How to install and use fltk-1.3.4 in Visual Studio 2017 [complete guide]

Blog, Principles and Practice Using C++ August 5, 2017December 16, 2017 8 Minutes
Last successfully tested: 30.11.2017

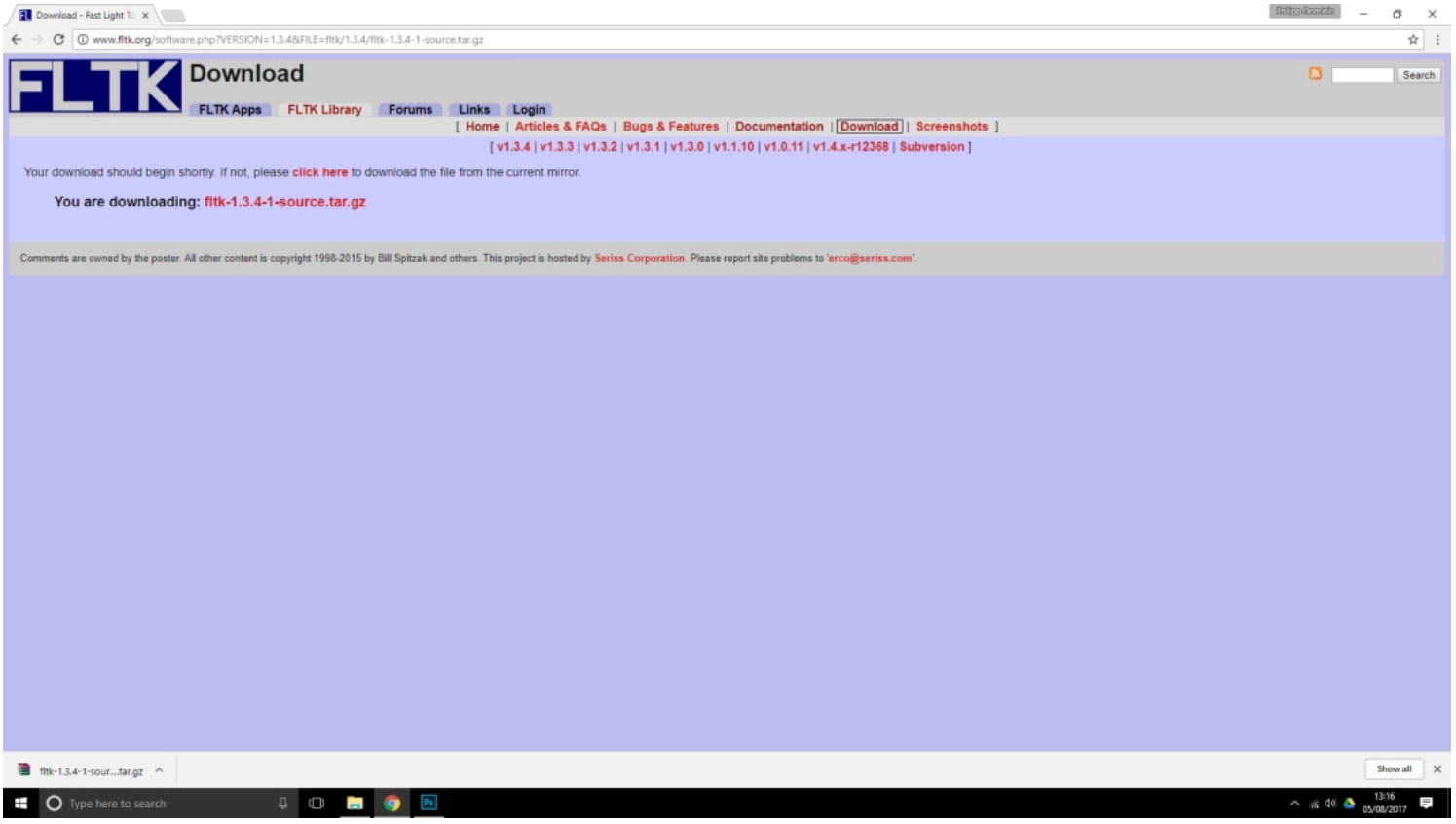
Chapter 12 to 16 of Bjarne Stroustrup's – "Programming Principles and Practice Using C++" (http://amzn.to/2y3yx9f) uses a GUI library called FLTK (Fast Light Tool Kit, "full tick"). This guide demonstrates how to download, compile from source, build and use the fltk version 1.3.4 with Visual Studio 2017 Community.
As it was quiet frustrating getting everything to run, I thought I would take the extra time to make it extra detailed, I hope it is not confusing or over complicating things for who ever is struggling with this.

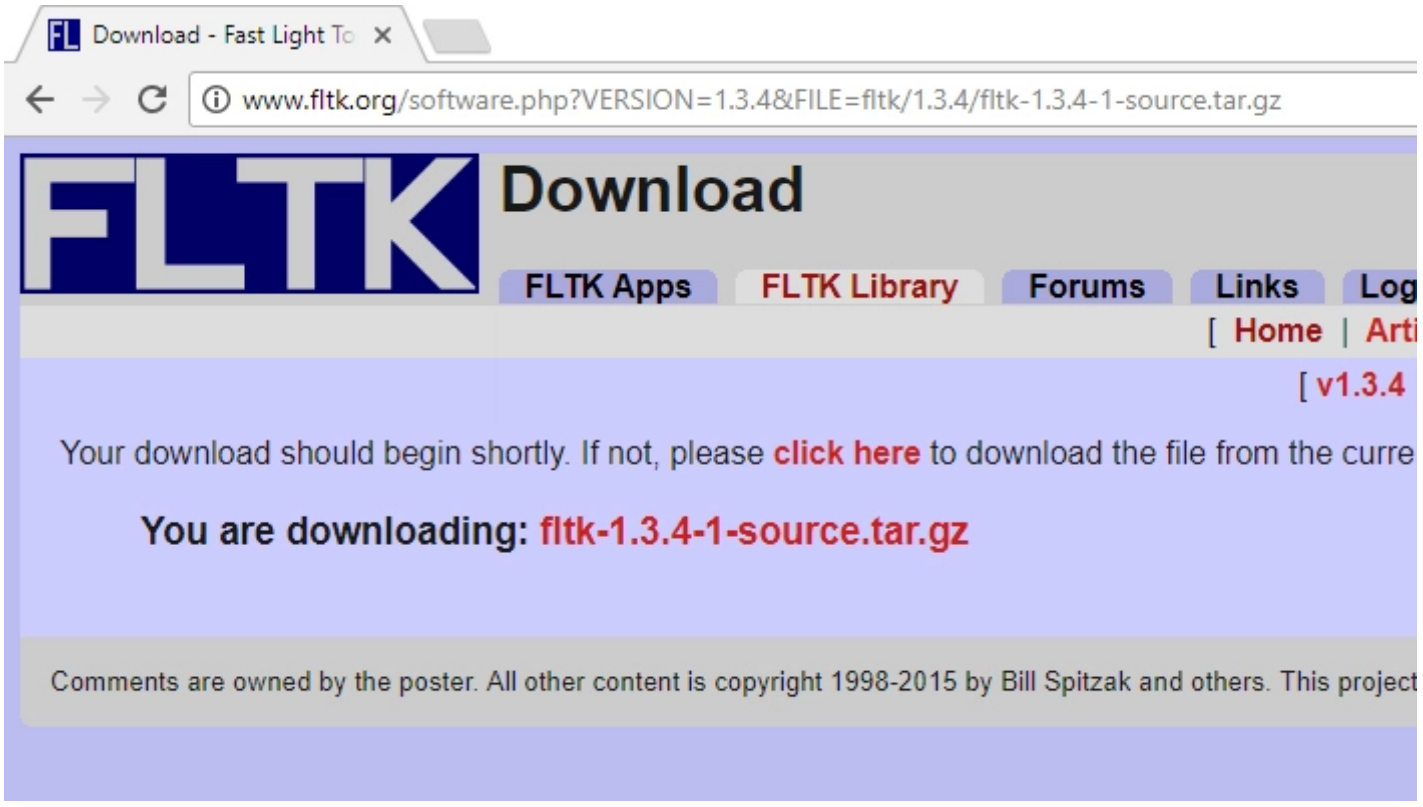
(NOTE: Before you start, even though the following path will get you to compile and run fltk projects, and it is the Bjarne Stroustrup's Programming Principles and Practice way, the fltk.org team recommended to do it like this guide (http://wp.me/p8xwHM-jw) shows. It is to prevent cross contamination and keep your Visual Studio 2017 environment clean and tidy. The original discussion can be found here (https://groups.google.com/d/msg/fltkgeneral/eLXGNQ3mndo/c0hBqxUhCQAJ). I just left this guide on my page in case people persist on doing it the Bjarne Stroustrup way.)

Step 1: Downloading fltk-1.3.4.

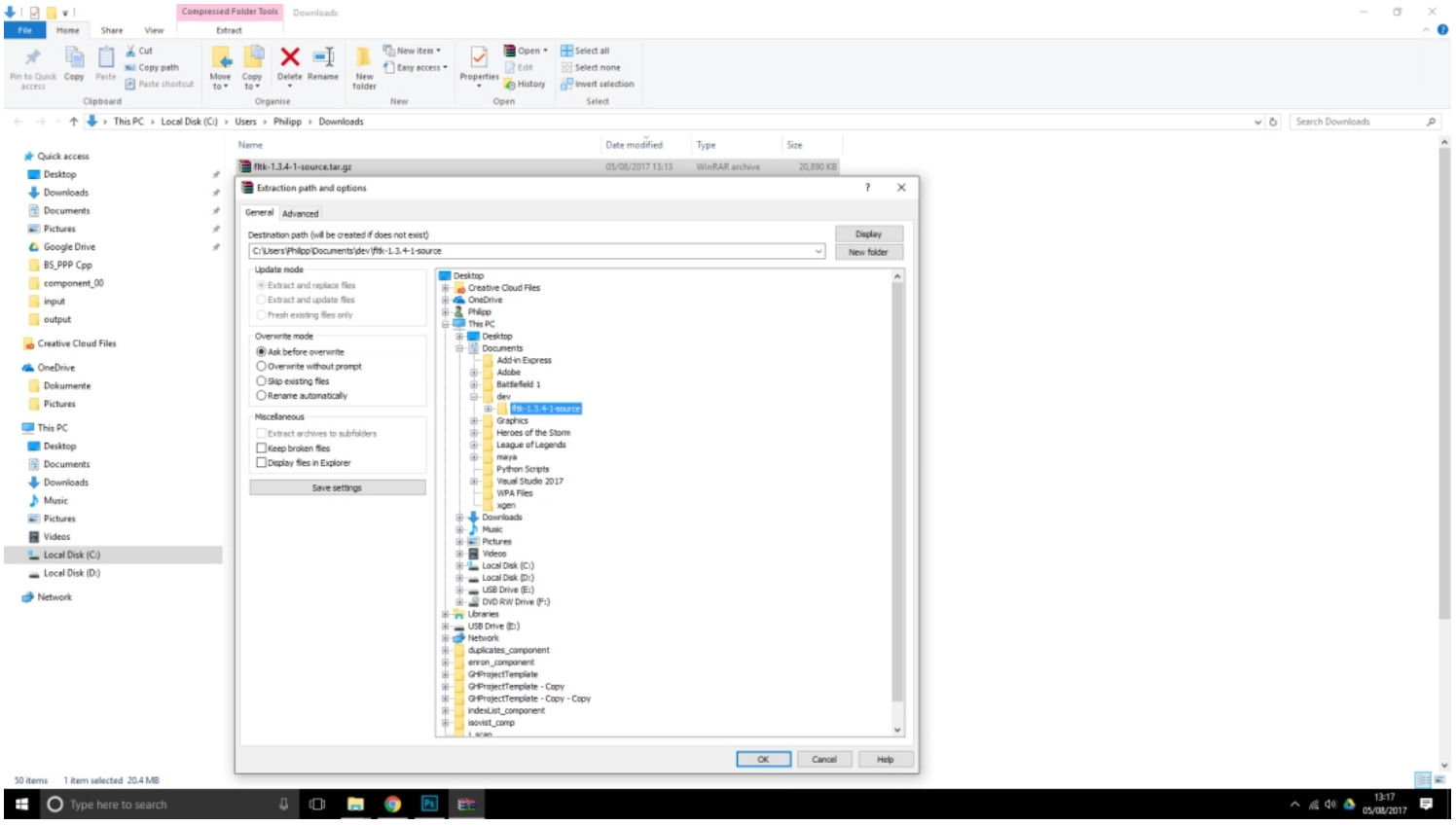
Go to: http://www.fltk.org/index.php (http://www.fltk.org/index.php)
Go to the download tab and click on **fltk-1.3.4-1-source.tar.gz**

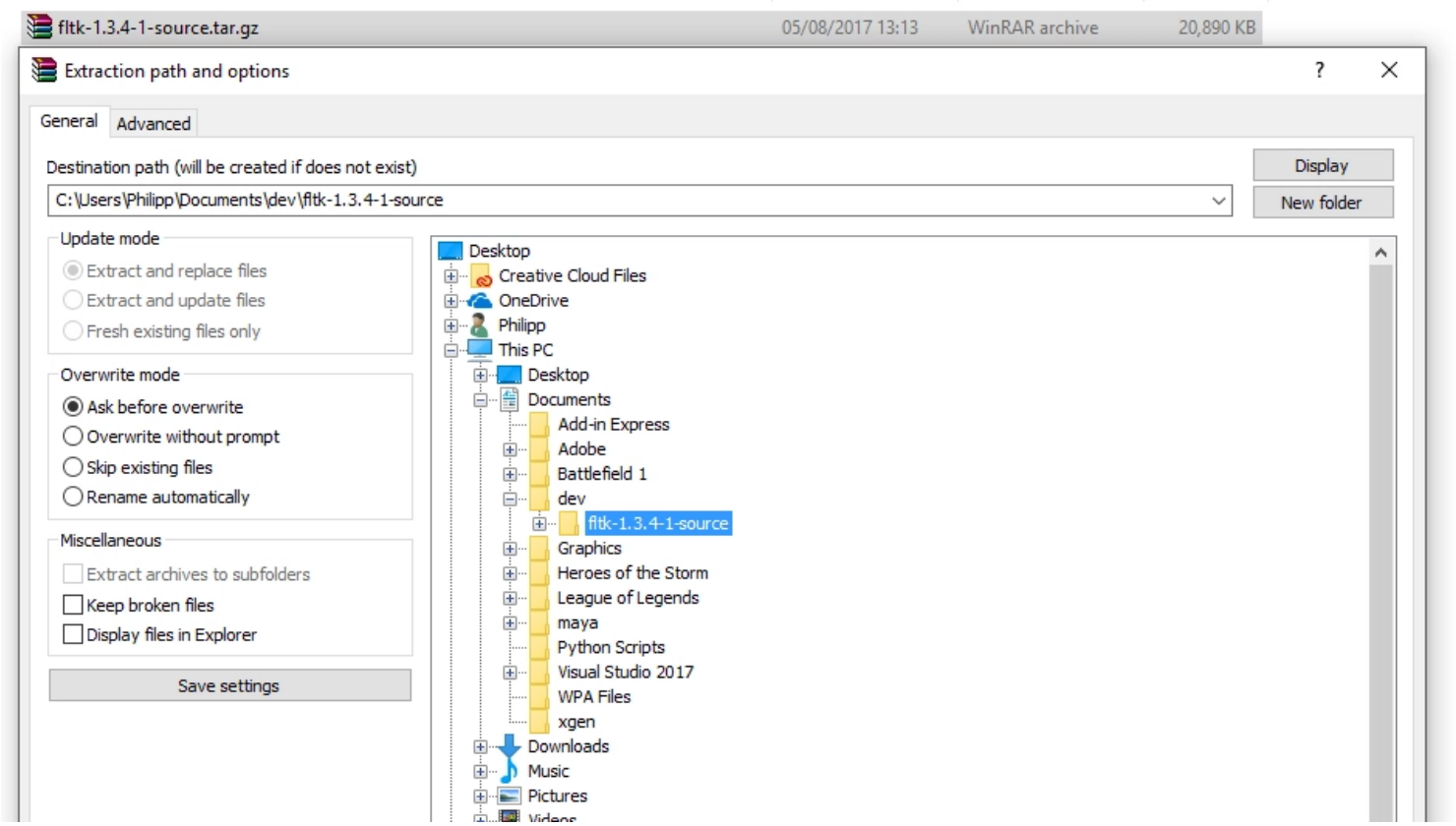
The **fltk-1.3.4-1-source.tar.gz** should be automatically downloaded.





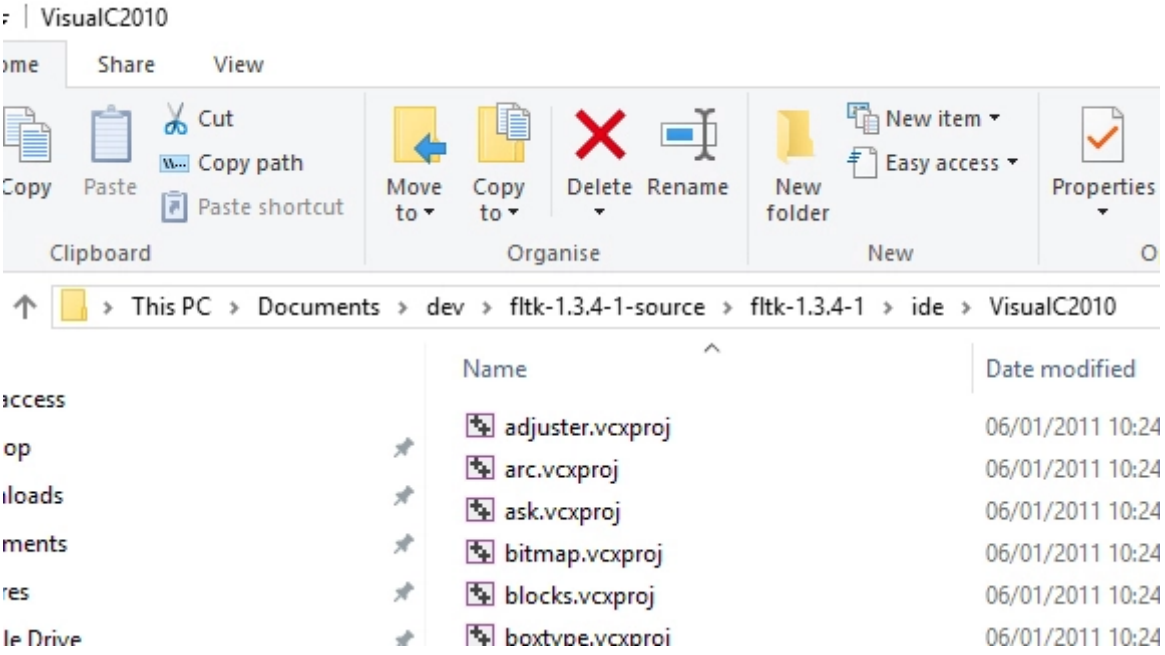
I used winRar (http://www.win-rar.com/start.html?&L=0) to extract **fltk-1.3.4-1-source.tar.gz** to my documents folder.

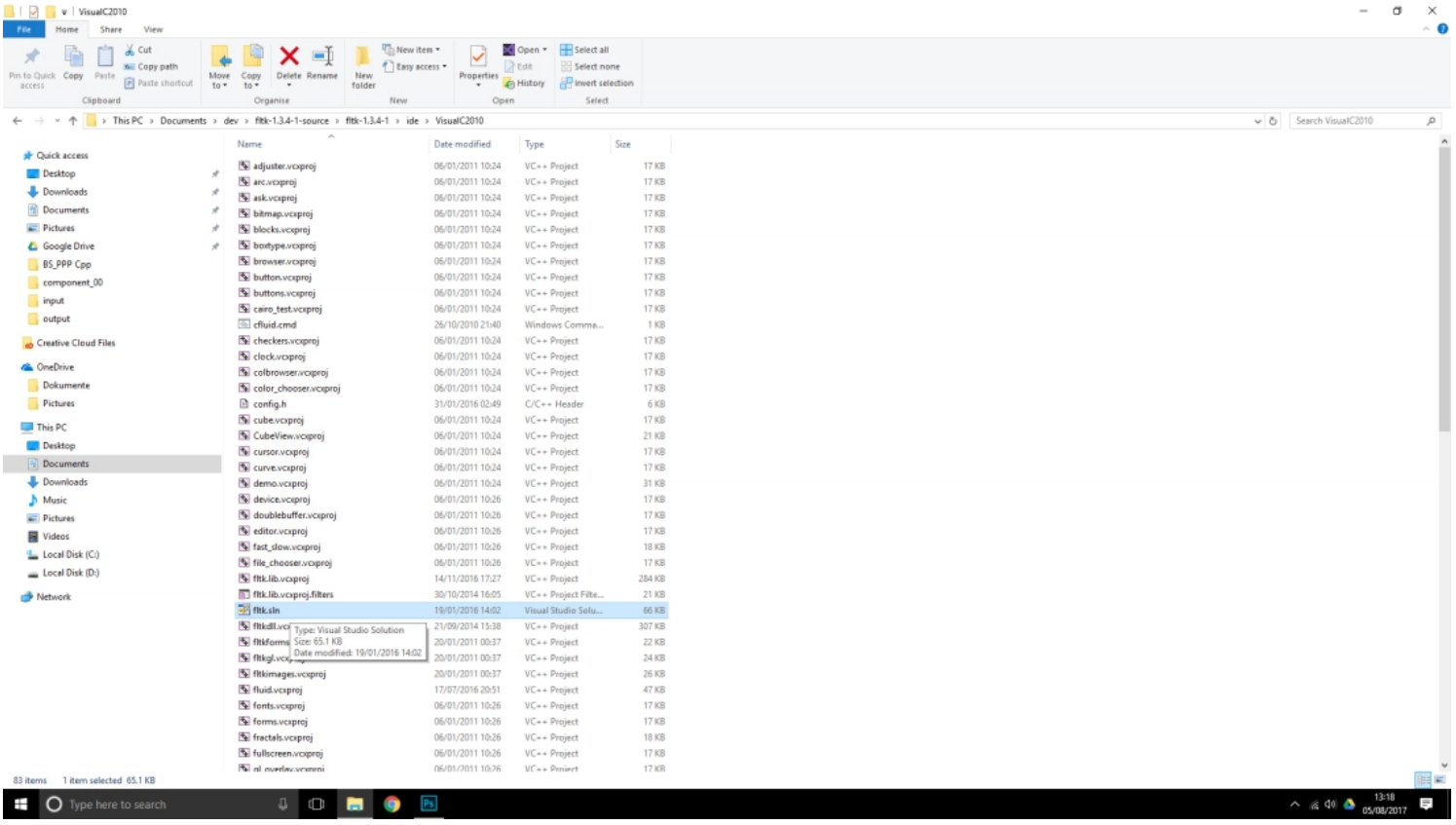




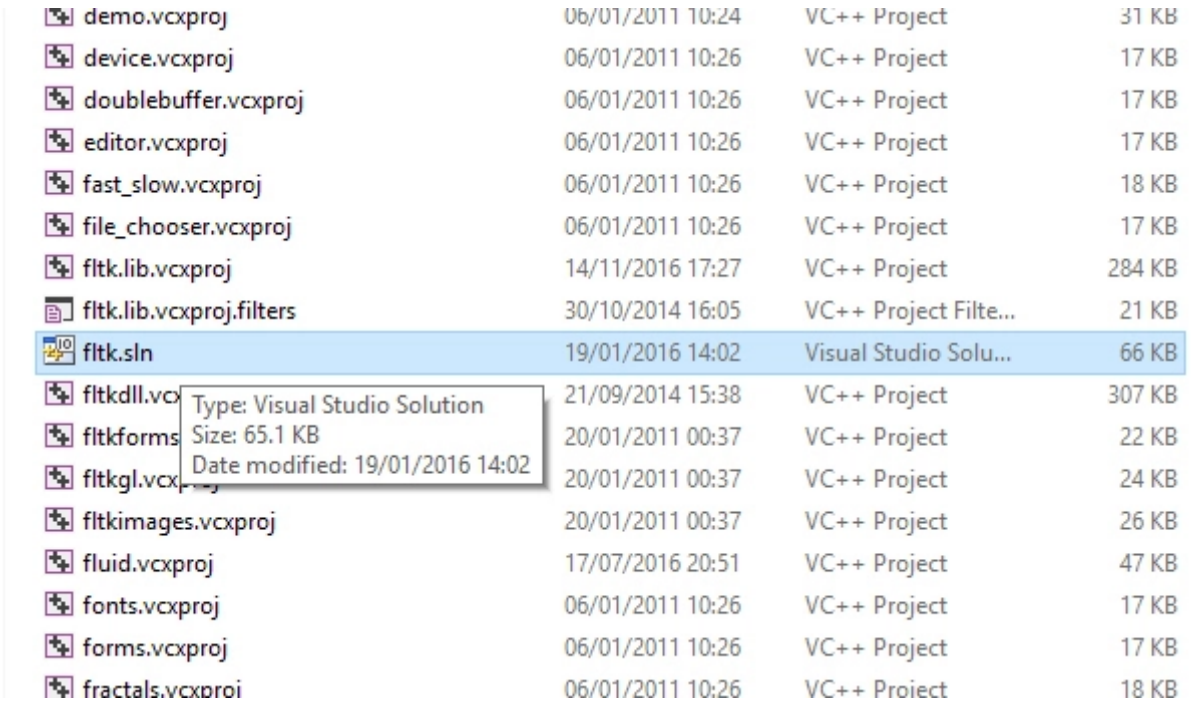
Step 2: Compiling and building fltk-1.3.4 from source.

