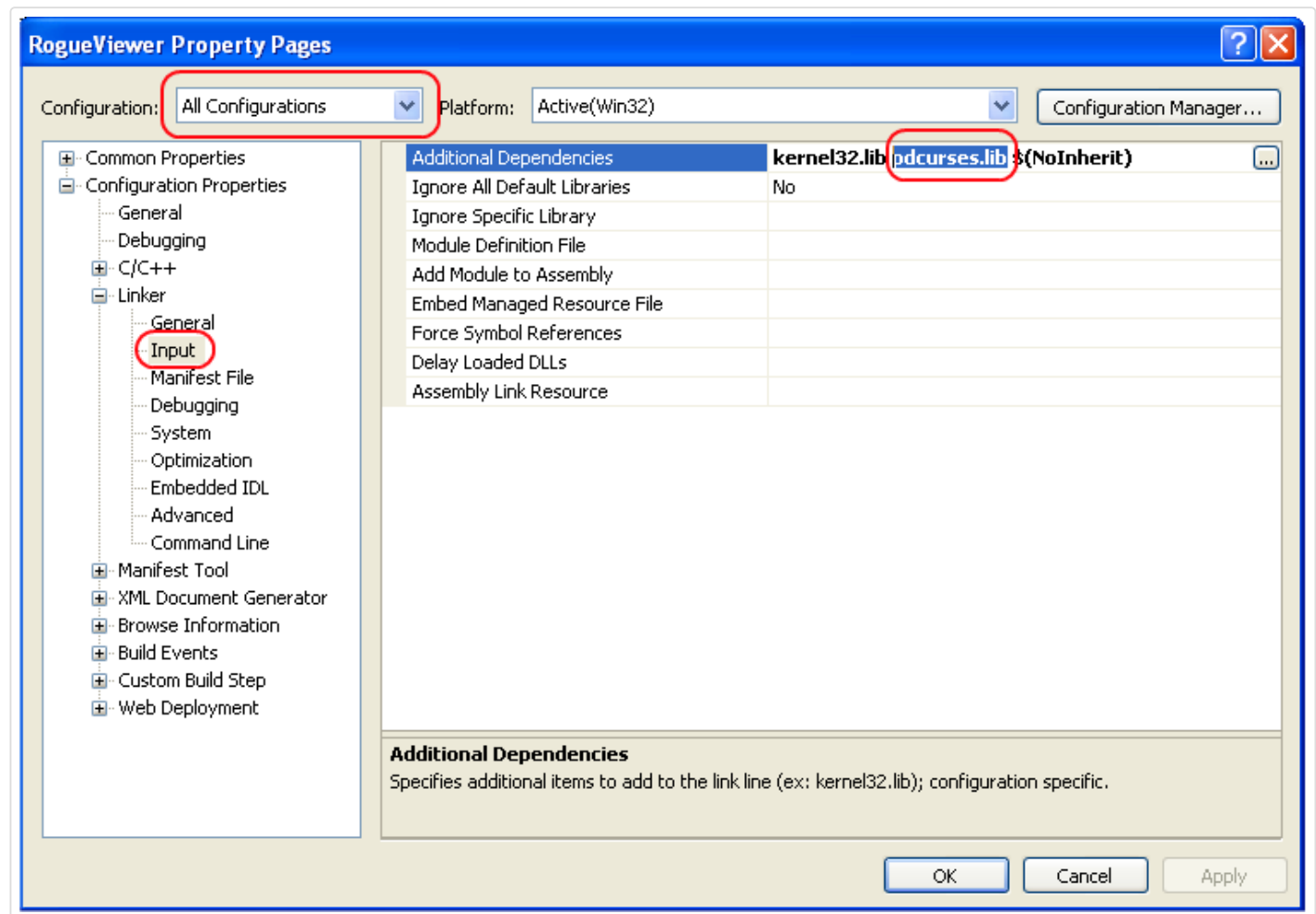
C-1) In the Solution Explorer, right click on the bolded project name and choose “Properties” from the menu.



C-2) Under the “Configuration:” dropdown, select “All Configurations”.

C-3) Open the “Configuration Properties” node, the “Linker” node, and click on “Input”.

C-4) Under “Additional Dependencies”, add the name of your library.



C-5) Click “OK”.

### 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 [A1 -- Static and dynamic libraries](#) for more information step 7.



[A.3 -- Using libraries with Code::Blocks](#)



[Index](#)



[A.1 -- Static and dynamic libraries](#)