Go to your extracted **fltk-1.3.4 folder**. In my case **C:\Users\Philipp\Documents\dev\fltk-1.3.4-1-source\fltk-1.3.4-1\ide\VisualC2010**

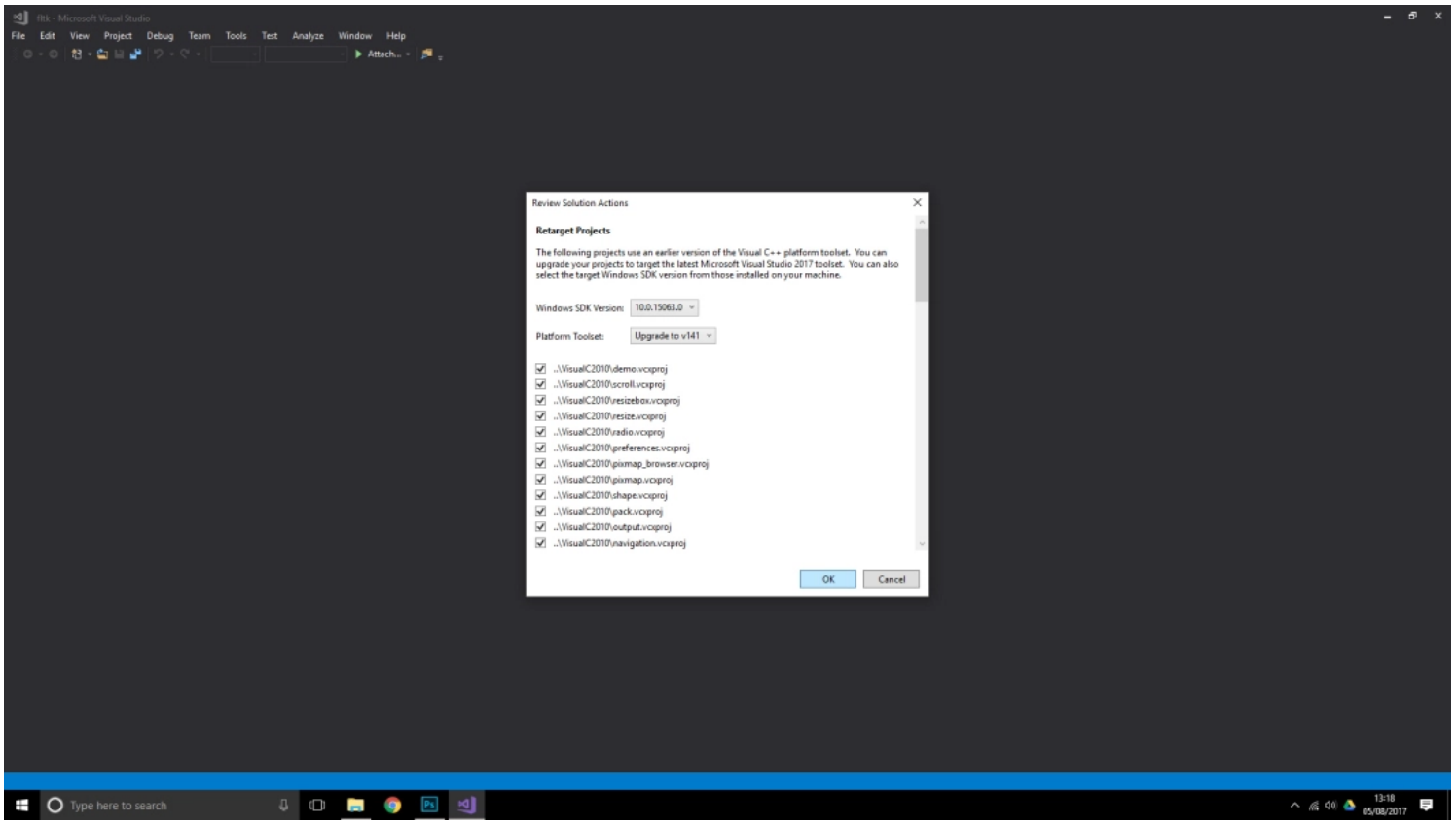


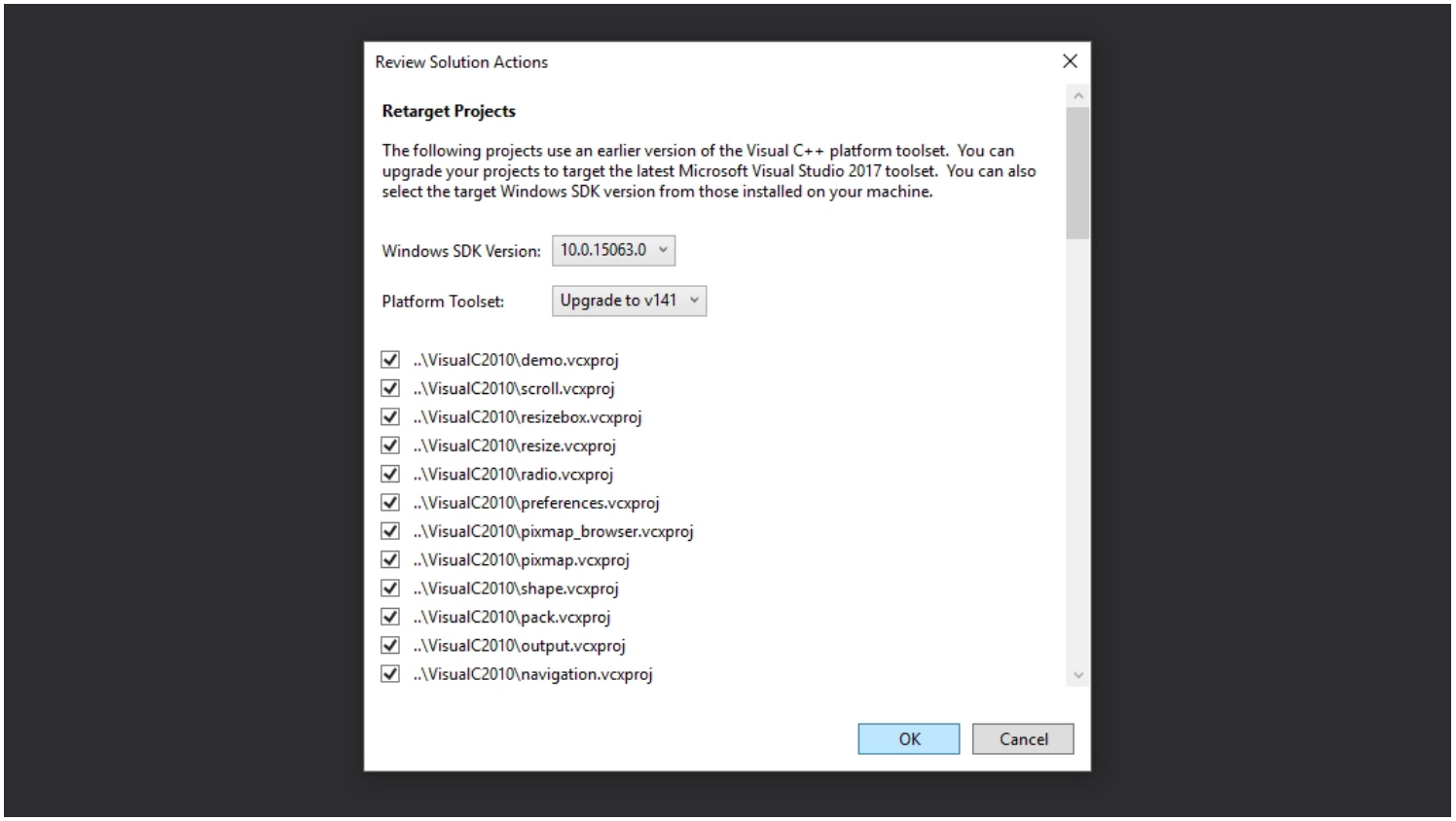


Open (double-click) **fltk.sln**.

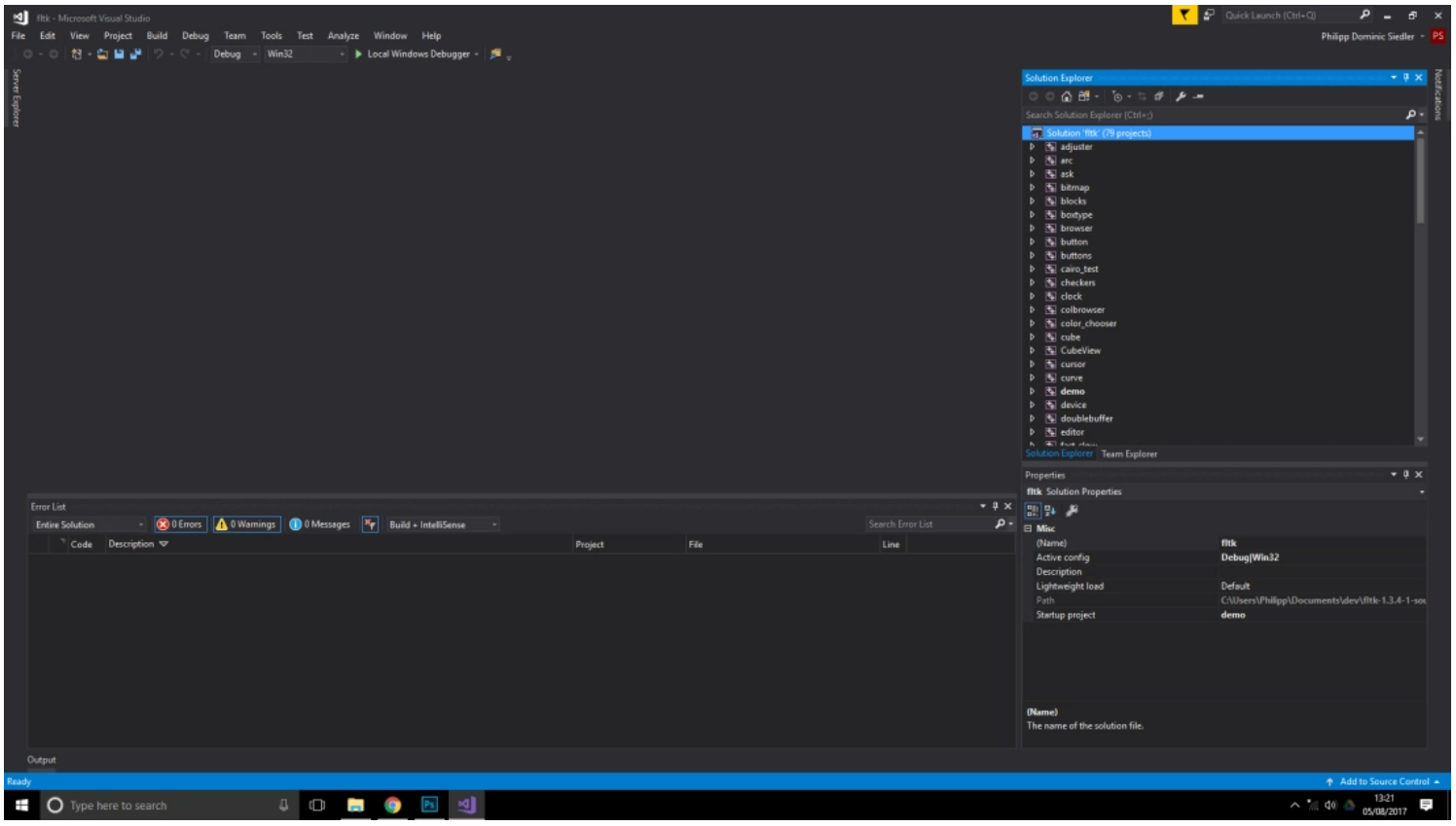


Visual Studio 2017 Community will open, and ask you if you want to update the files.

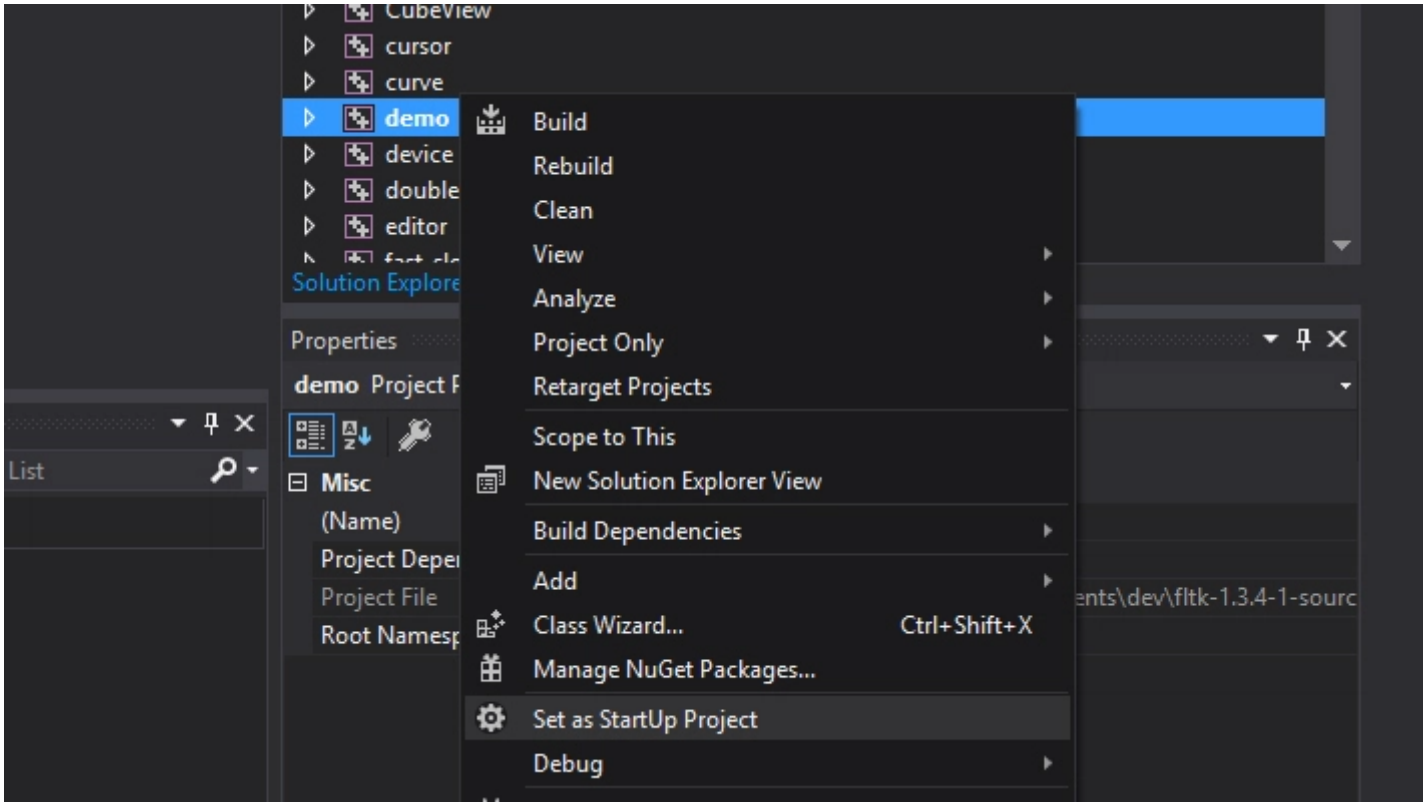




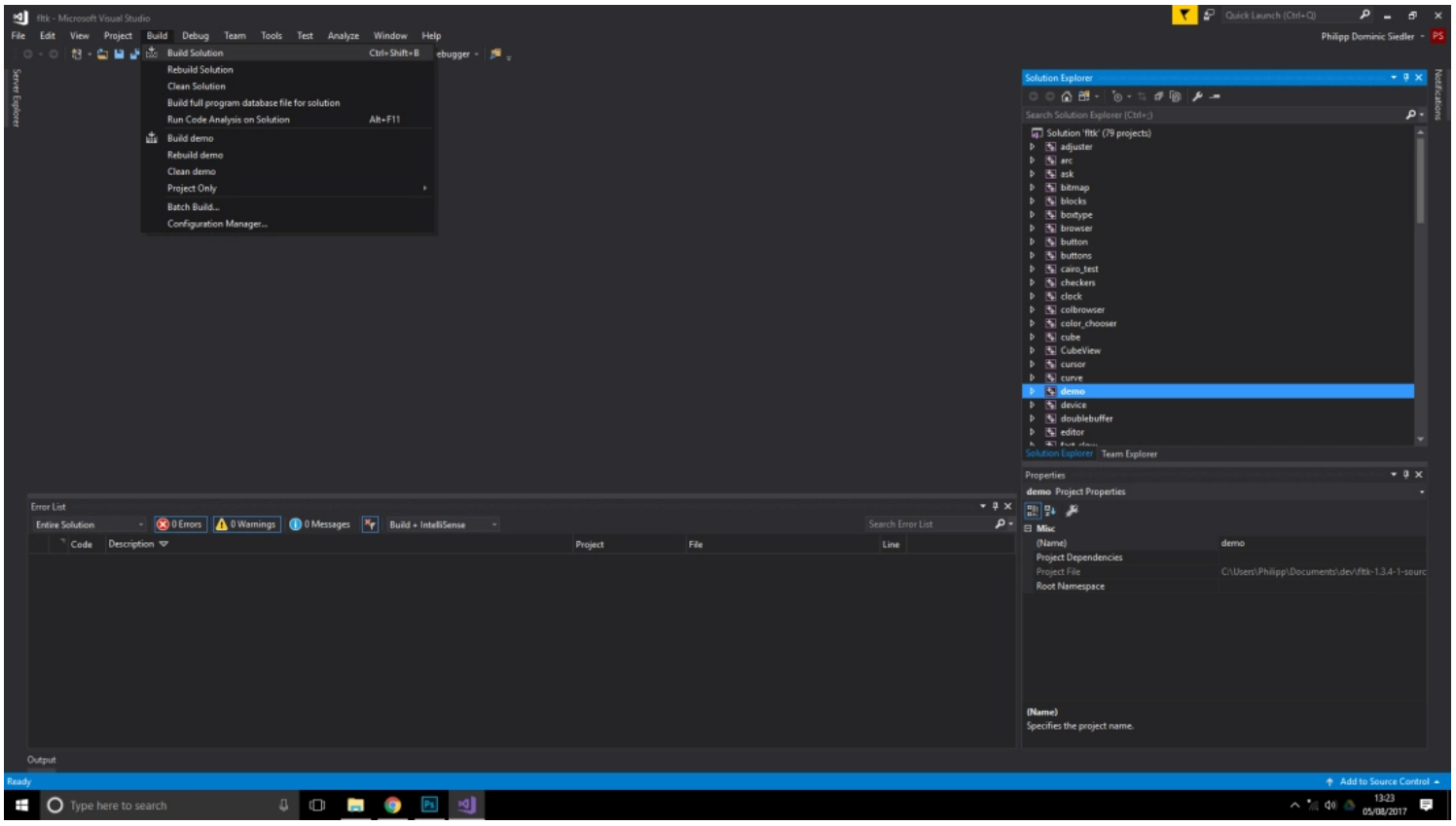
Click **OK** and wait for a few moments.
Your window will look like this:

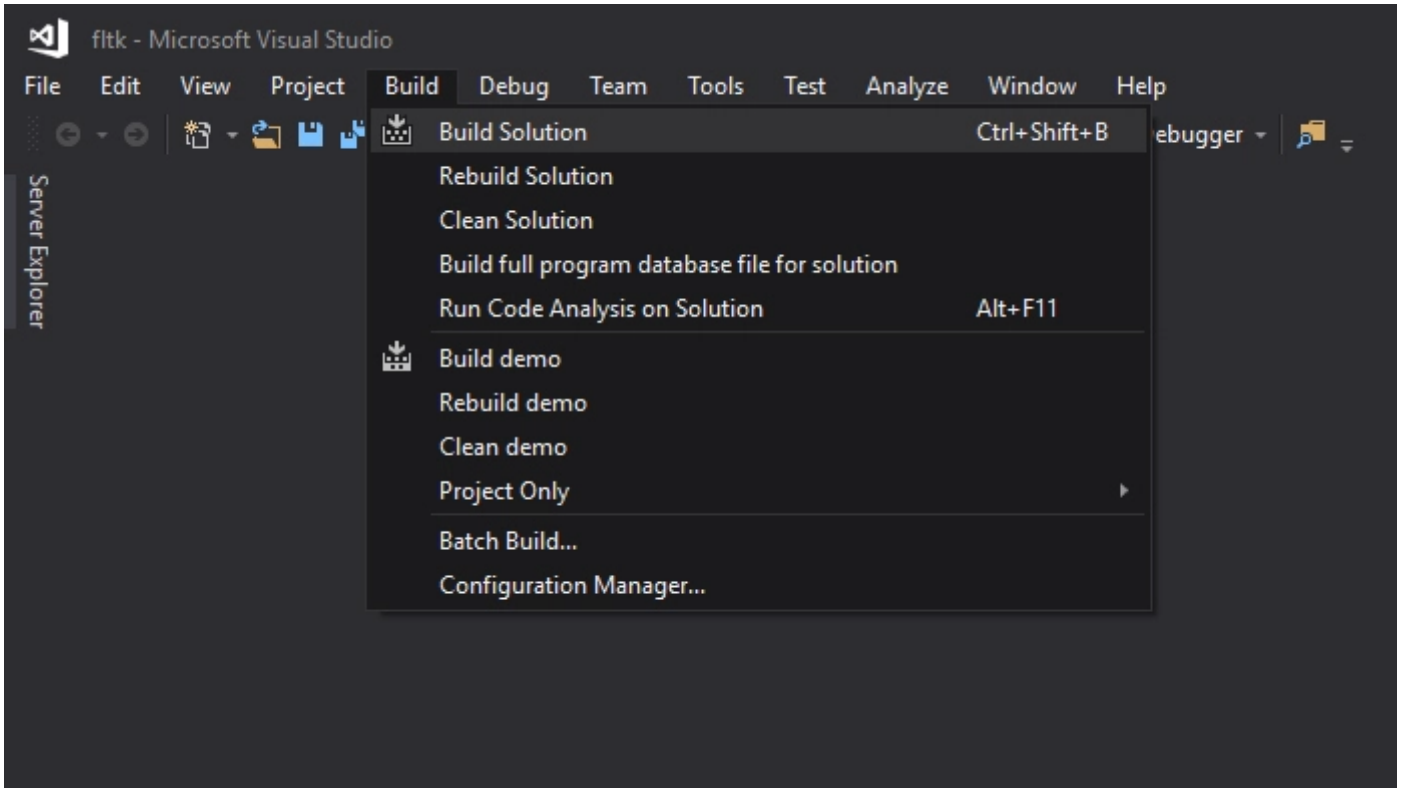


Right-click the **demo** solution file.
Click on **Set as StartUp Project**.

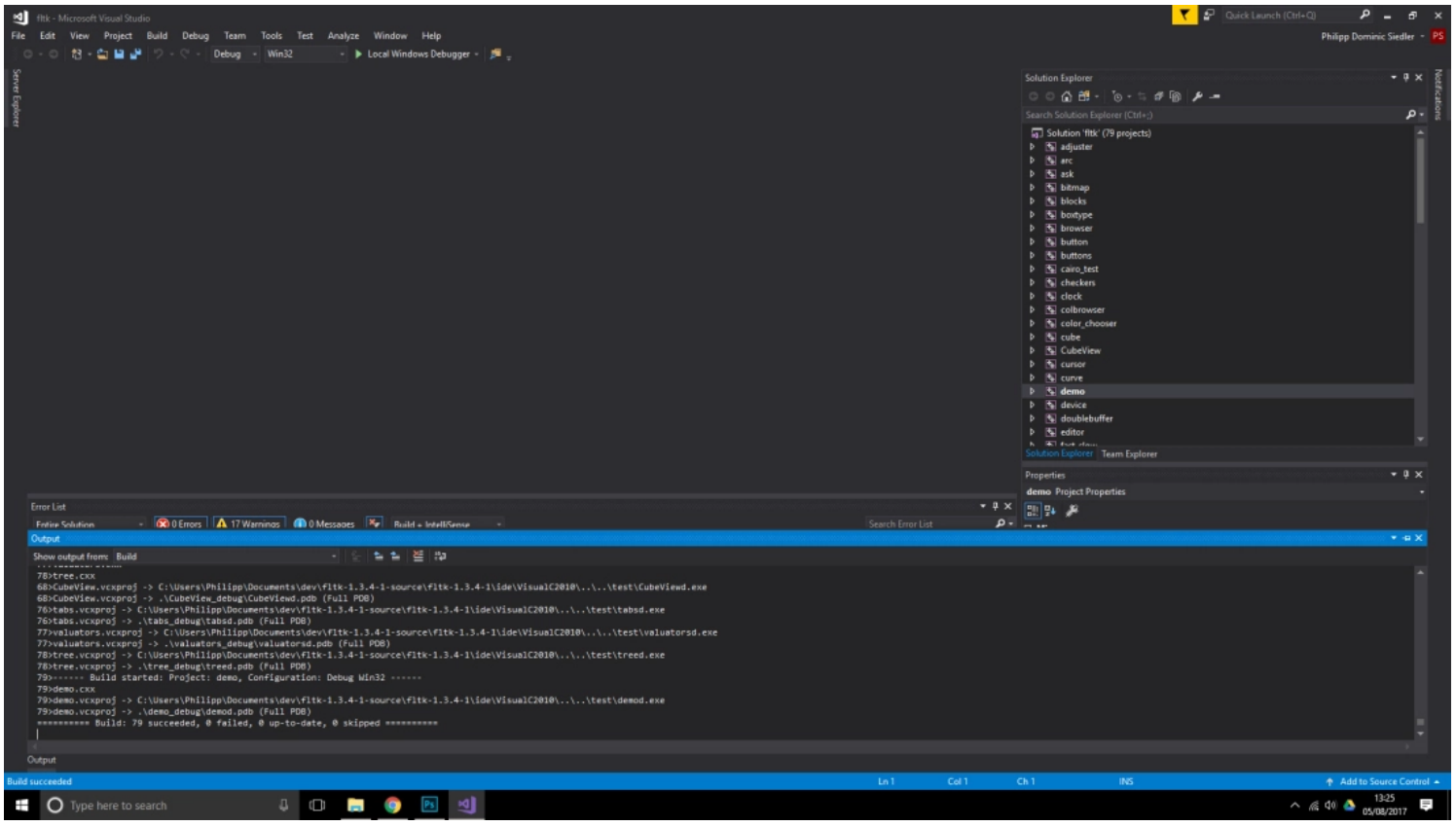


Now click on **Build** and **Build Solution**.

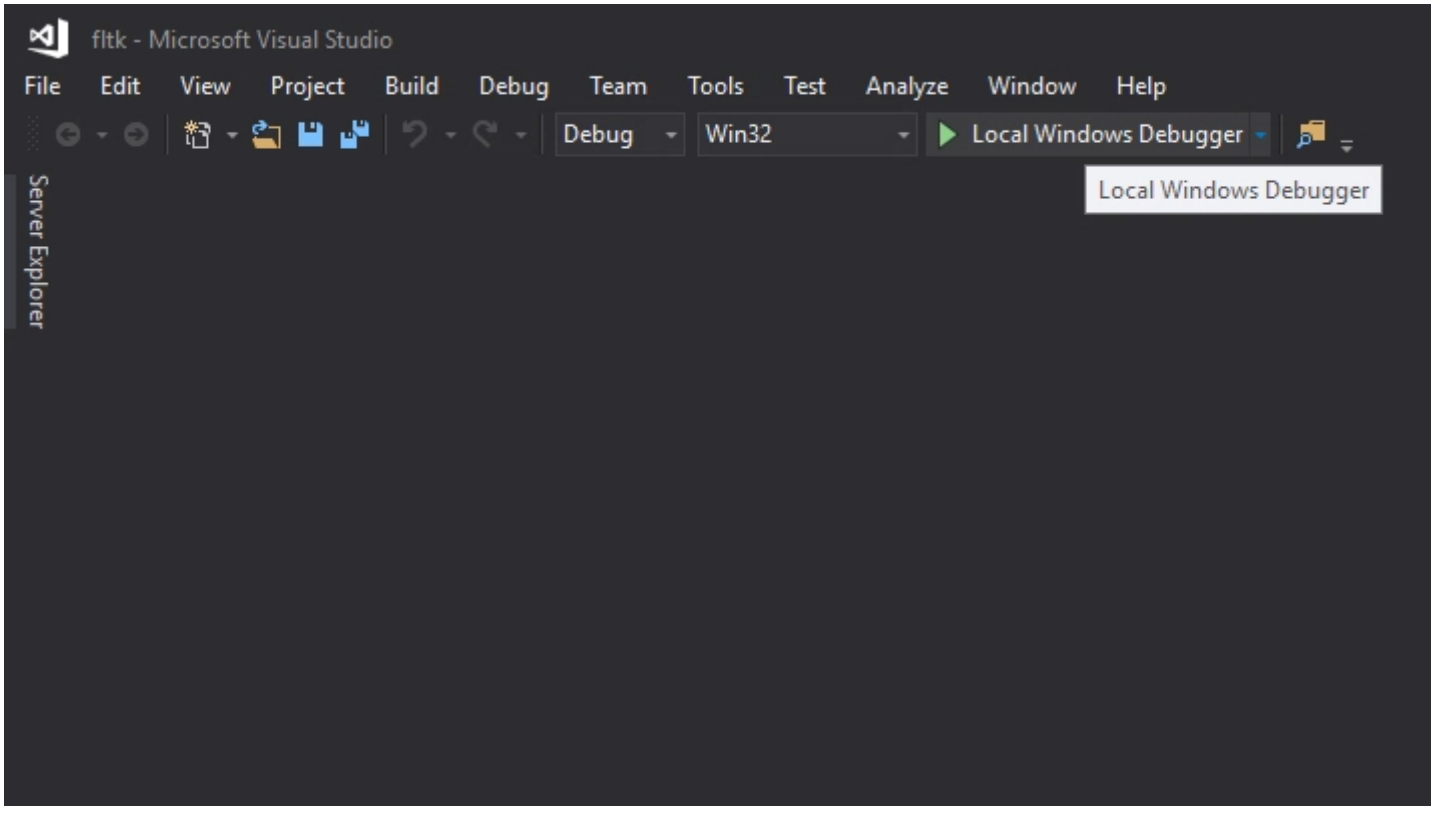




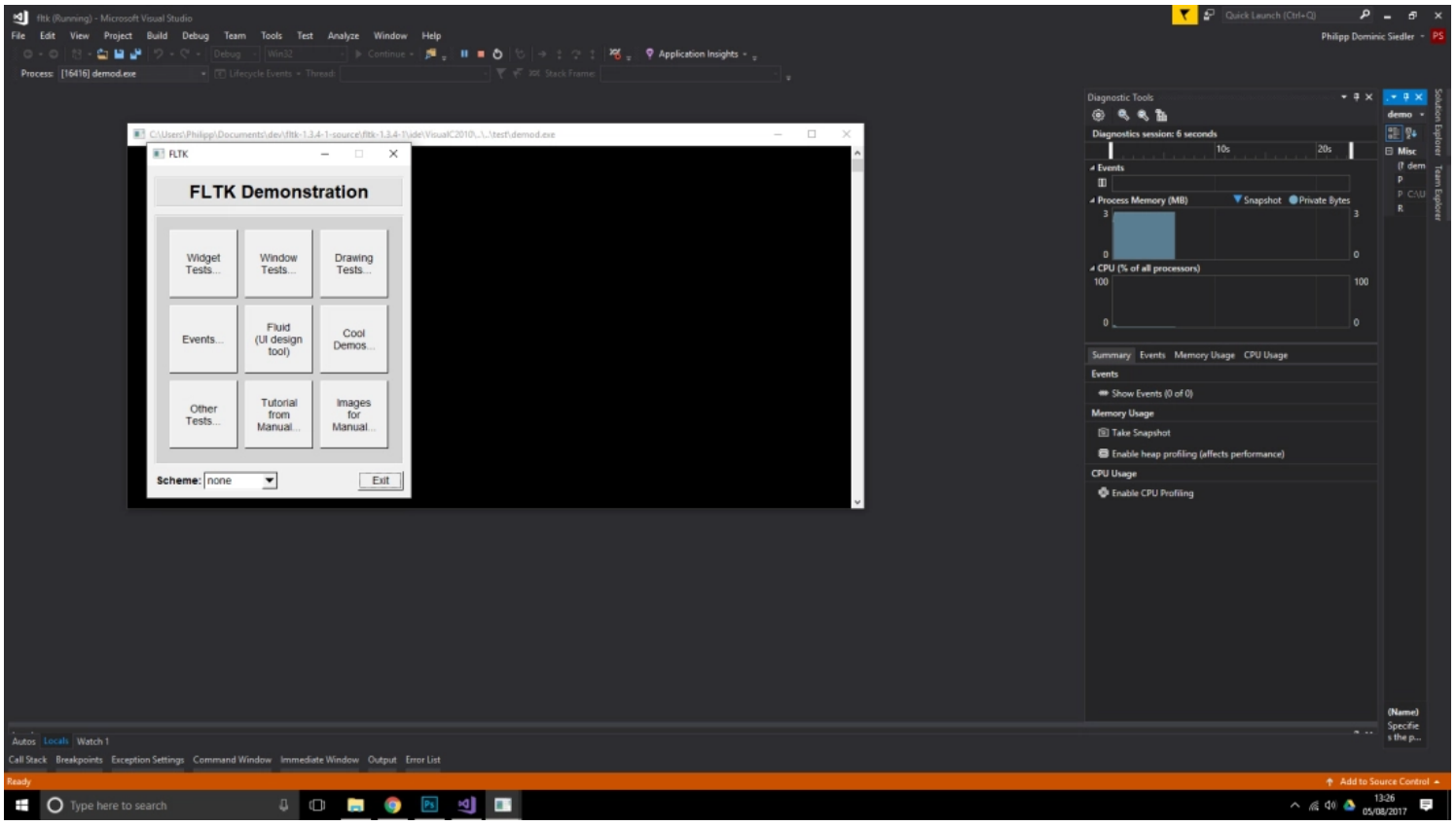
After a couple of moments your **Output** window shows **Build: 79 succeeded…**

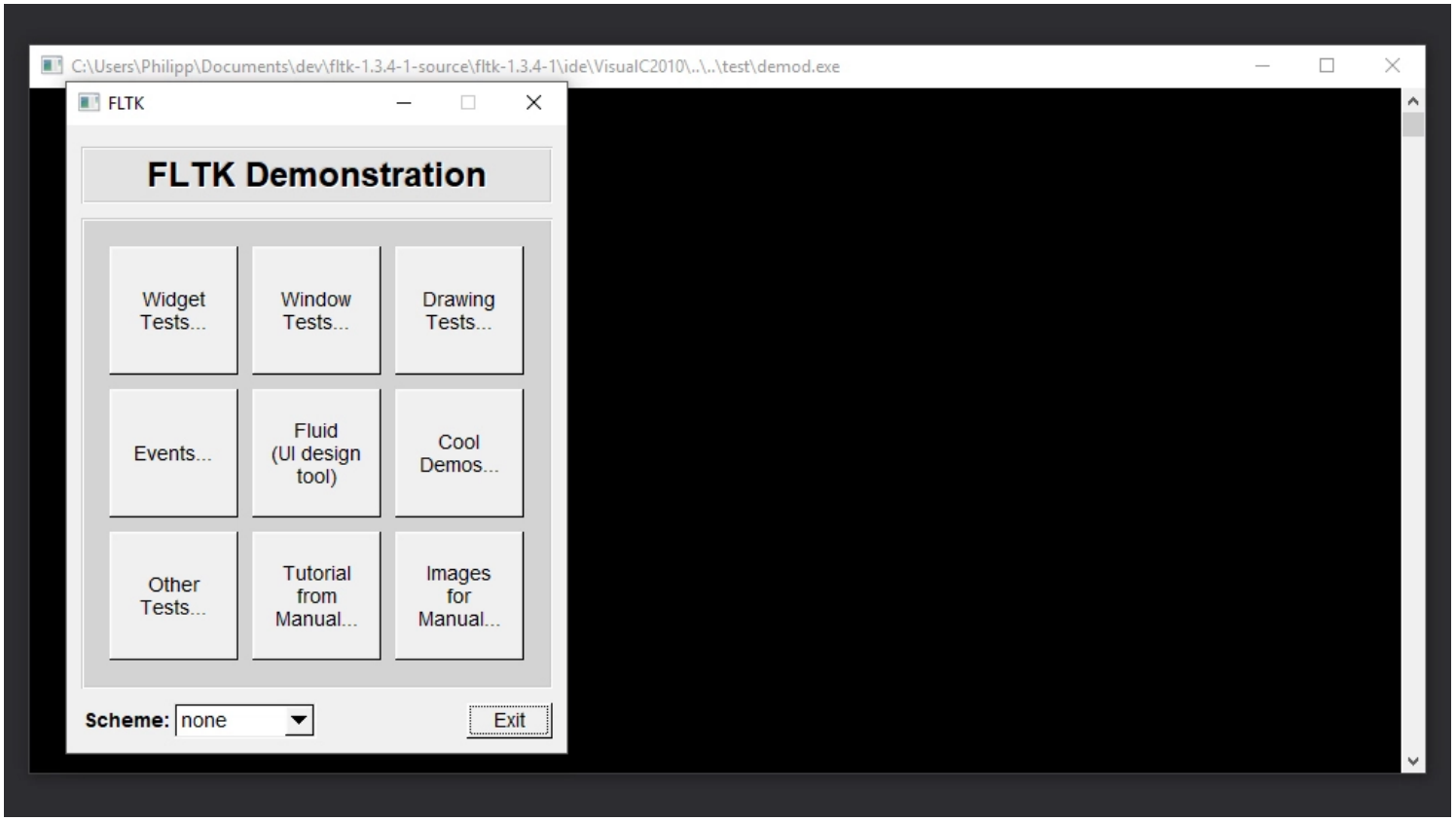


Make sure to set **Debug** mode.



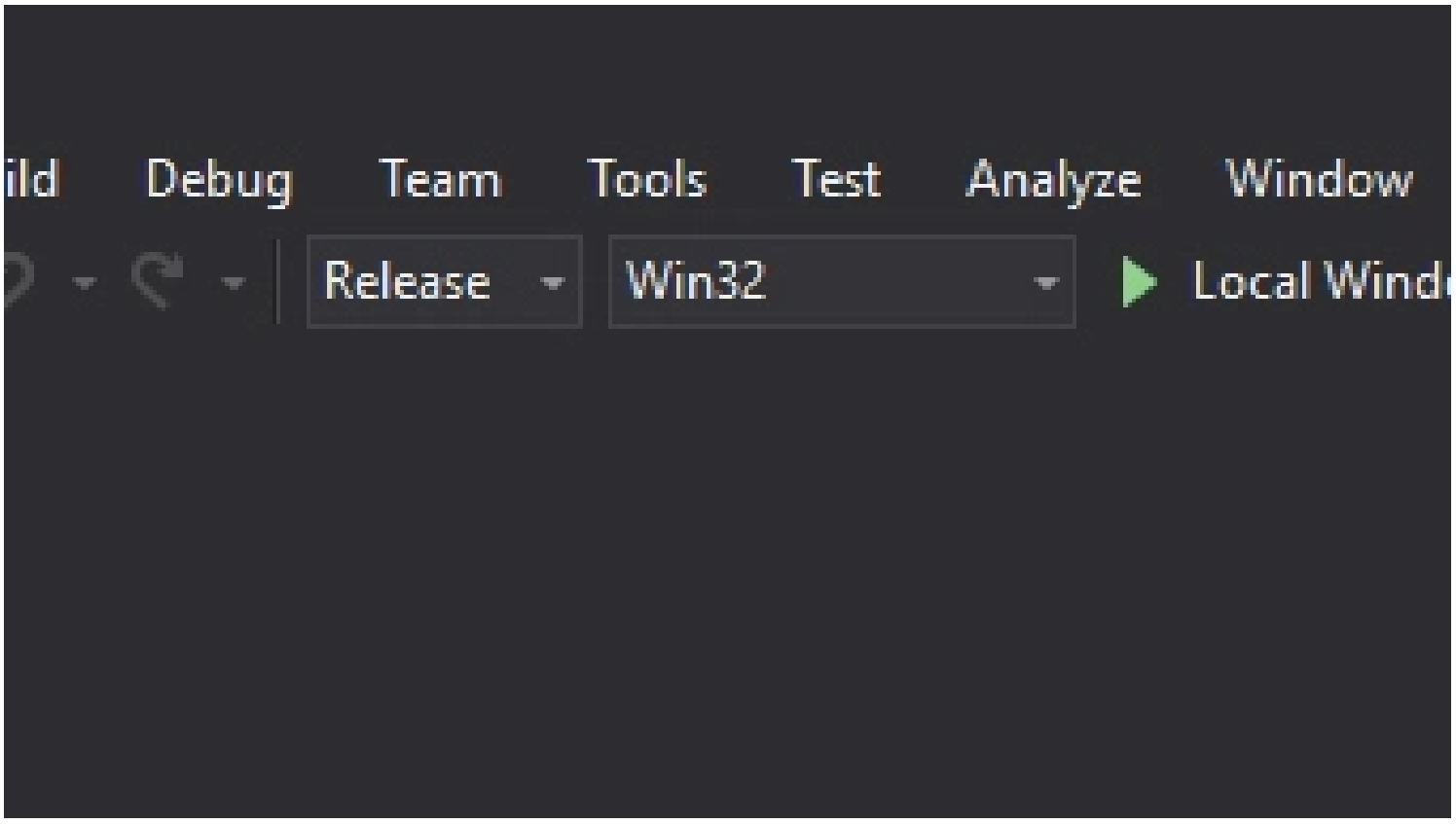
Press **F5** on your keyboard. Or the playbutton on top of your canvas **Local Windows Debugger**.



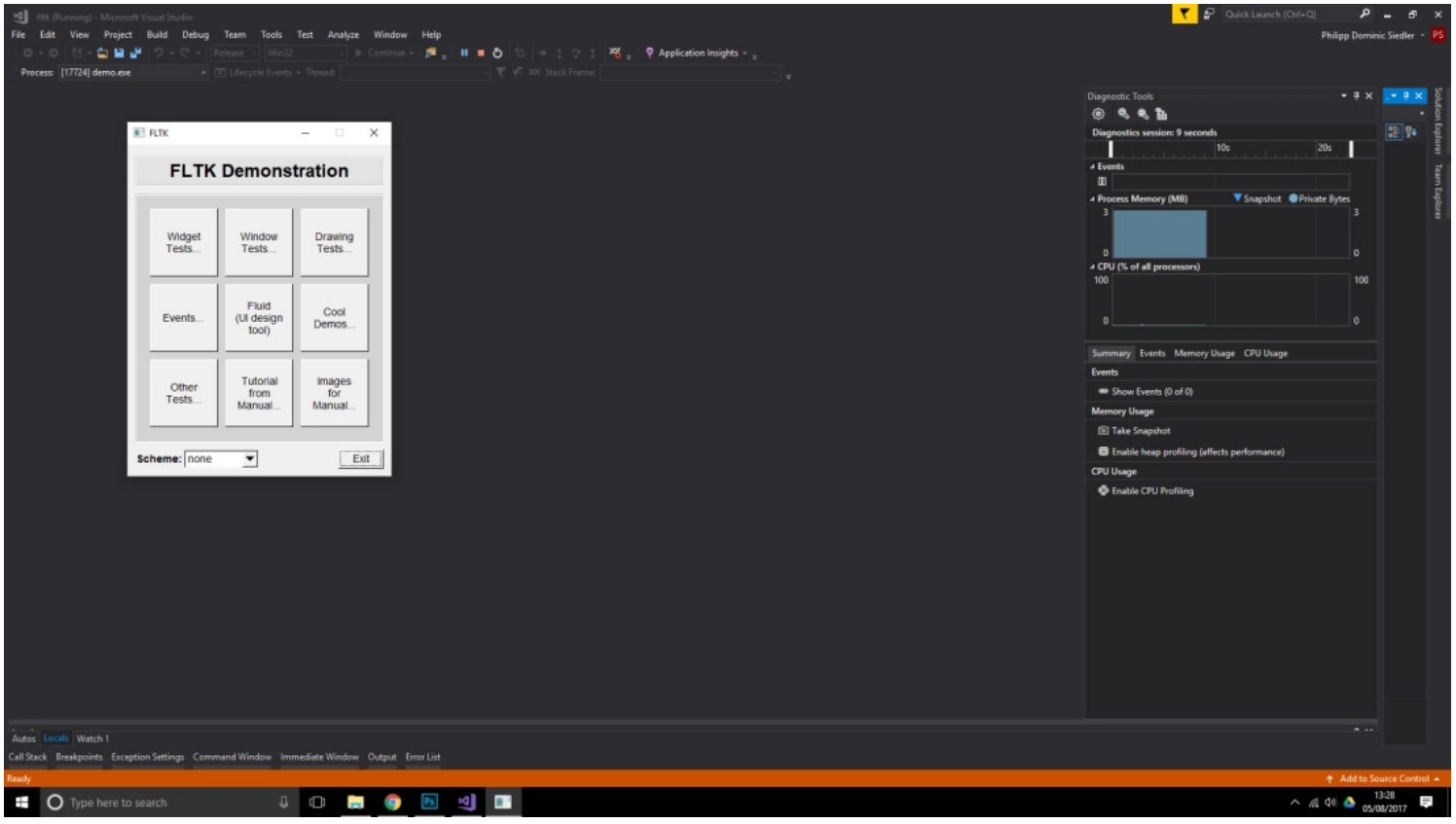


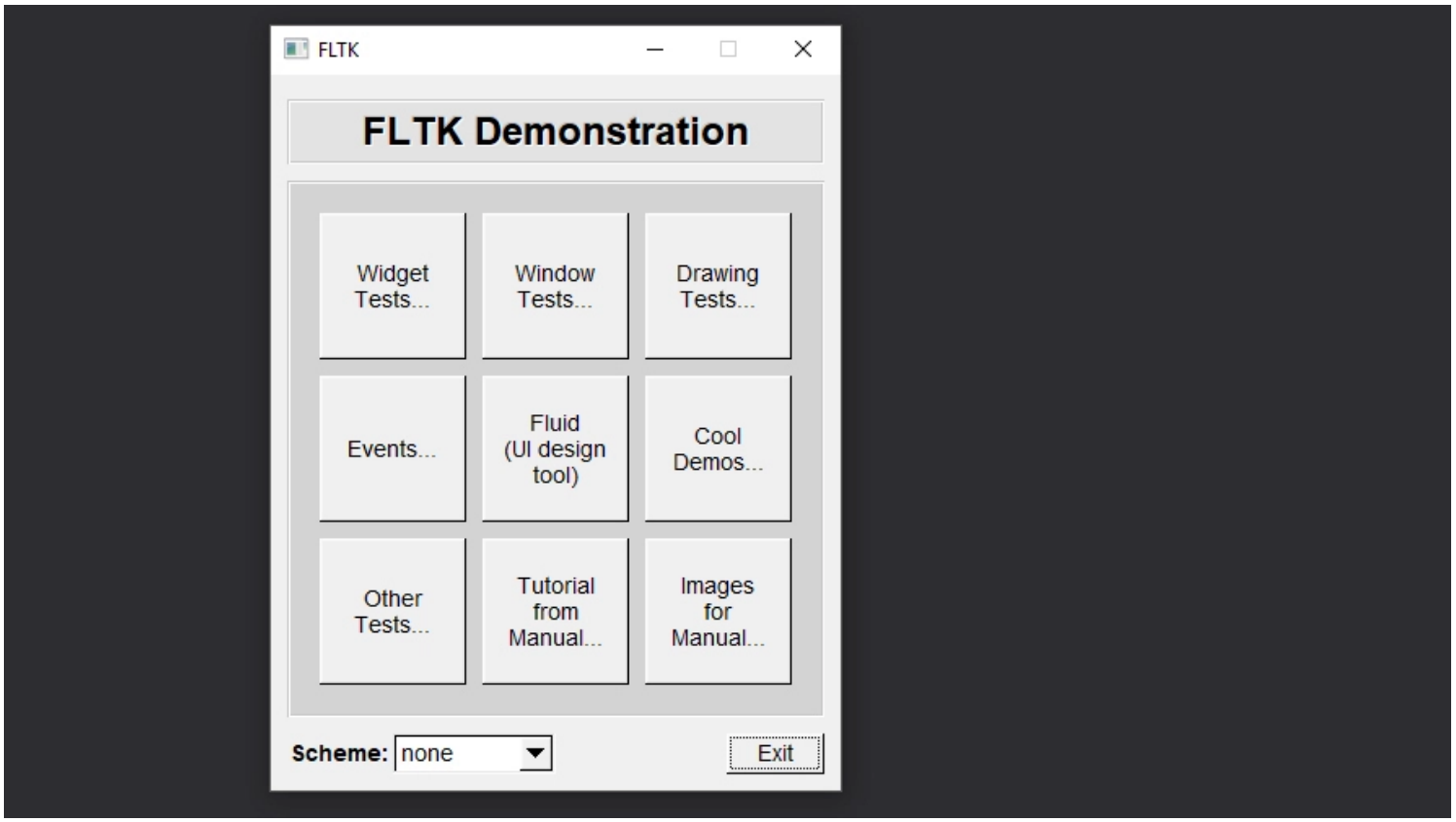
So far so good. You should get two windows, one **console window** and another one showing the **fltk-test layout**.

Now change the **Debug** mode to **Release** mode on the top left corner of your Visual Studio window.



Again: Press **F5** on your keyboard. Or the playbutton on top of your canvas **Local Windows Debugger** to run your project.
You should now see the **fltk-test layout** and no **console window**.

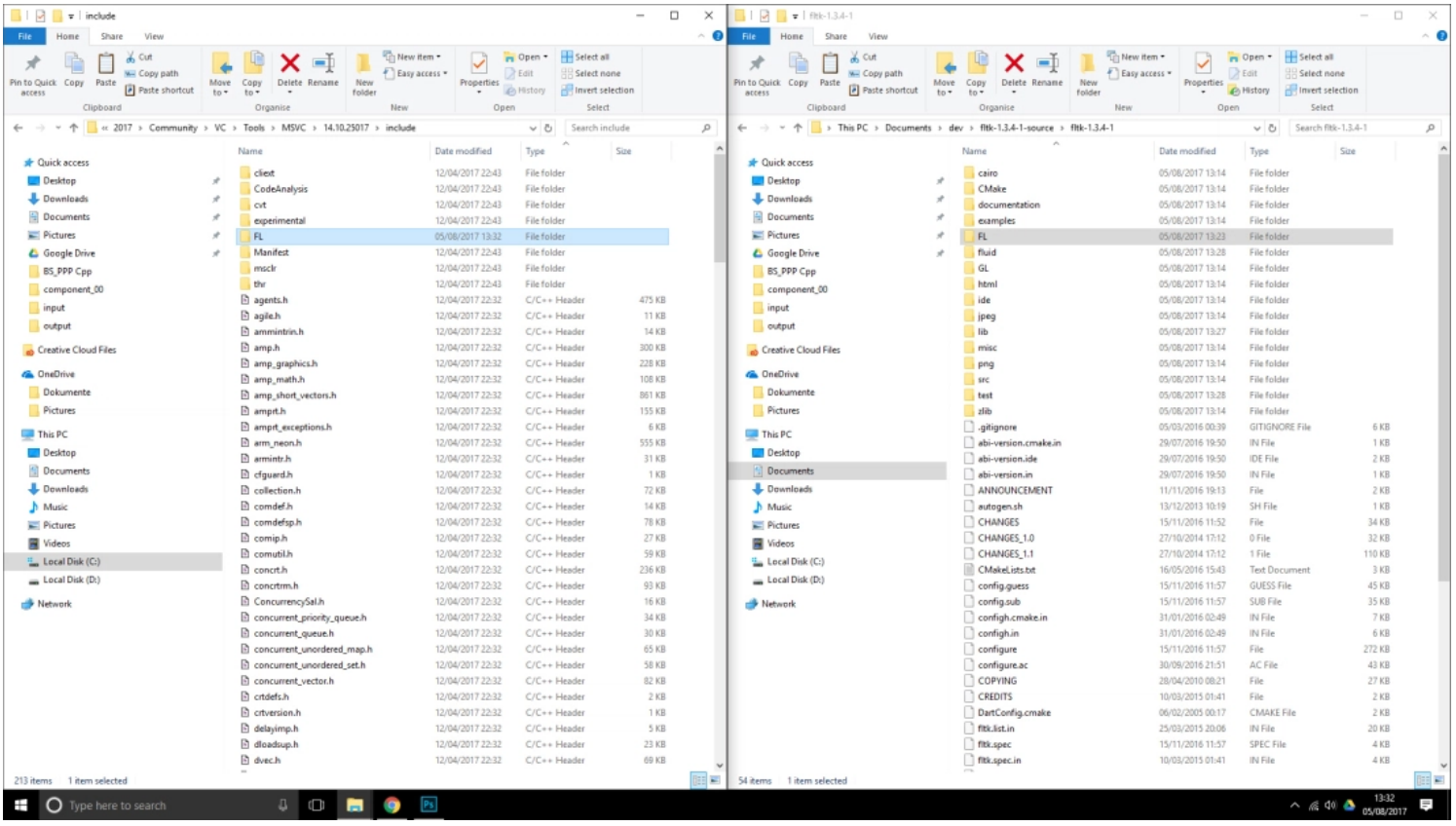


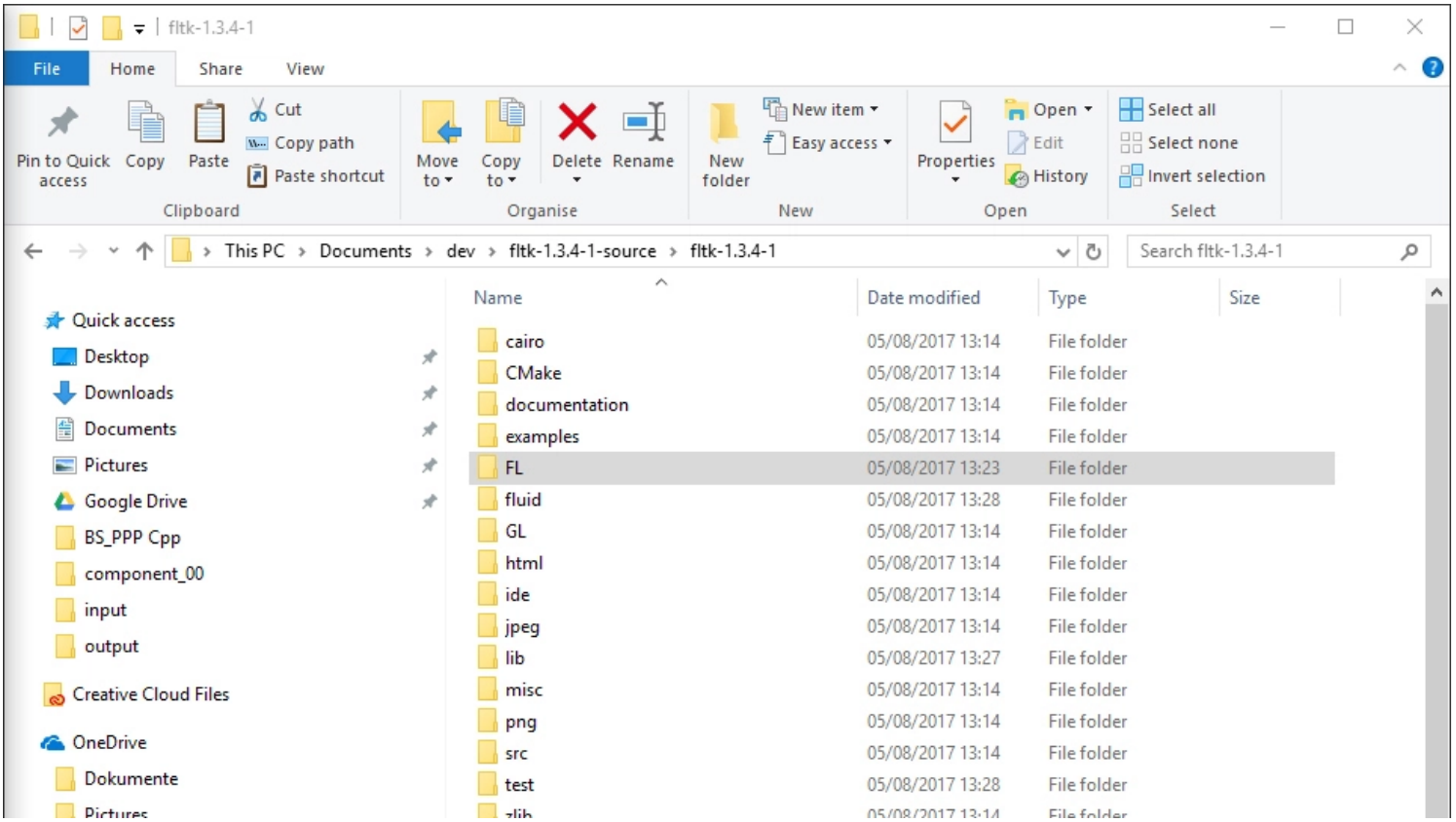


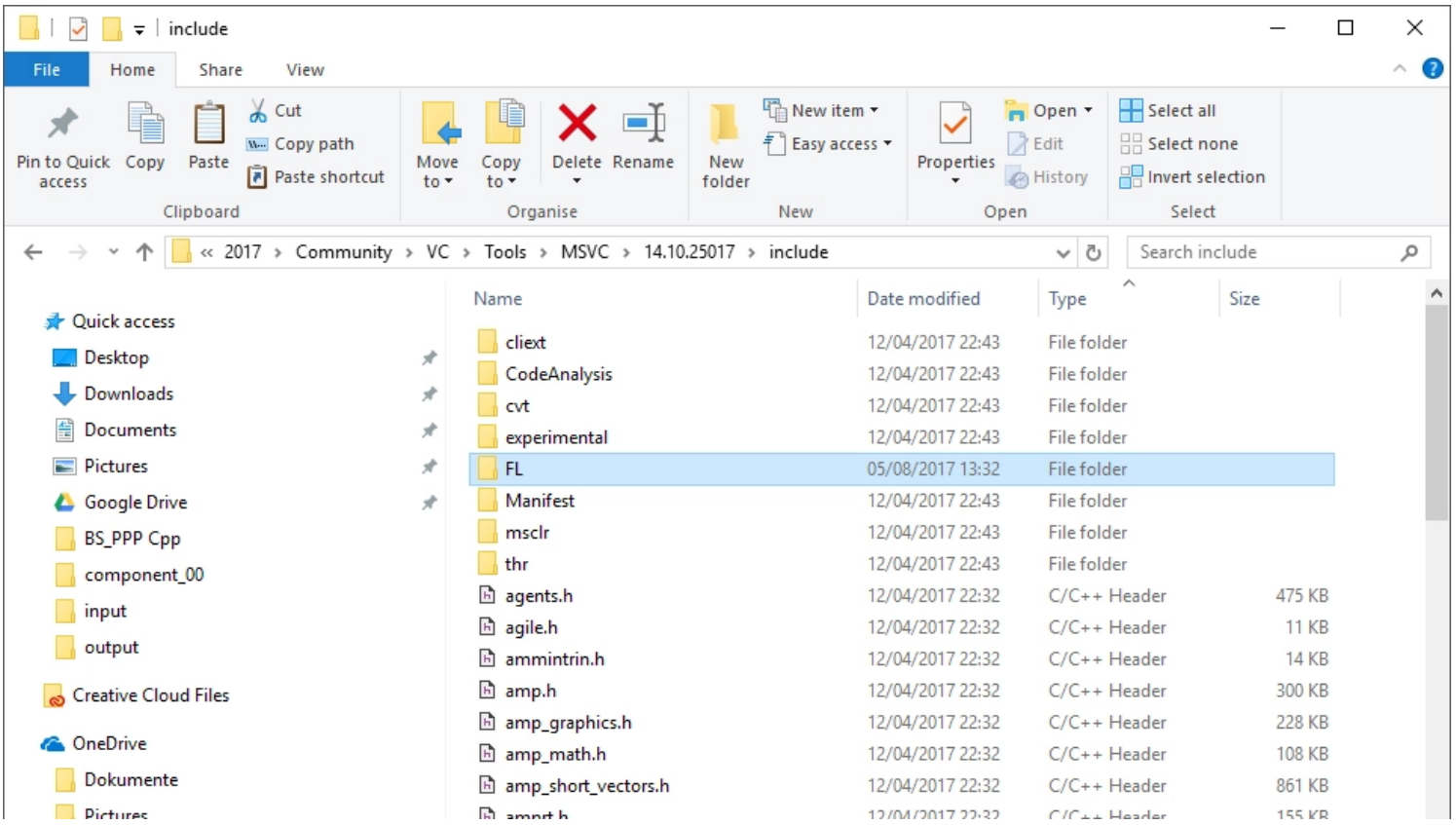
Good job. Done compiling and test-running.

Step 3: Integrating fltk-1.3.4 files into your Visual Stuido 2017 Community folder structure.
Now you need to copy those freshly compiled files (explained in three steps in the next paragraph) into your Visual Studio 2017 Community folder structure so it knows where to find them:

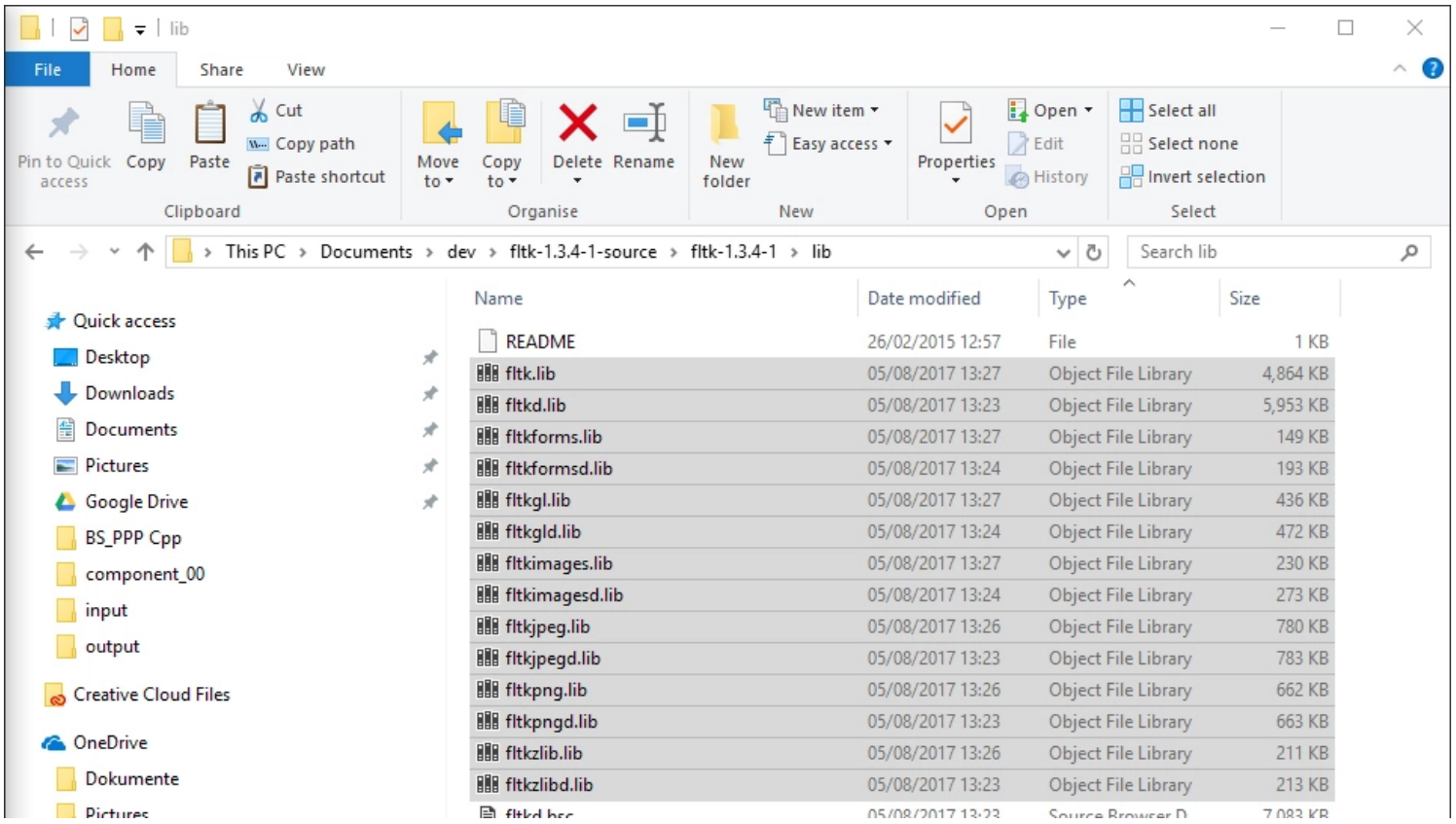
1. Go to: your **fltk-1.3.4-1** folder (in my case: **C:\Users\Philipp\Documents\dev\fltk-1.3.4-1-source\fltk-1.3.4-1**) and copy the **Fl** folder into your **include** folder (in my case): **C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\VC\Tools\MSVC\14.10.25017\include**



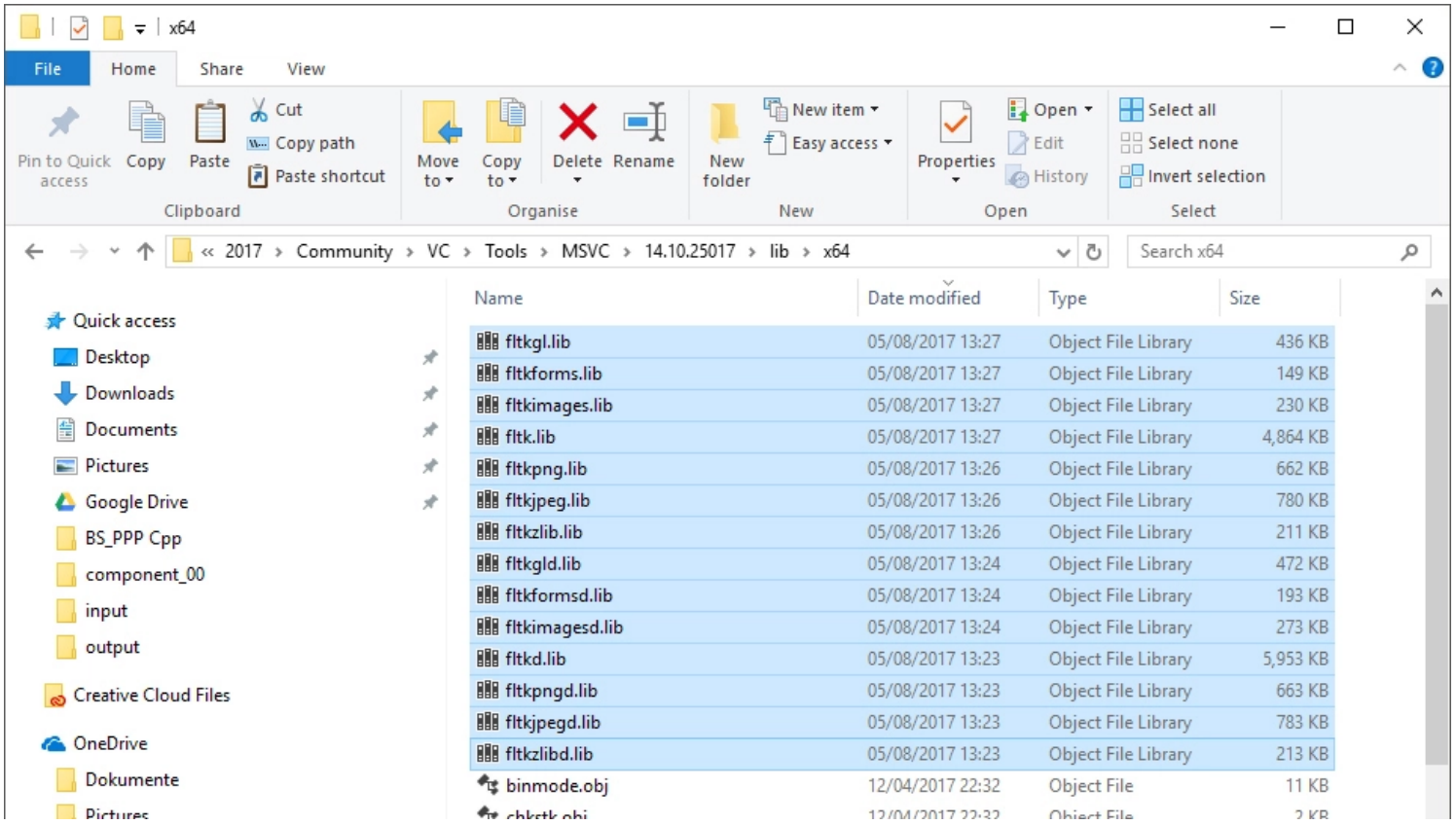

fltk-1.3.4-1
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organise
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC › Documents › dev › fltk-1.3.4-1-source › fltk-1.3.4-1
Search fltk-1.3.4-1
Quick access
Desktop
Downloads
Documents
Pictures
Google Drive
BS_PPP Cpp
component_00
input
output
Creative Cloud Files
OneDrive
Dokumente
Pictures
Name
Date modified
Type
Size
cairo 05/08/2017 13:14 File folder
CMake 05/08/2017 13:14 File folder
documentation 05/08/2017 13:14 File folder
examples 05/08/2017 13:14 File folder
FL 05/08/2017 13:23 File folder
fluid 05/08/2017 13:28 File folder
GL 05/08/2017 13:14 File folder
html 05/08/2017 13:14 File folder
ide 05/08/2017 13:14 File folder
jpeg 05/08/2017 13:14 File folder
lib 05/08/2017 13:27 File folder
misc 05/08/2017 13:14 File folder
png 05/08/2017 13:14 File folder
src 05/08/2017 13:14 File folder
test 05/08/2017 13:28 File folder



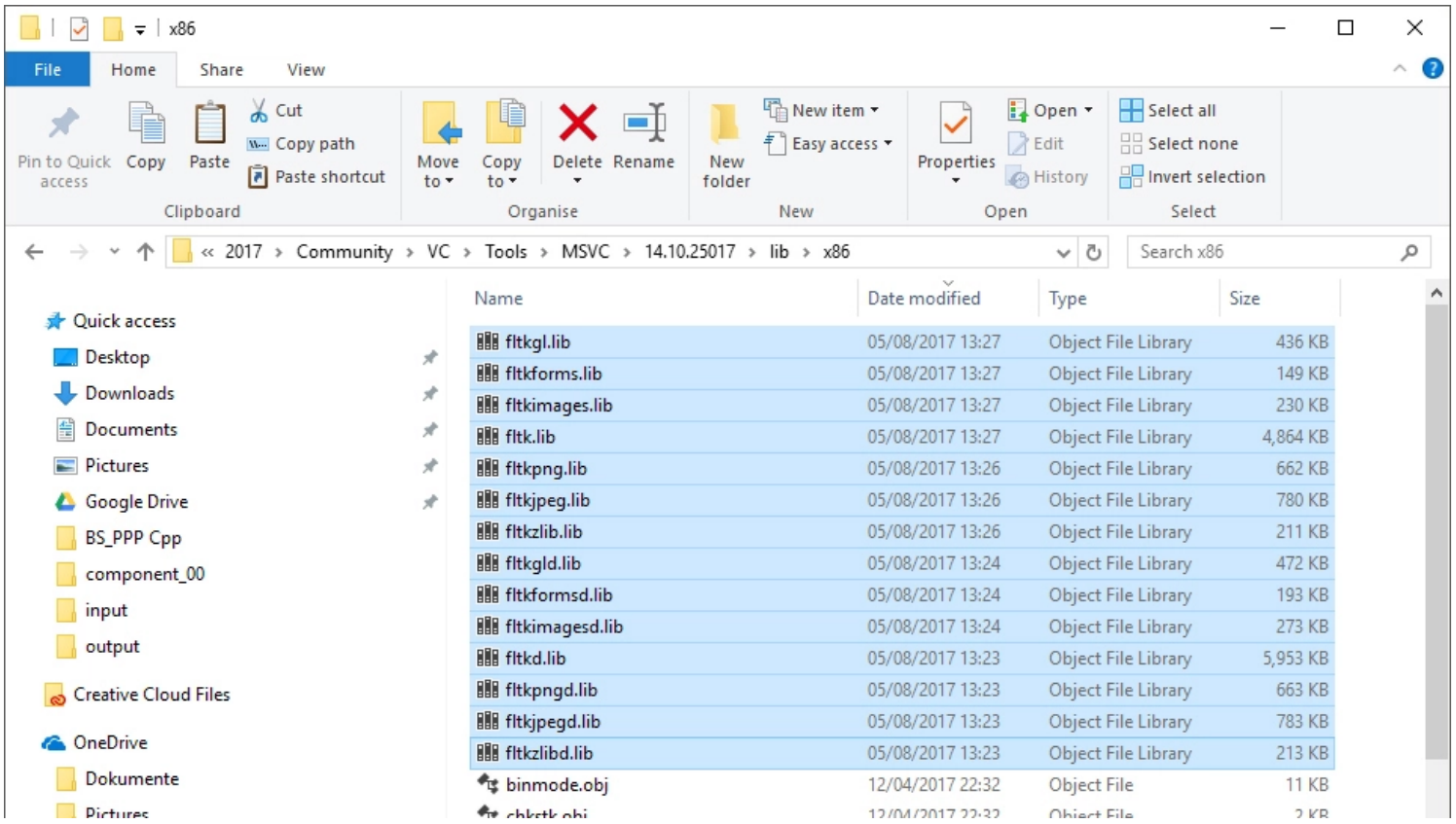
2. Go to: the **lib** folder in your fltk source folder (in my case **C:\Users\Philipp\Documents\dev\fltk-1.3.4-1-source\fltk-1.3.4-1\lib**) and copy all 14 **.lib** files (Note that there are pairs of 2, one of them named with a "d" for debug in the end). And paste them into your Visual Studio 2017 lib folder (in my case: **C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\VC\Tools\MSVC\14.10.25017\lib\x64**). Of course you noticed that in the lib folder there are two more folders: **x64** and **x86**. I just pasted the 14 .lib files into both of them (In case there is someone out there who can tell me if that is wrong or right, please contact me).



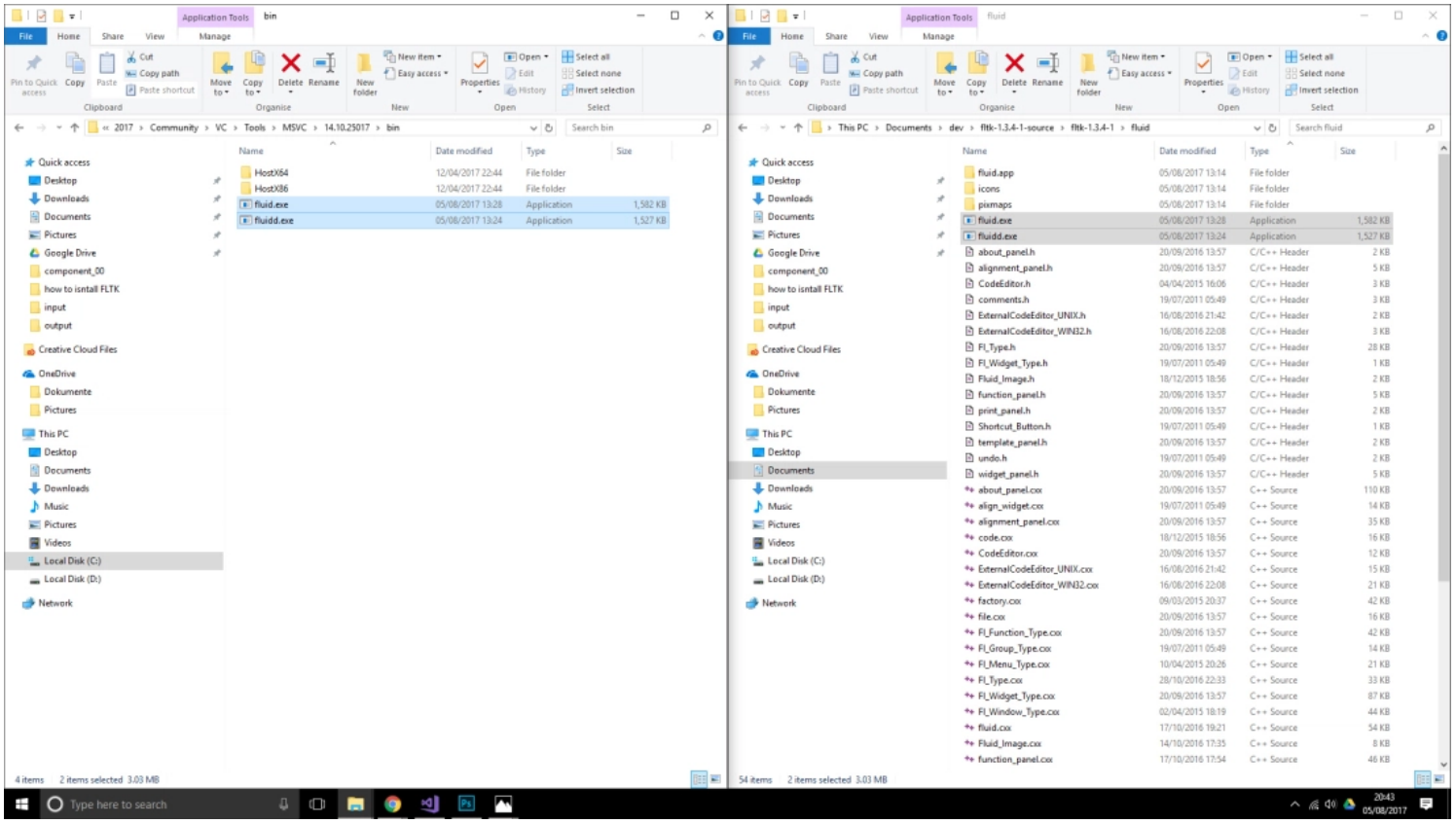
fltk-1.3.4-1\lib to **VC\Tools\MSVC\14.10.25017\lib\x64**

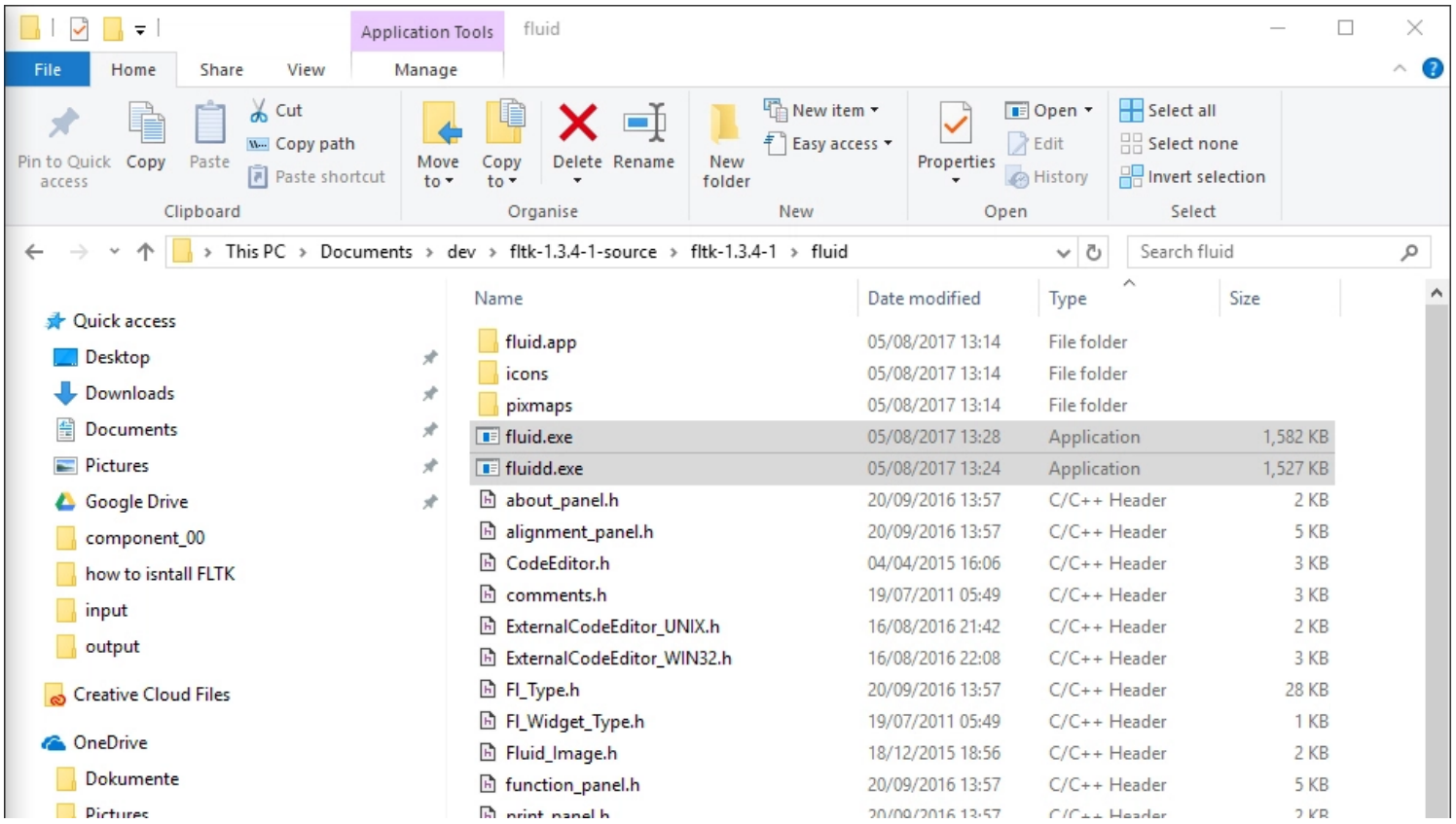


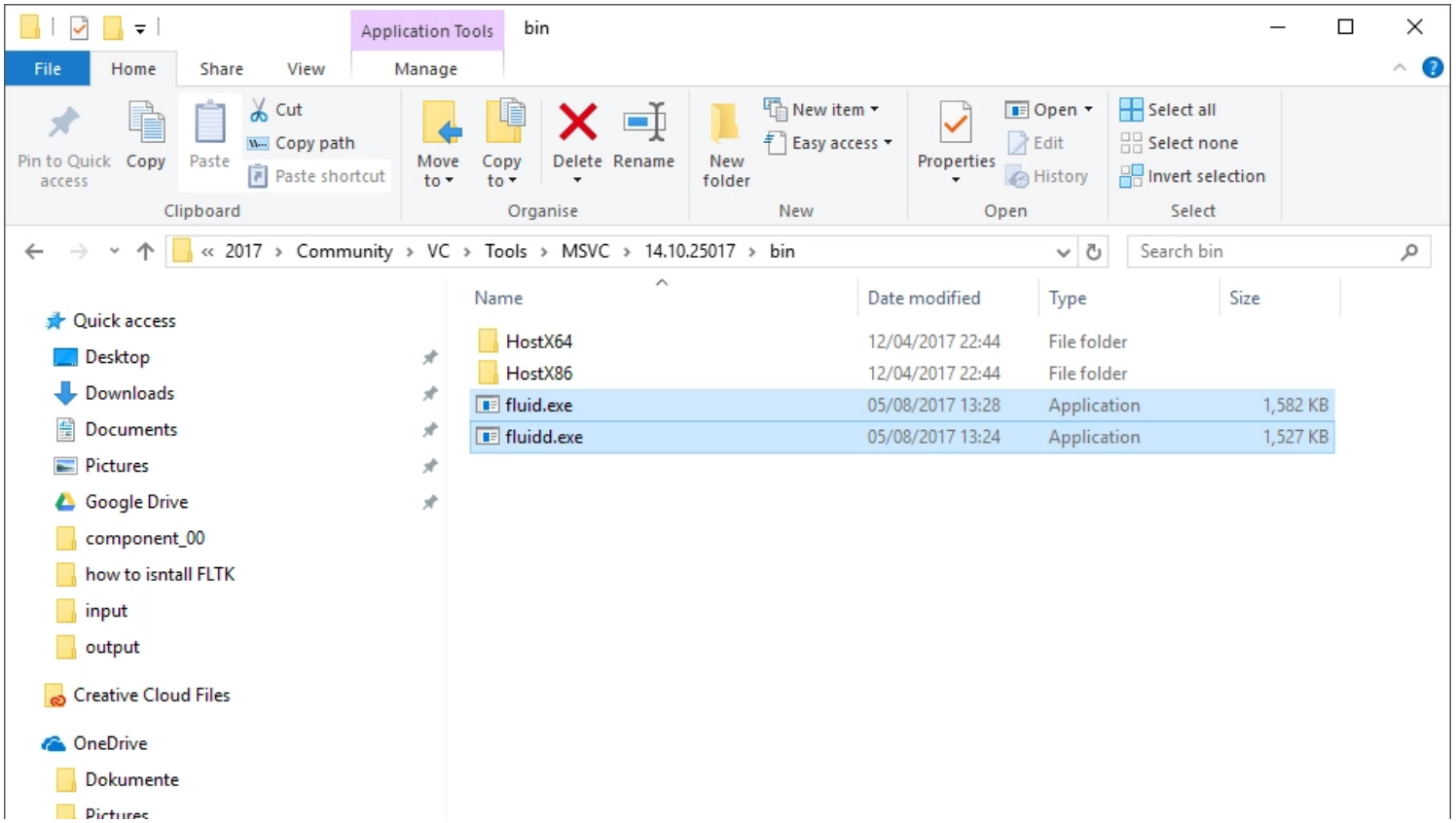
fltk-1.3.4-1\lib to **VC\Tools\MSVC\14.10.25017\lib\x86**



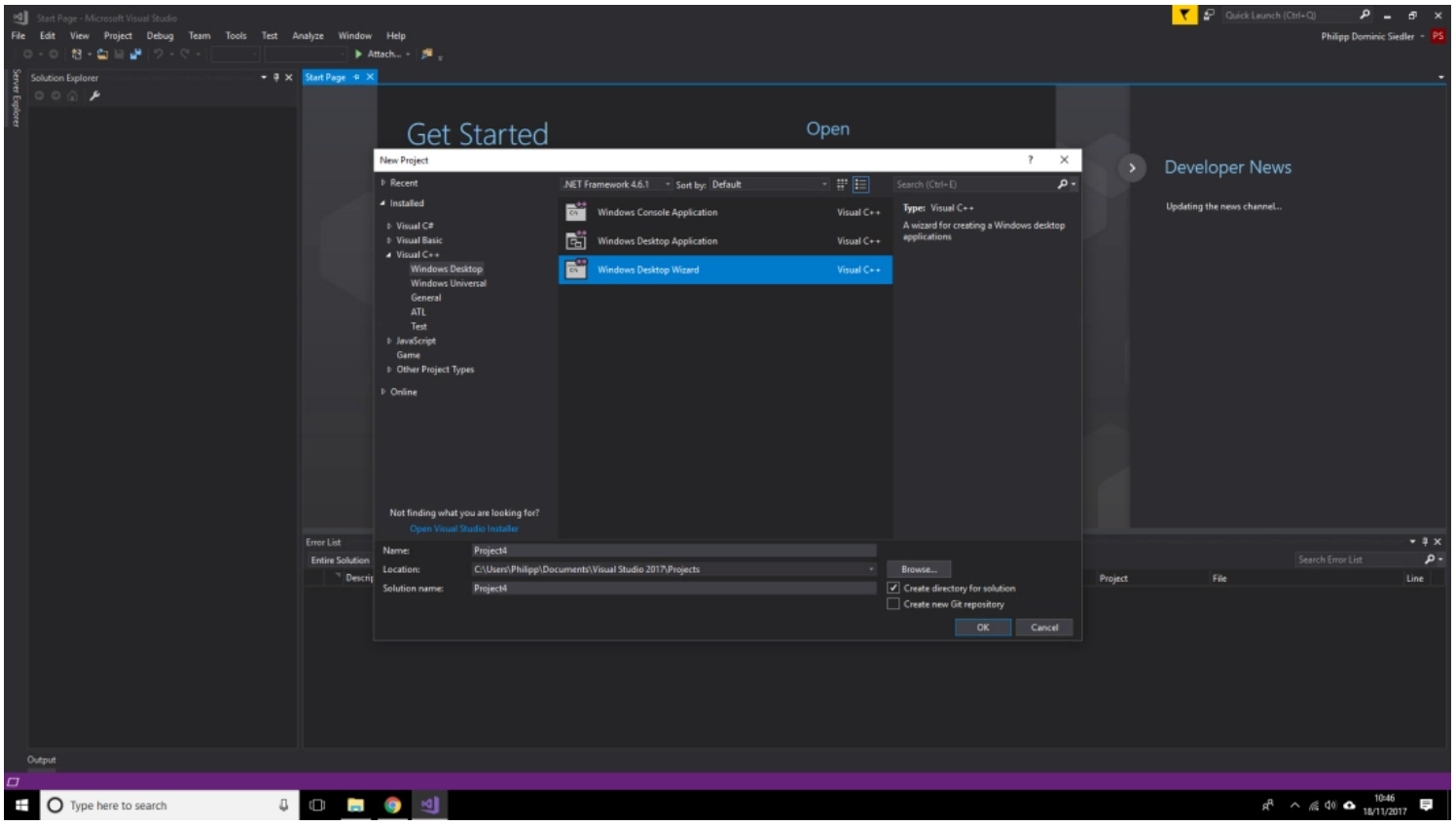
3. Go to: the **fluid** folder in your fltk source folder (in my case: **C:\Users\Philipp\Documents\dev\fltk-1.3.4-1-source\fltk-1.3.4-1\fluid**) and copy **fluid.exe** and **fluidd.exe** into your Visual Studio 2017 **bin** folder (in my case: **C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\VC\Tools\MSVC\14.10.25017\bin**).



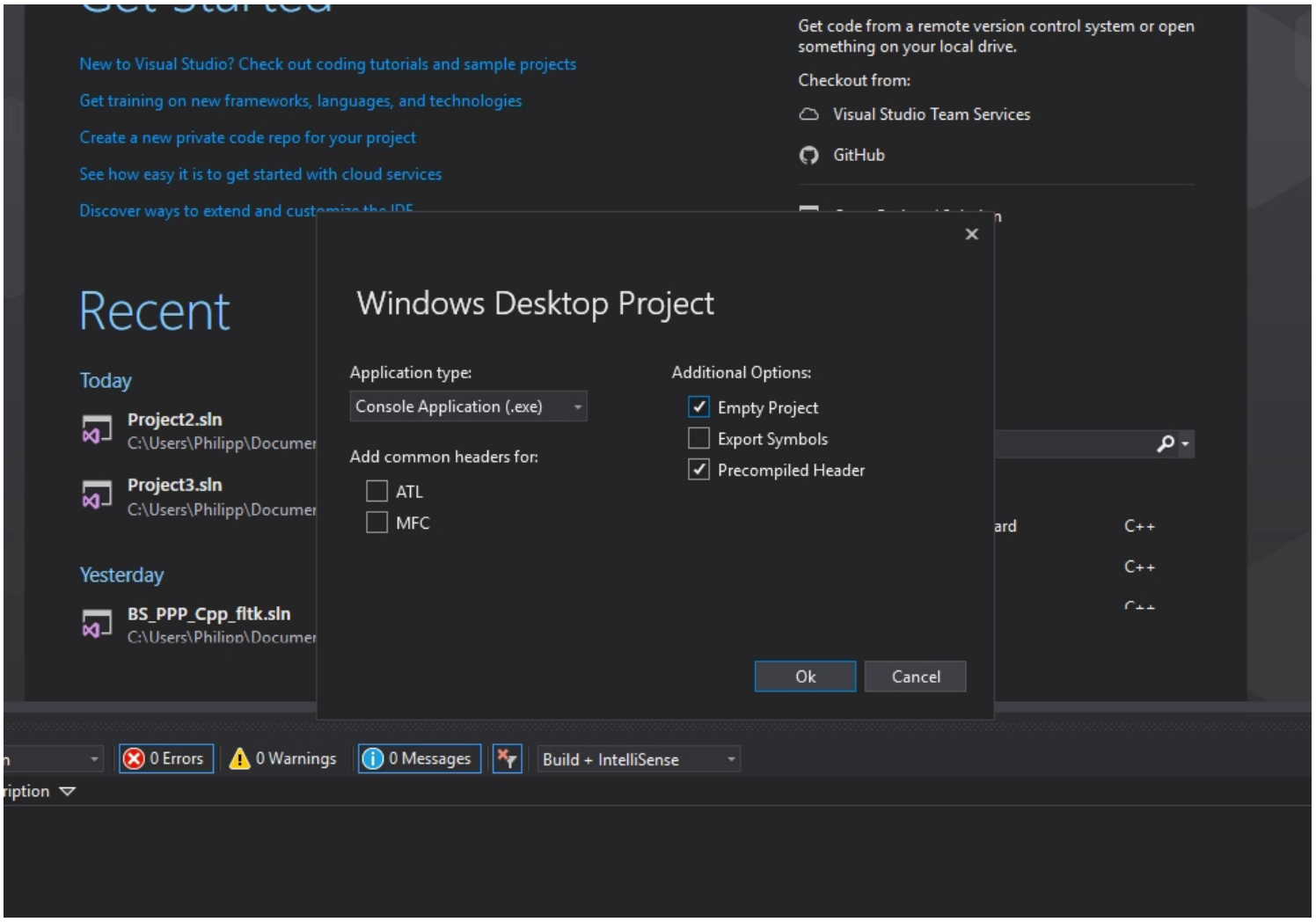




Excellent, that's all done. Now lets setup a new project and see if Visual Studio 2017 Community is able to locate and use the files we just copied over.
Let's open Visual Studio 2017 Community (or any other of course) and under the **Visual C++ – Windows Desktop** tab create a new **Windows Desktop Wizard** project.

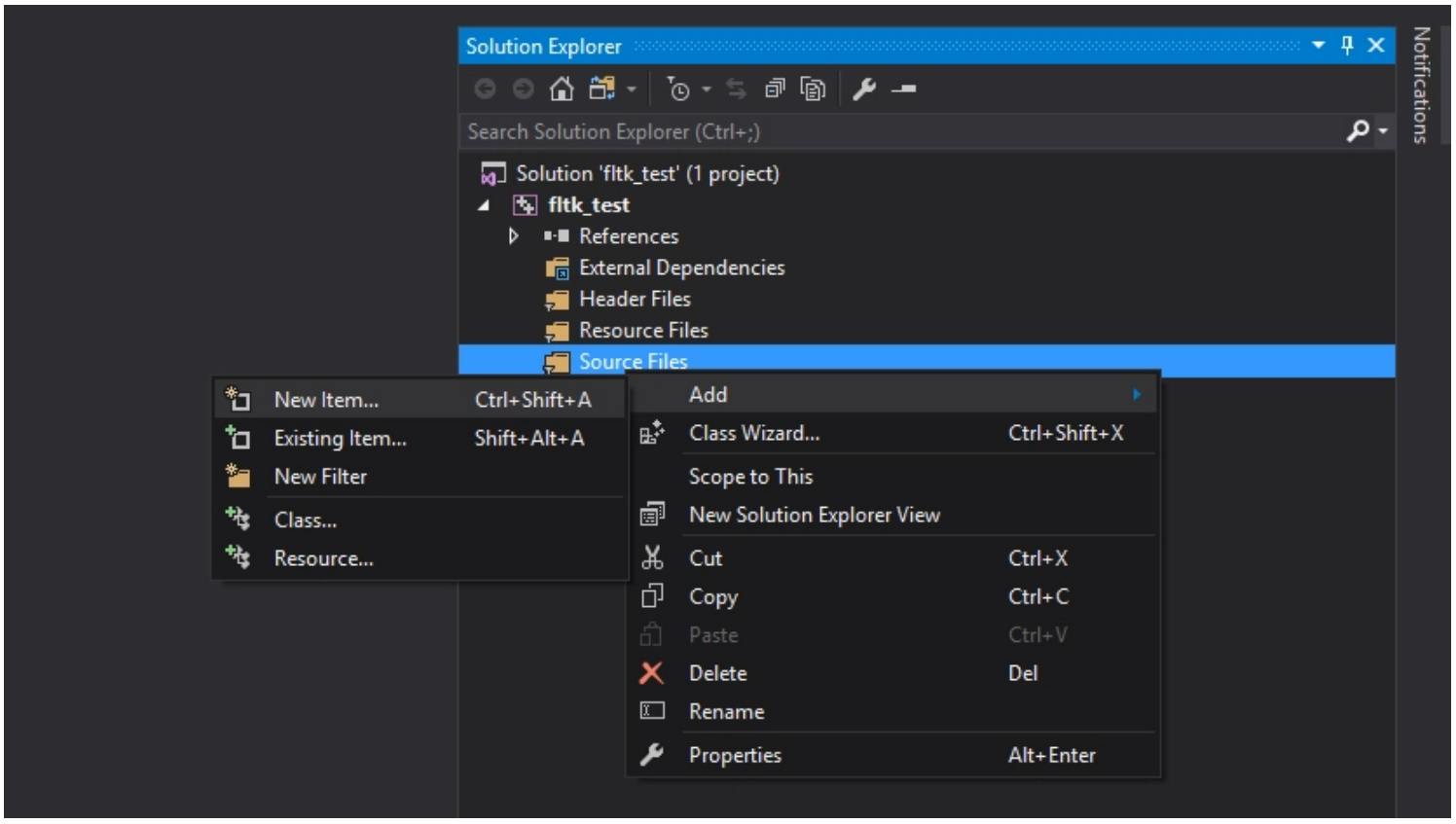

Get Started
Open
Developer News
New Project
Recent
Installed
Visual C#
Visual Basic
Visual C++
Windows Desktop
Windows Universal
General
ATL
Test
JavaScript
Game
Other Project Types
Online
.NET Framework 4.6.1
Sort by: Default
Windows Console Application
Windows Desktop Application
Windows Desktop Wizard
Visual C++
Type: Visual C++
A wizard for creating a Windows desktop applications
Not finding what you are looking for?
Open Visual Studio Installer
Name: Project4
Location: C:\Users\Philipp\Documents\Visual Studio 2017\Projects
Solution name: Project4
Browse...
Create directory for solution
Create new Git repository
OK
Cancel

Tick the box **Empty project** (very important).



I called it fltk_test.

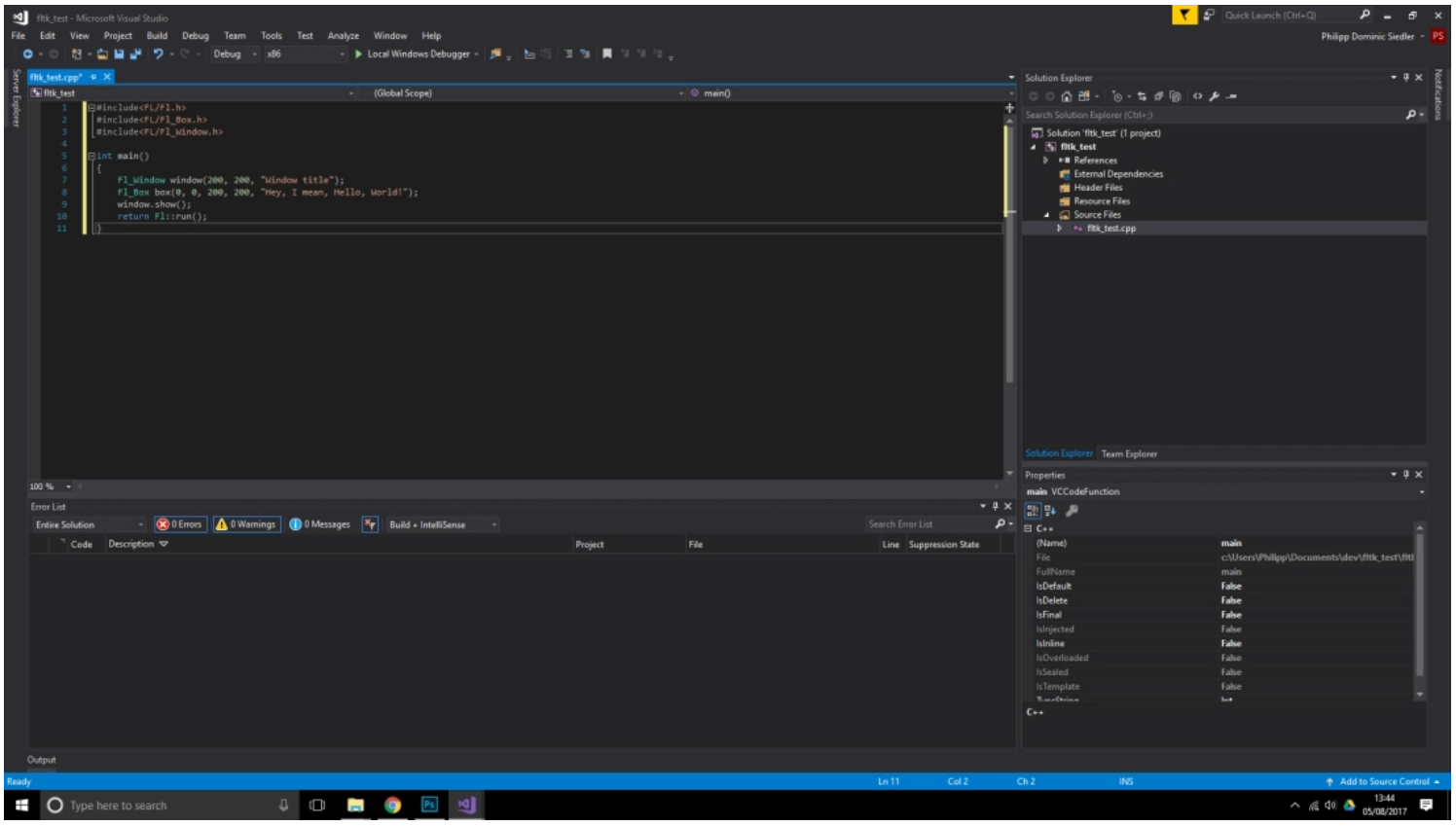
In your newly created **Empty Project** right click the **Source Files** folder and **Add** a **New Item** which is going to be a **.cpp** file.
I called it **fltk_test.cpp**.



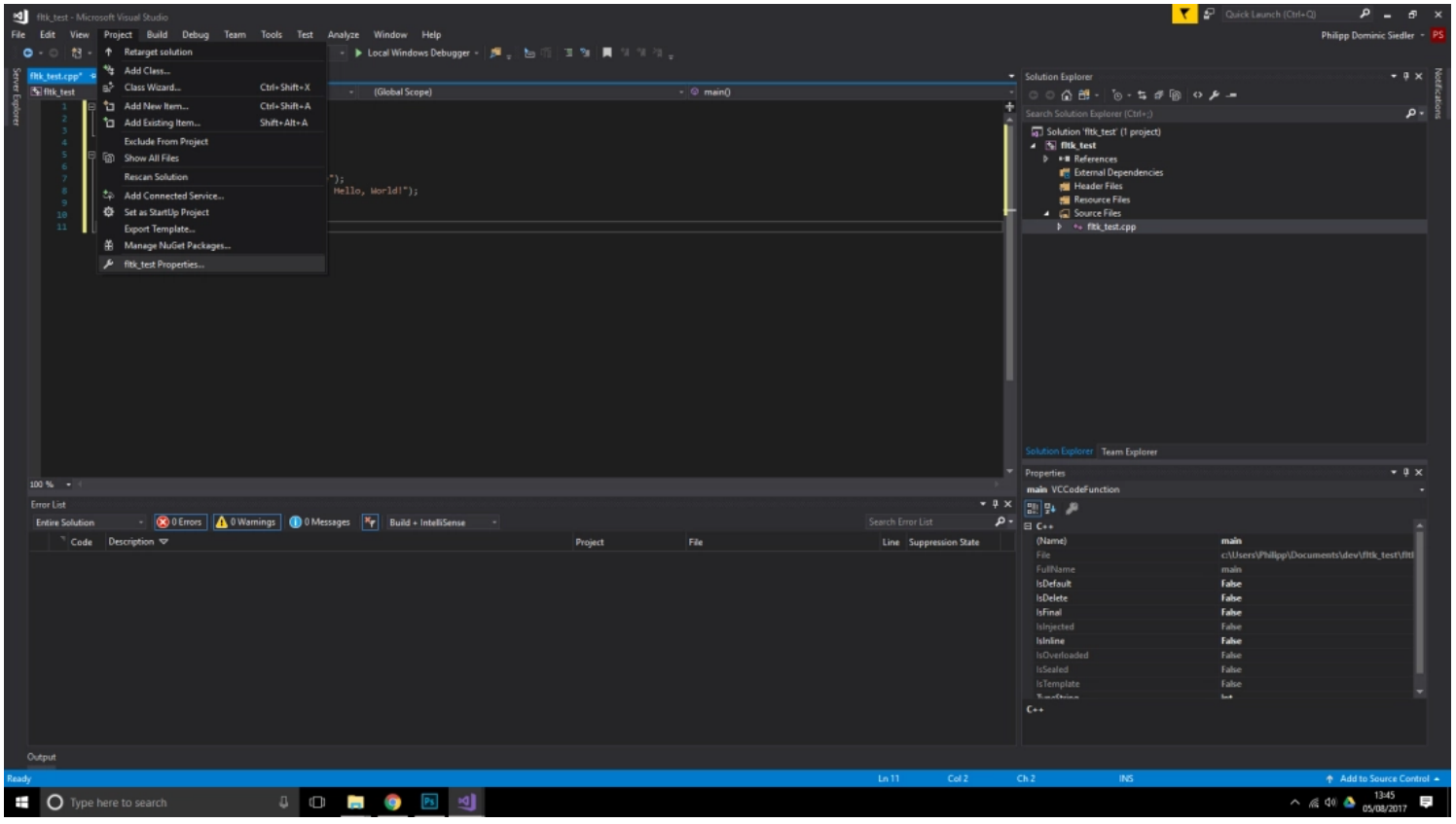
Copy the following code into your .cpp file:

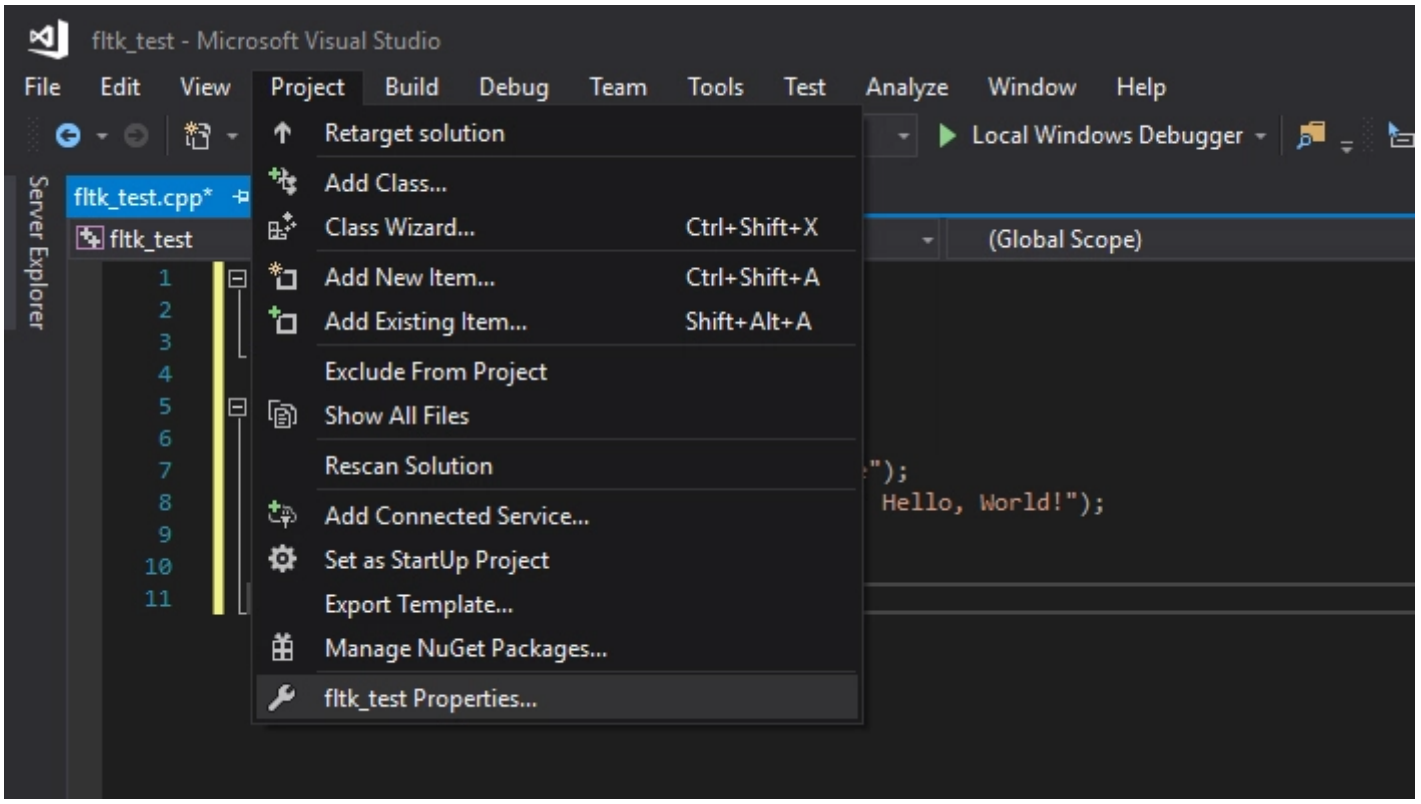
```
#include<FL/Fl.h>
#include<FL/Fl_Box.h>
#include<FL/Fl_Window.h>

int main()
{
    Fl_Window window(200, 200, "Window title");
    Fl_Box box(0, 0, 200, 200, "Hey, I mean, Hello, World!");
    window.show();
    return Fl::run();
}
```

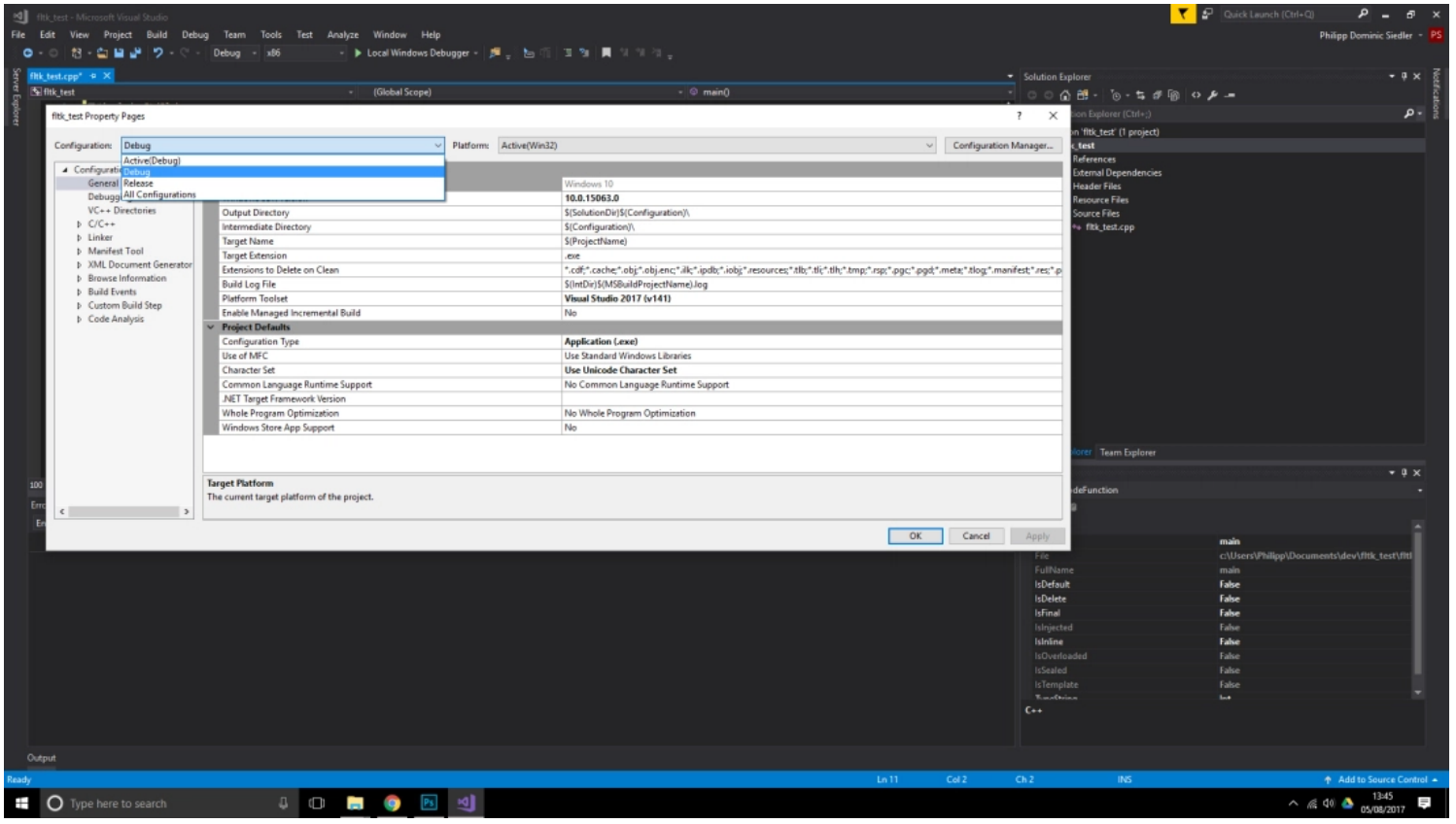


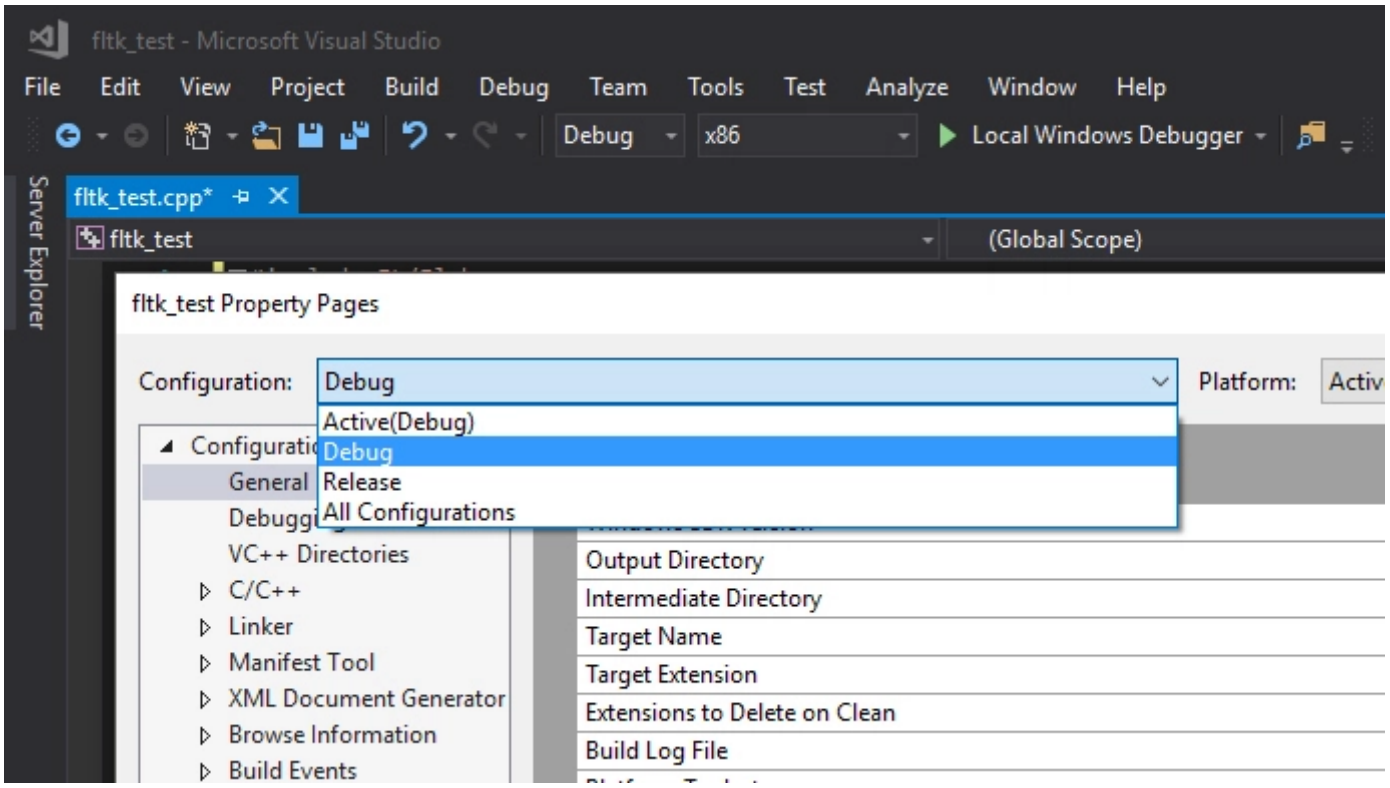
(NOTE: The following Properties setup needs to be done for every new fltk project you are creating! I will tell you later to come back to this point.)
Then go to your top navigation bar and click **Project** and in the drop-down menu **Properties**.



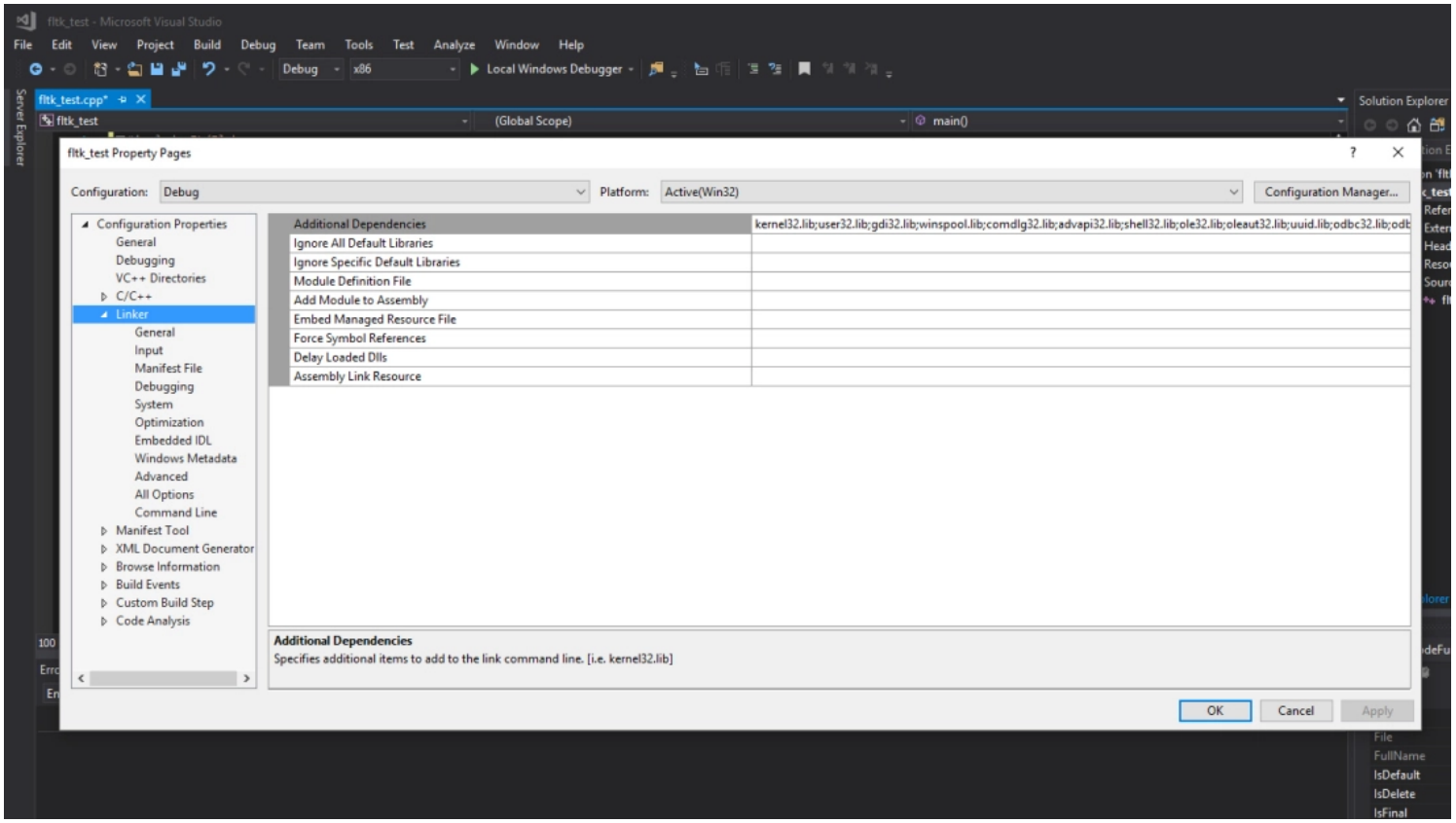


Make sure to set the Configuration to **Debug**.

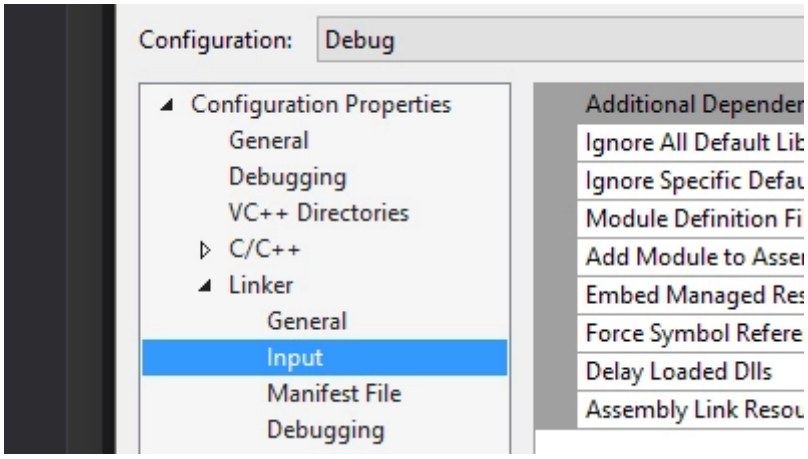




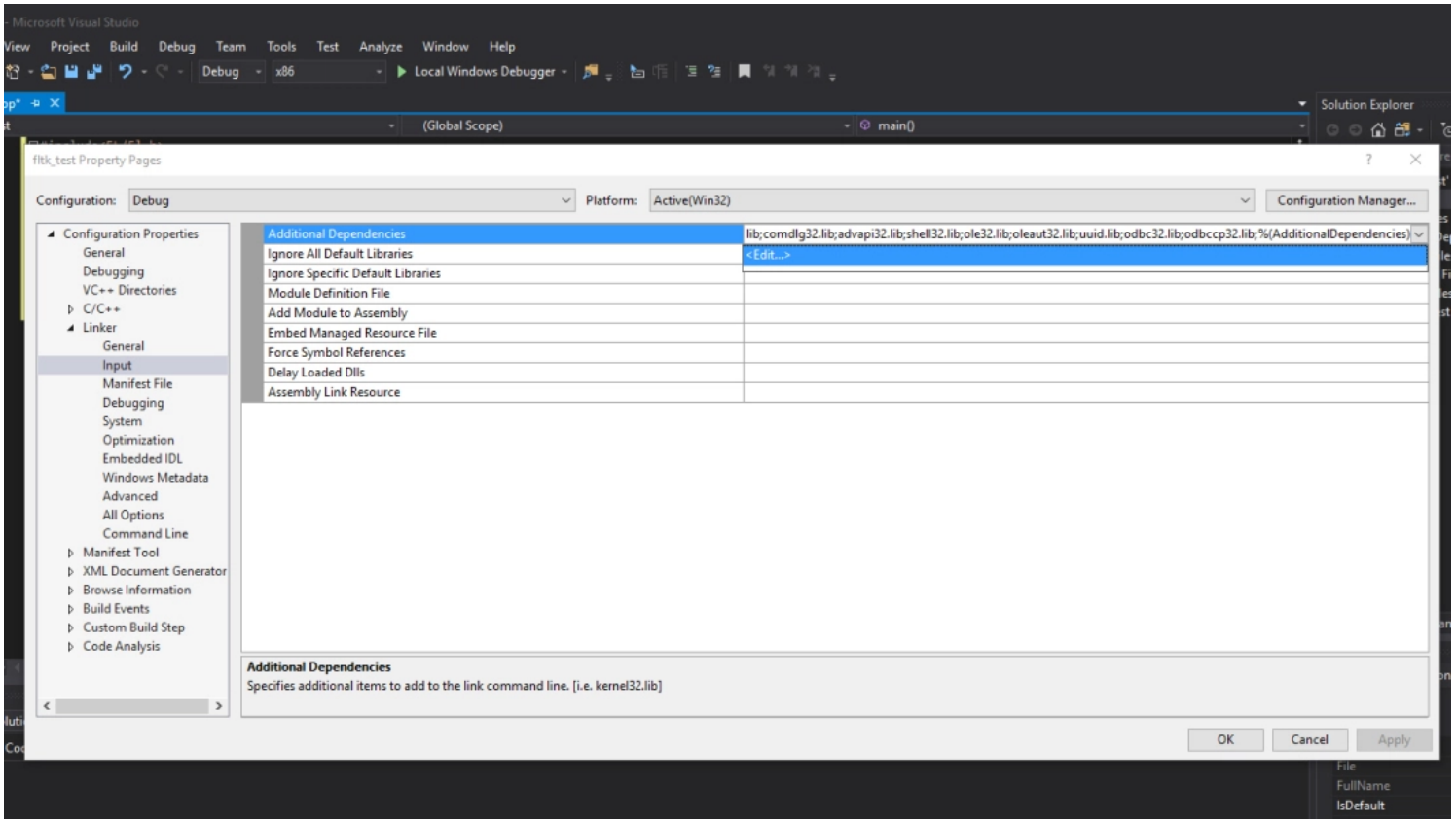
On your left hand side go to **Linker**.



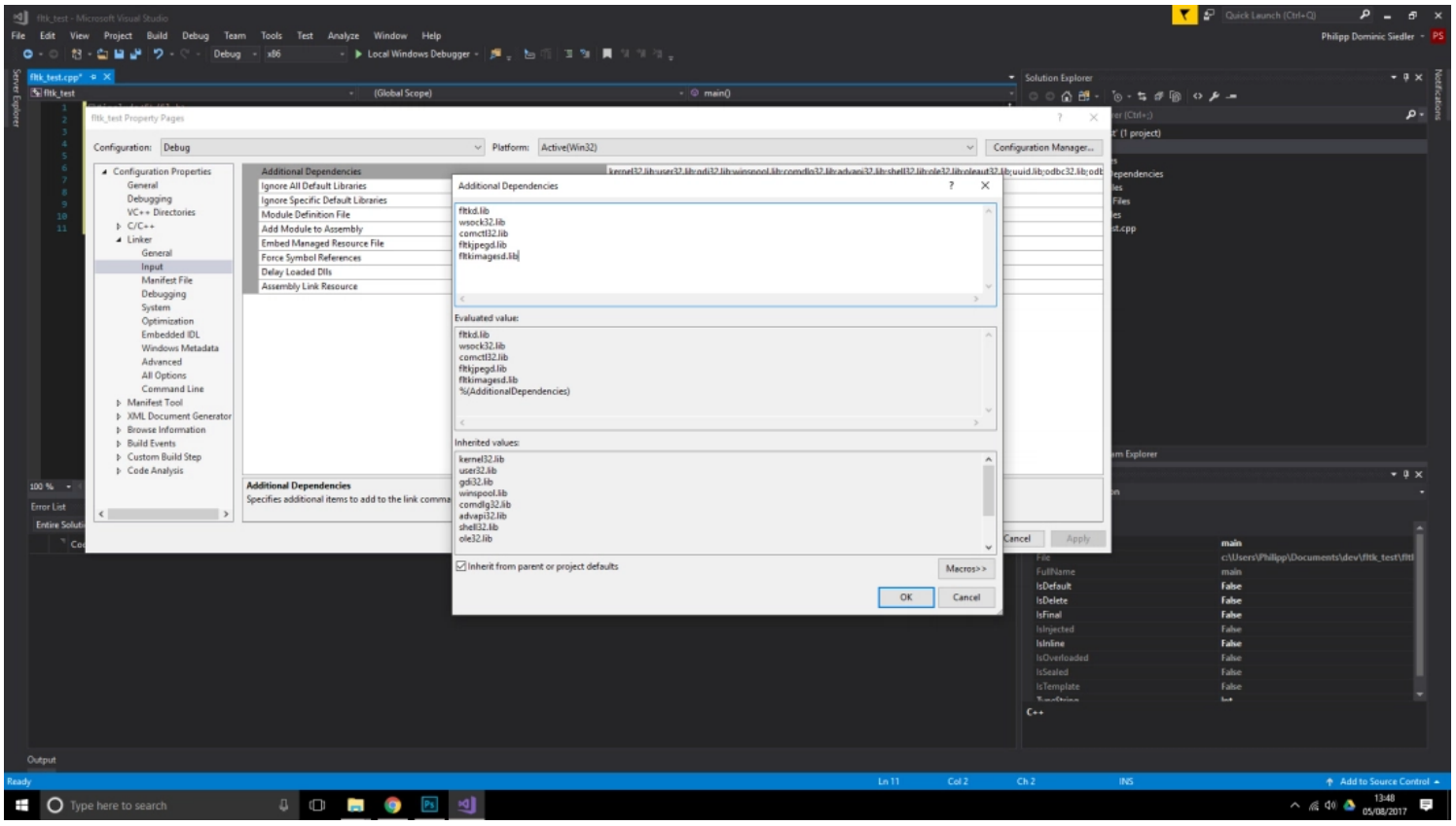
And **Input**.



Click on the drop-down arrow in the **Additional Dependencies** tab.



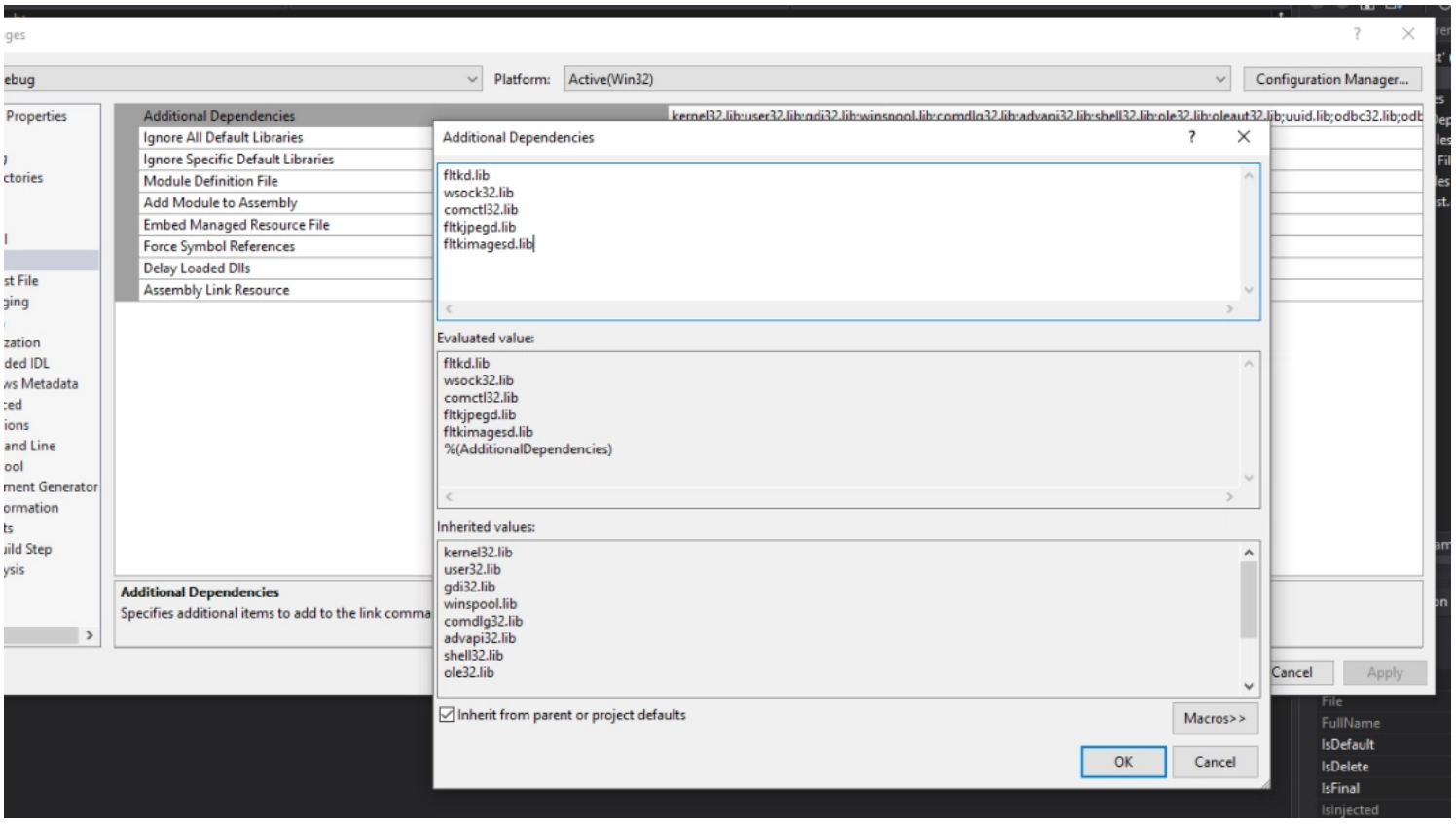
And click on **Edit**.



A new window will pop up. In the screenshot shown I filled in already the following entries:

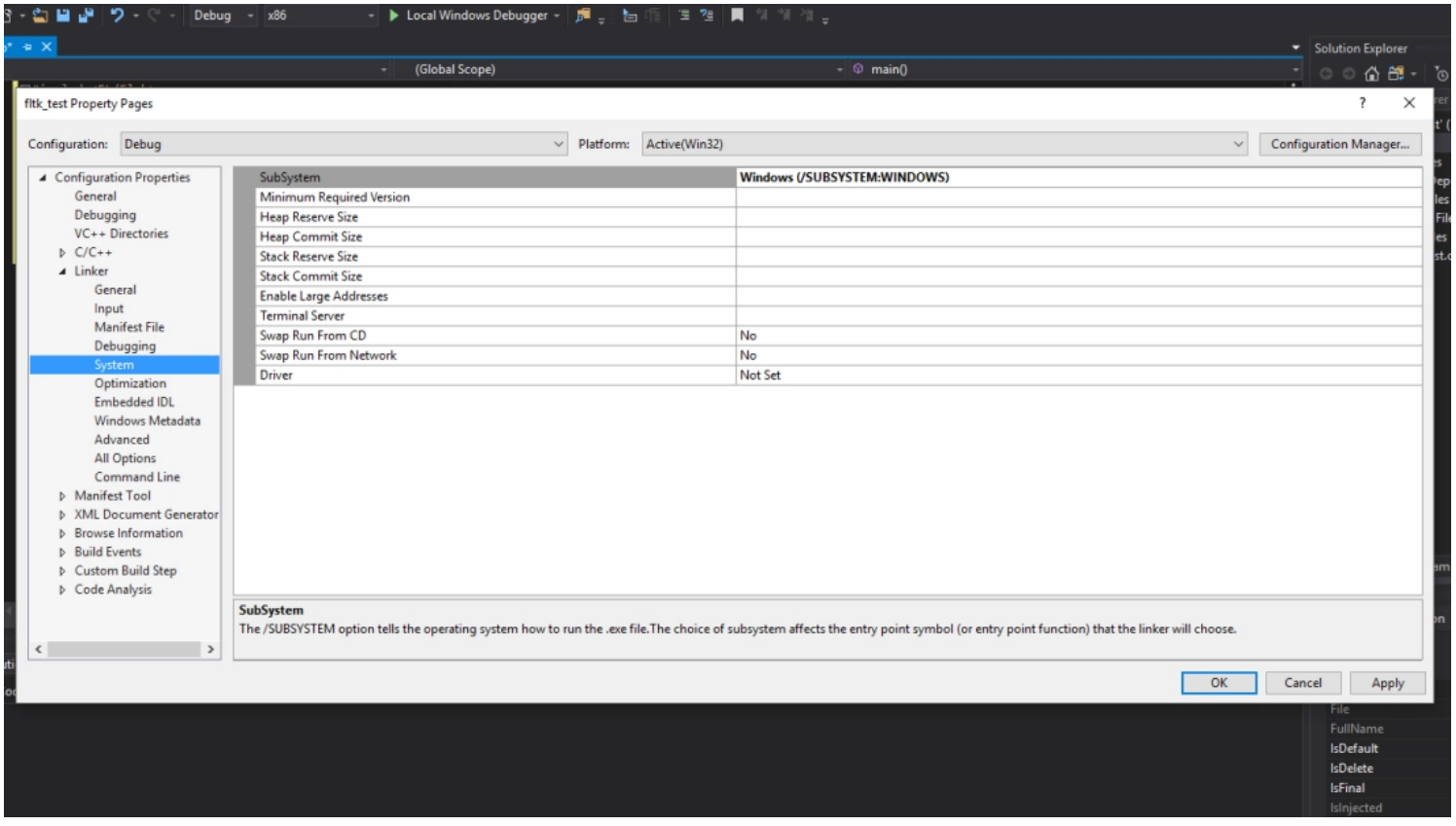
fltkd.lib
wsock32.lib
comctl32.lib
fltkjpegd.lib
fltkimagesd.lib

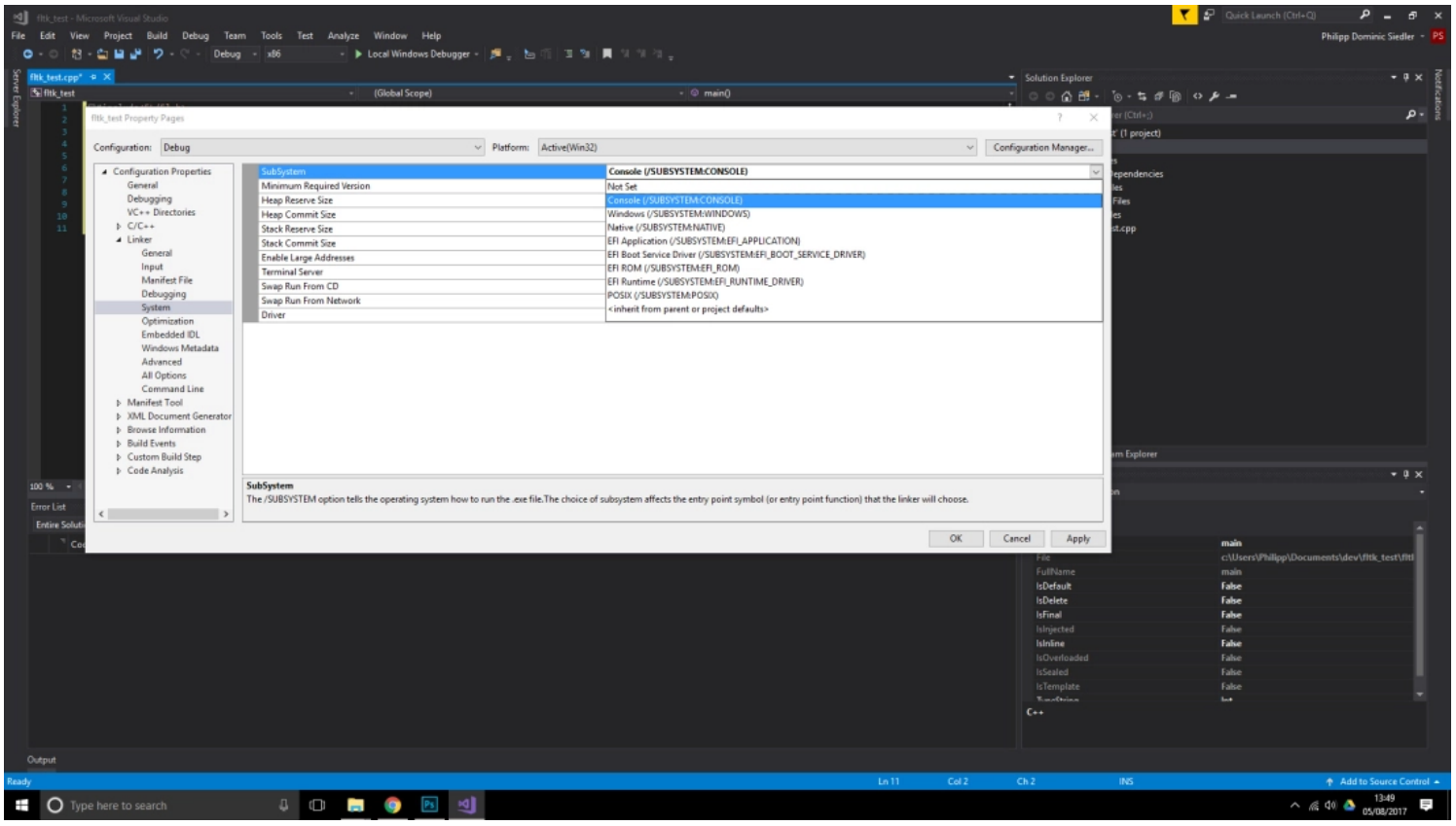
Please do the same.



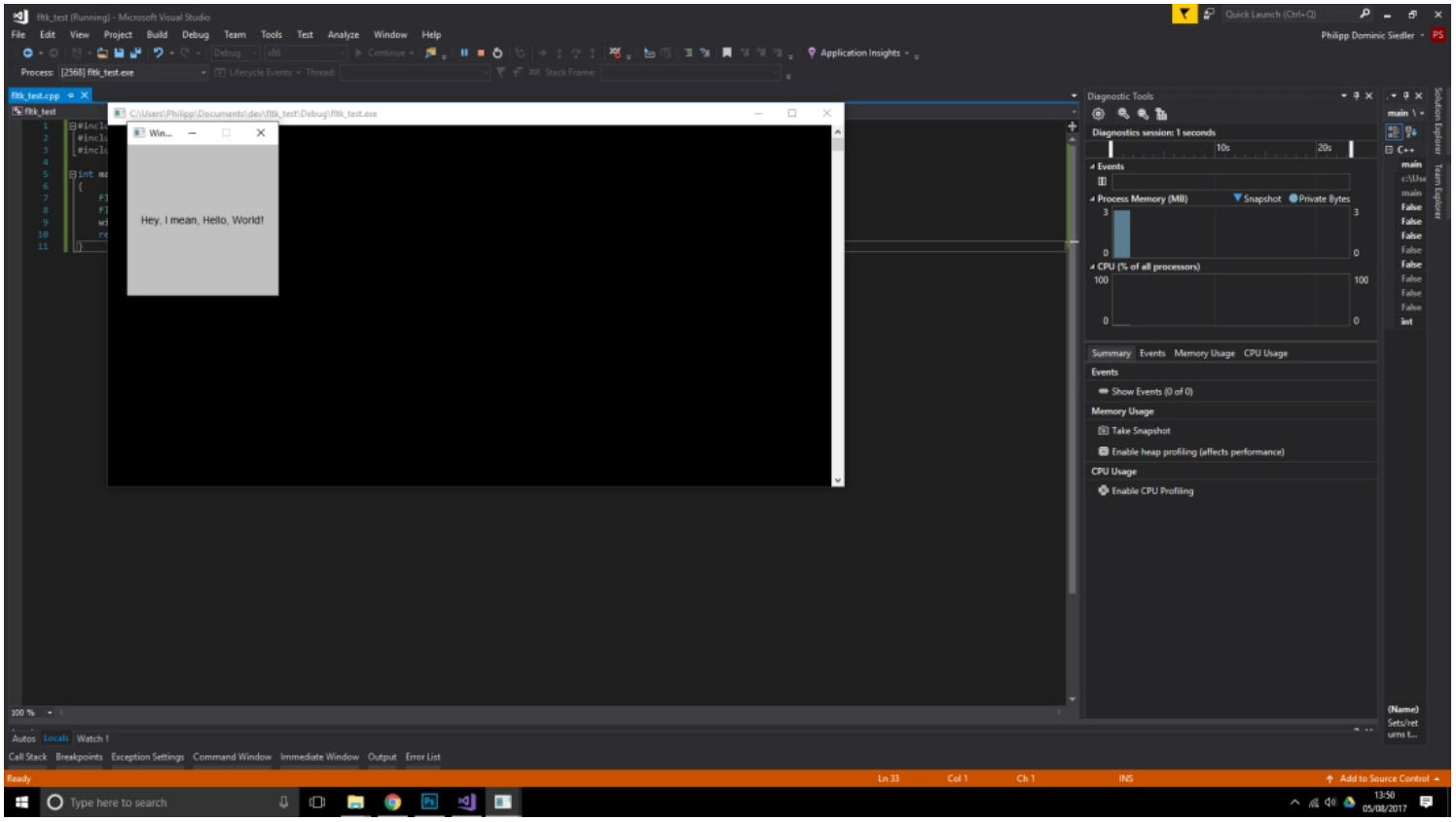
Click **OK** to confirm your entry.

Now go to **System**.
And set the **SubSystem** to **Console (/SUBSYSTEM:CONSOLE)** with the help of the drop-down menu.

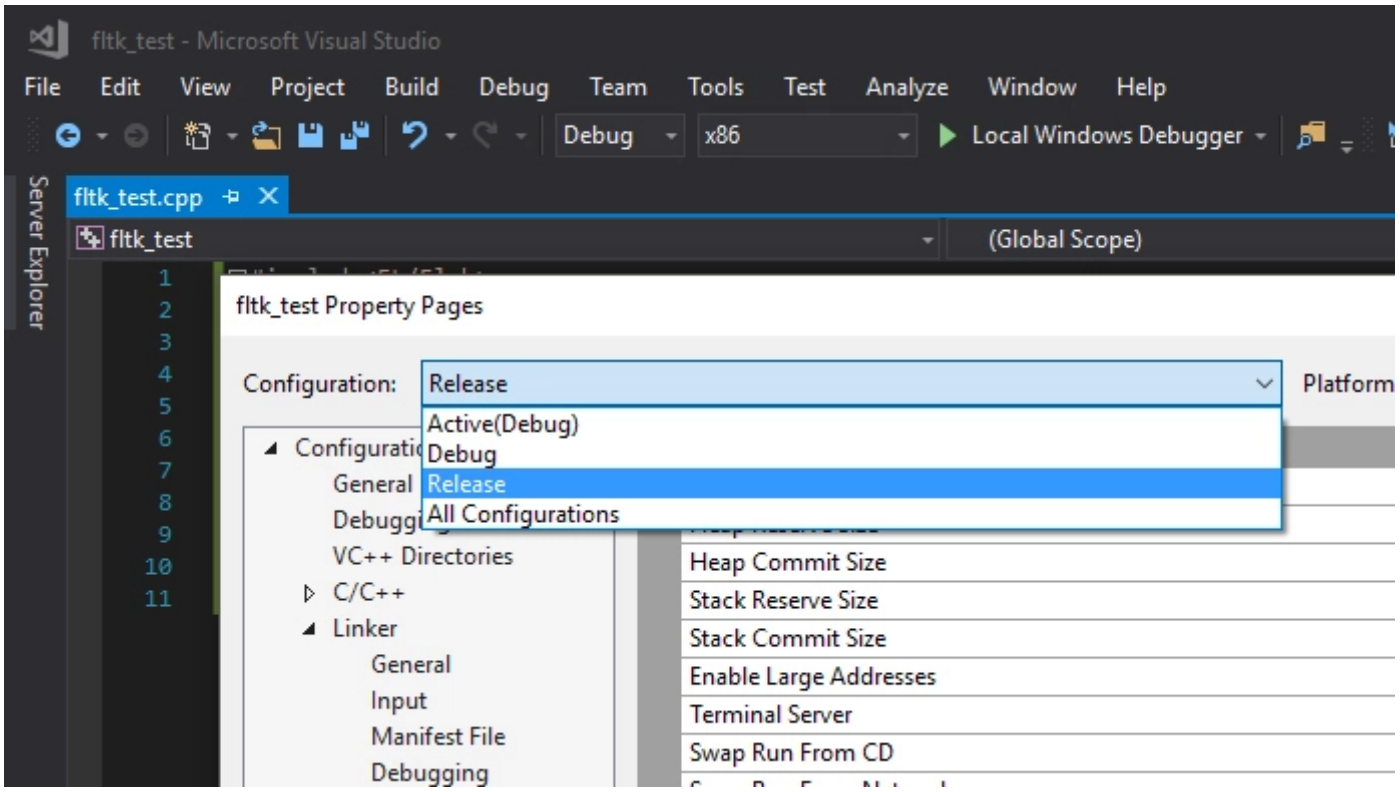




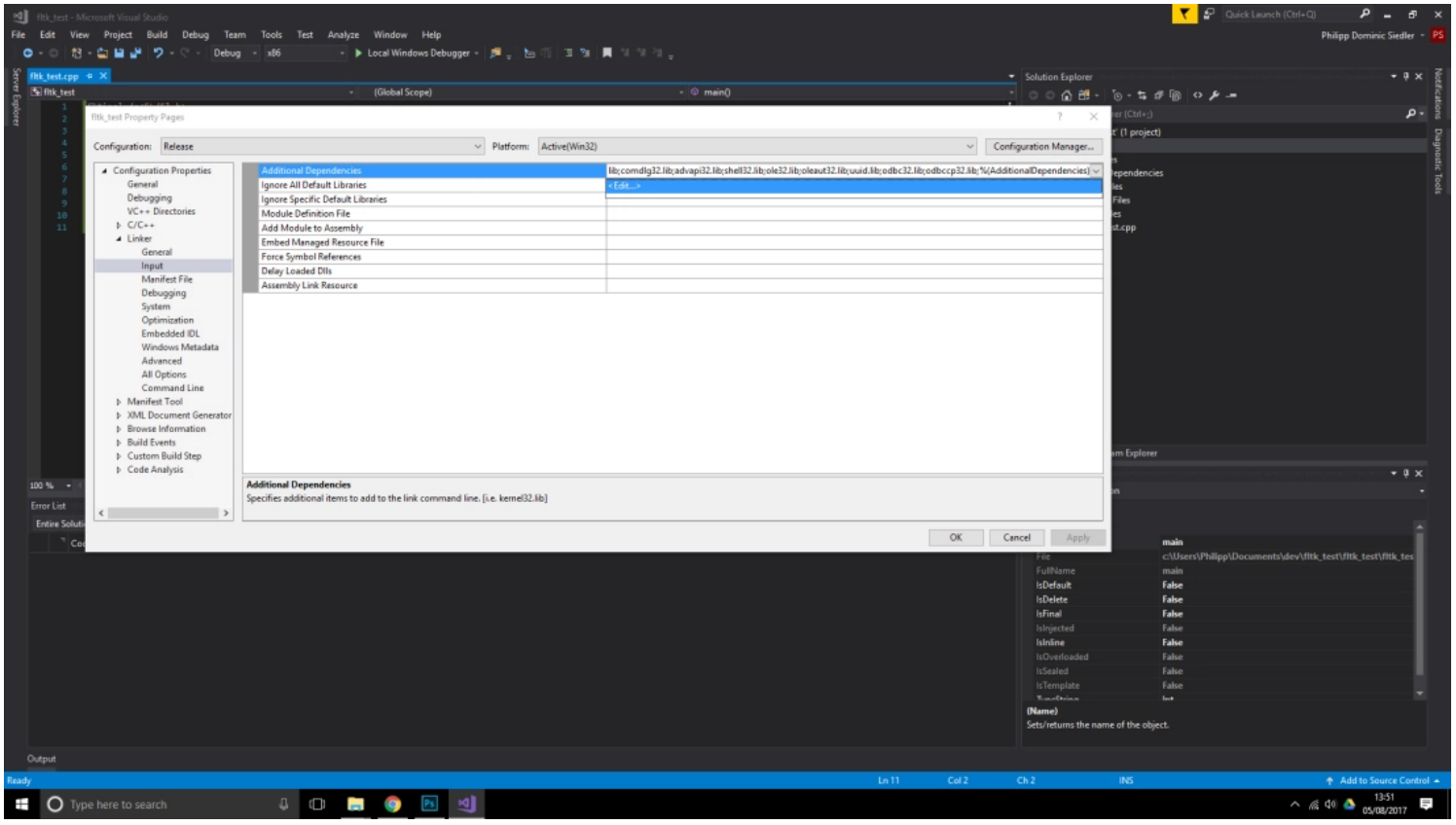
Click **Apply** and hit **F5** to test-run the project.
If you have done everything I told you so far your project should compile and run and show something like this:



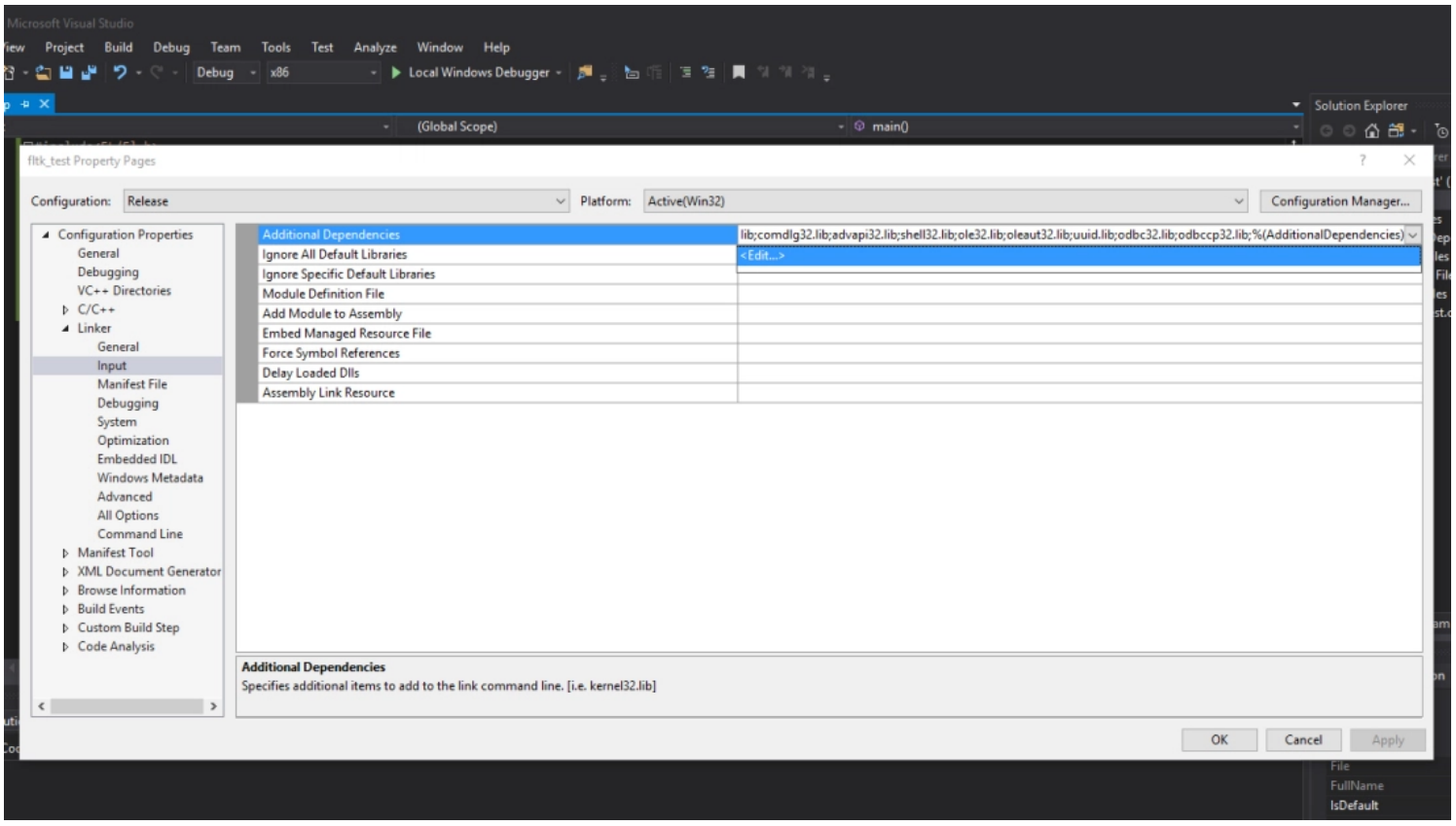
Great, now go back to your **Properties – Project** and **Properties**.
And chose from the Configuration **Release**.



On the left hand side to **Linker** and then to **Input**, just like you did before.

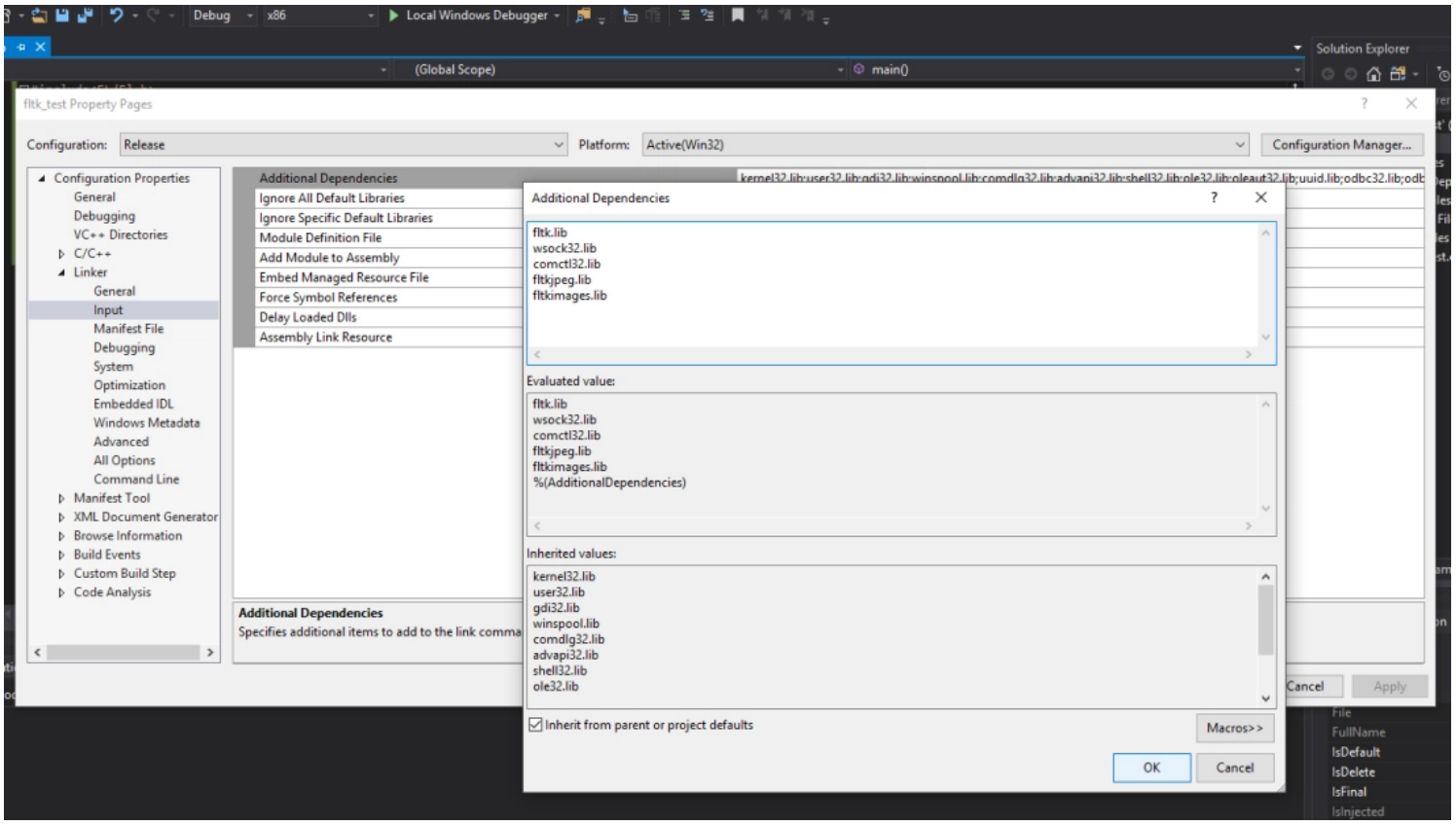


Click on the drop-down arrow to show the **Additional Dependencies** menu and select **Edit**.

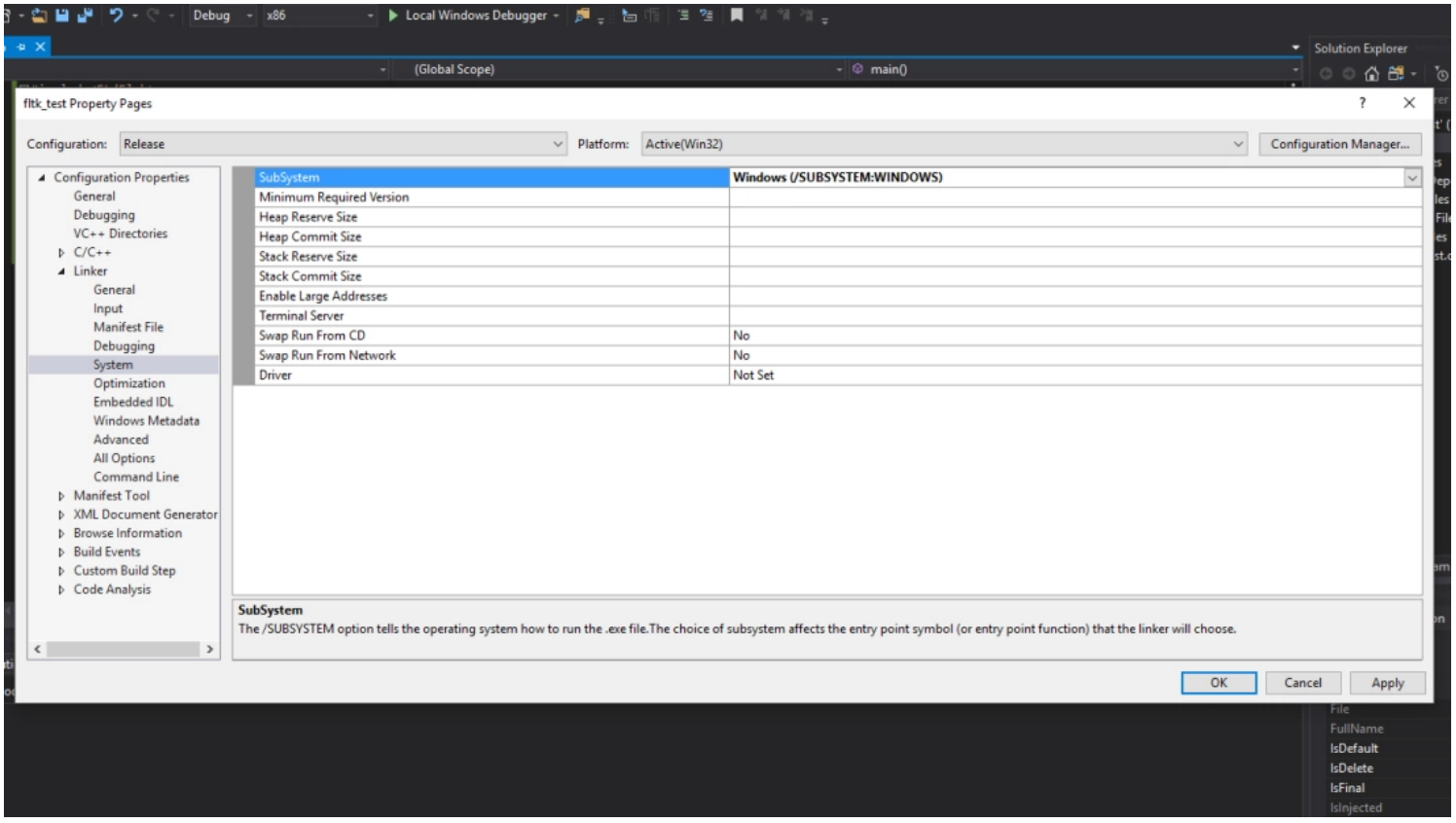


Copy paste the following entries into the empty field (I filled them out already, Note: Since this is the Release Category there is no d for debug in the .lib names):

fltk.lib
wsock32.lib
comctl32.lib
fltkjpeg.lib
fltkimages.lib

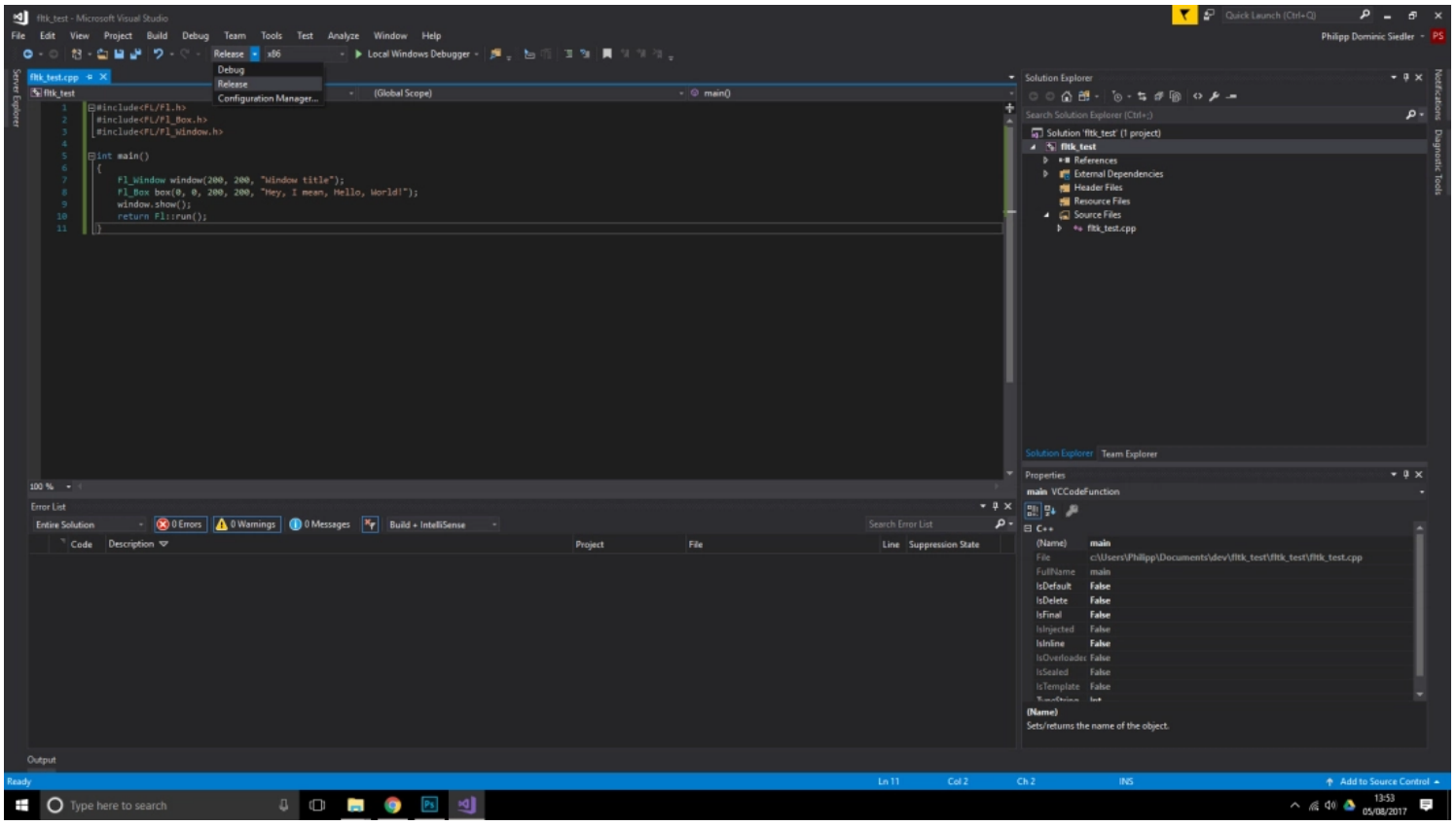


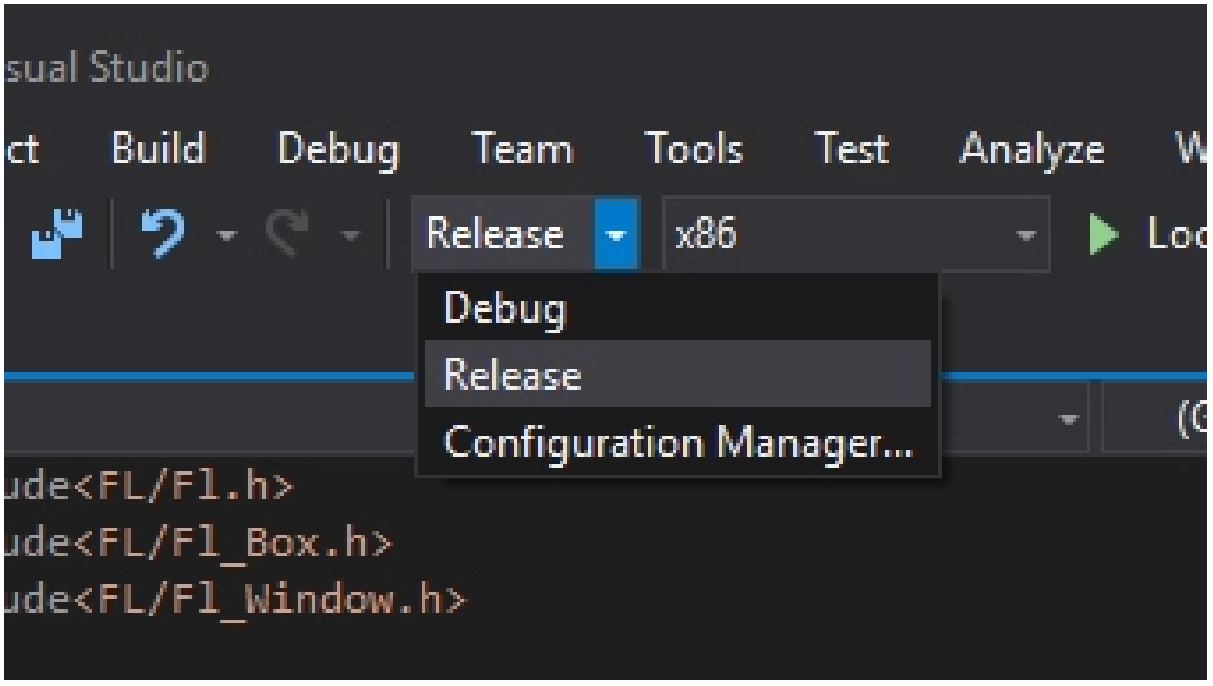
Click **OK** and go to **Sysyem**.



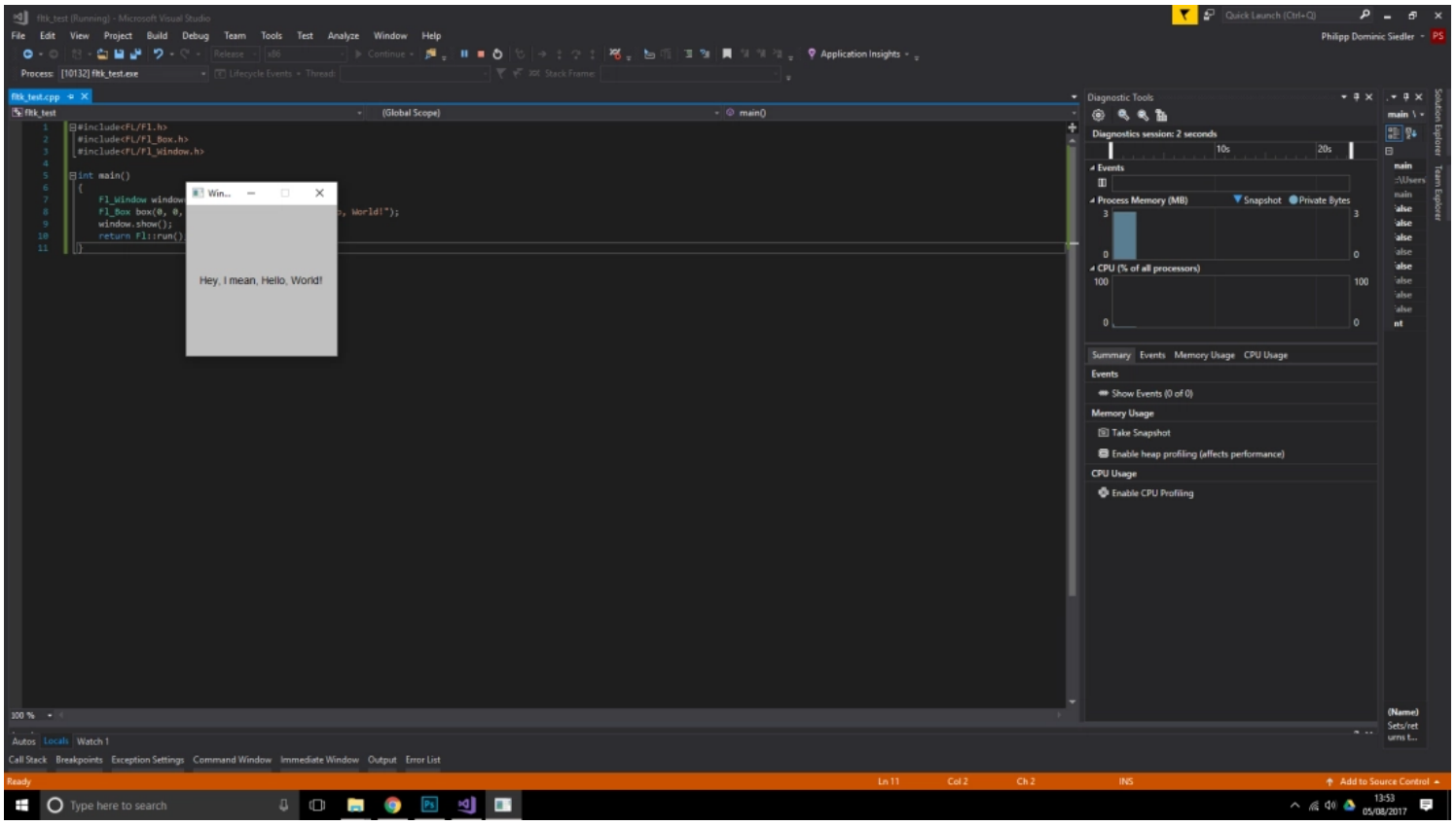
Set the **SubSystem** to **Windows (/SUBSYSTEM:WINDOWS)**, which can be found in the drop-down menu and click **OK**.

Apply your changes and exit your **Properties** window with **OK**.
Now change the **Solution Configurations** to **Release**.



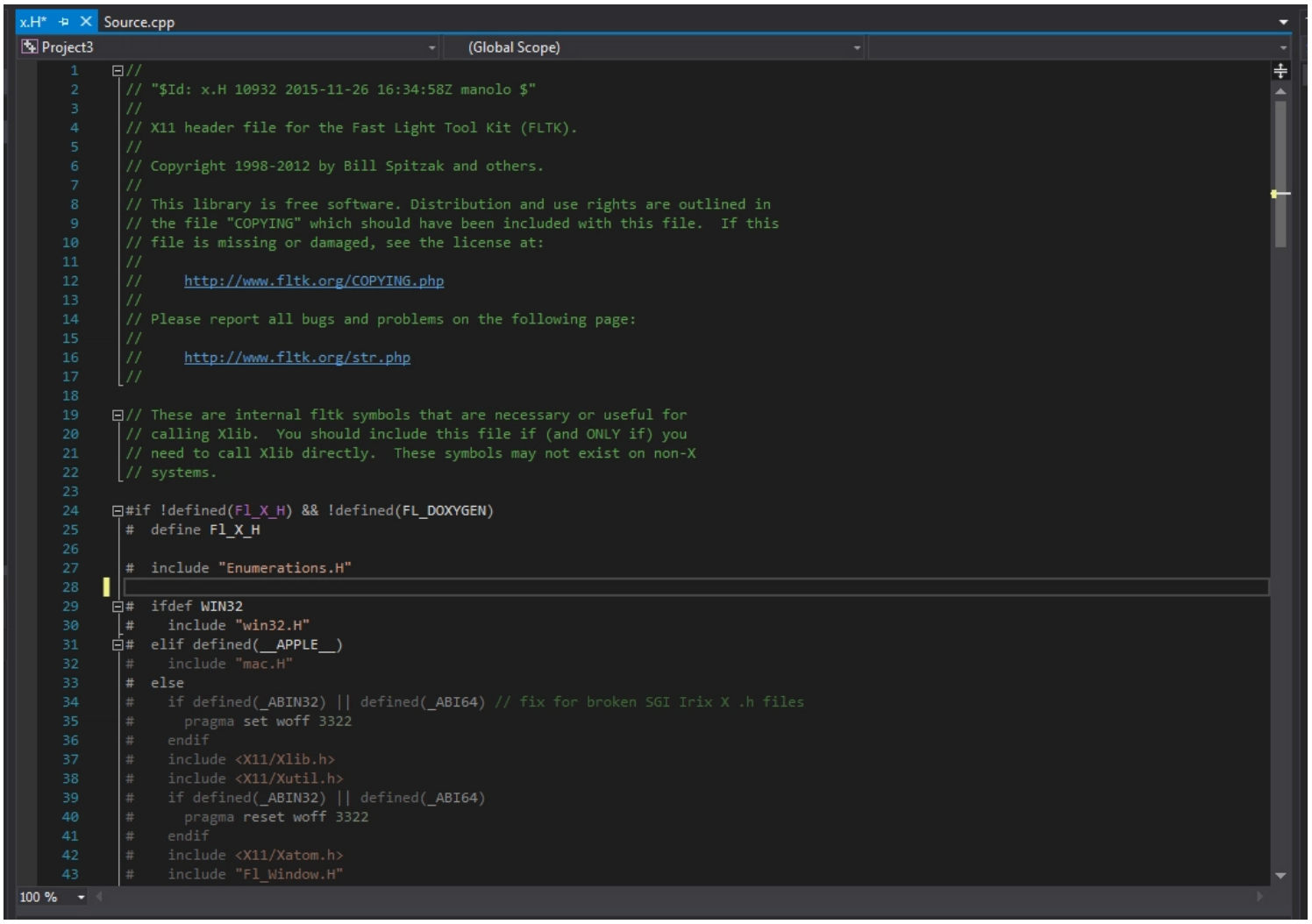


And run your project by pressing **F5** on your keyboard, or the playbutton on the top of your canvas **Local Windows Debugger**.
If everything went well you should be able to look at something like this (application window with the fltk-interface and no console window, if this didn't go as planned please go to the next step):



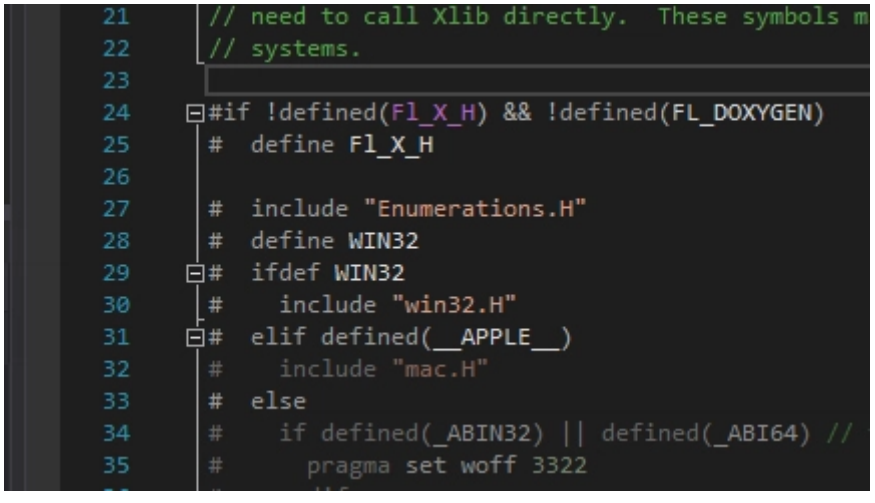
At this point maybe people experience an error message saying "… **cannot open include file 'x11/xlib.h' no such file or directory** …". Here is how to fix this:
(If everything works for you just fine, please skip ahead!)

Double click the error message in your errors tab, the **x.H** file will open in **Visual Studio 2017 Community** as a new tab, if not go to your **External Dependencies** folder in your **Solution Explorer** within Visual Studio and look for the file manually and open it. As you may notice already, the errors will be underlined in curly red.

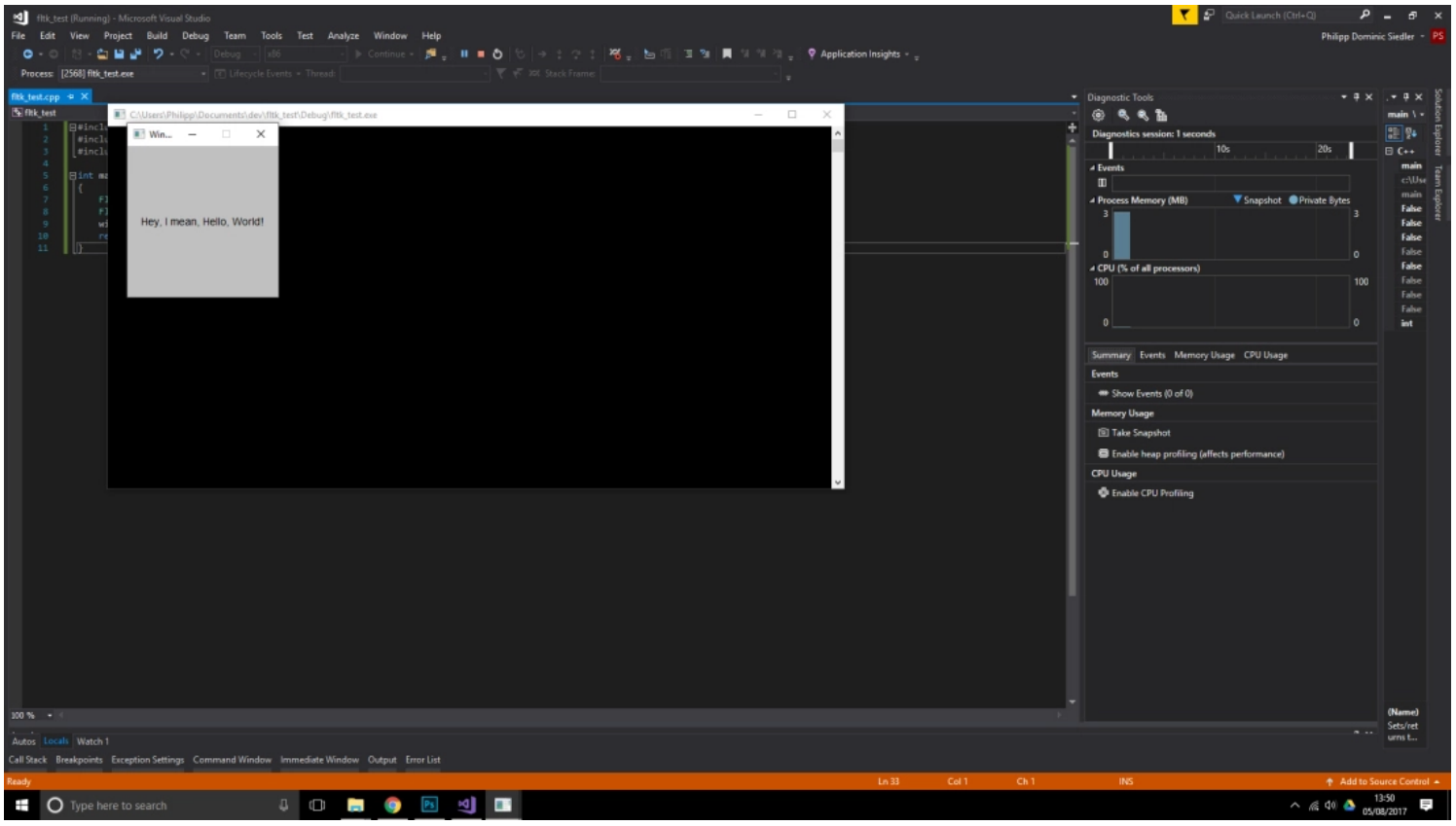


Now go to **line 28**, between # **include “Enumerations.H”** on line 27 and # **ifdef WIN32** and include # **define WIN32** like this:

…
 # include “Enumerations.H”
 # define WIN32
 # ifdef WIN32
…

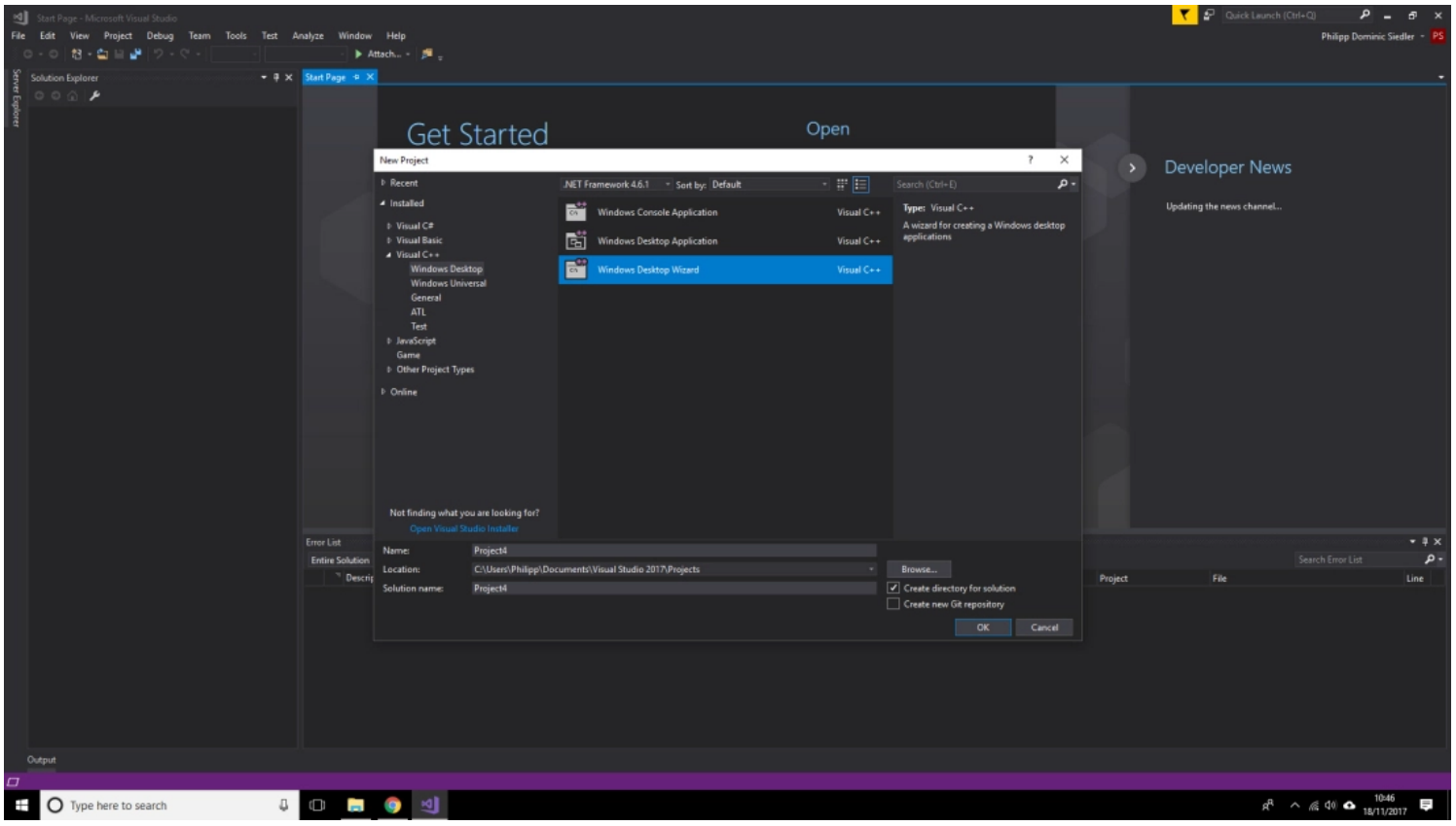


Safe the file and you should be good to go, now: Hit F5 to test-run the project and it should look like this:

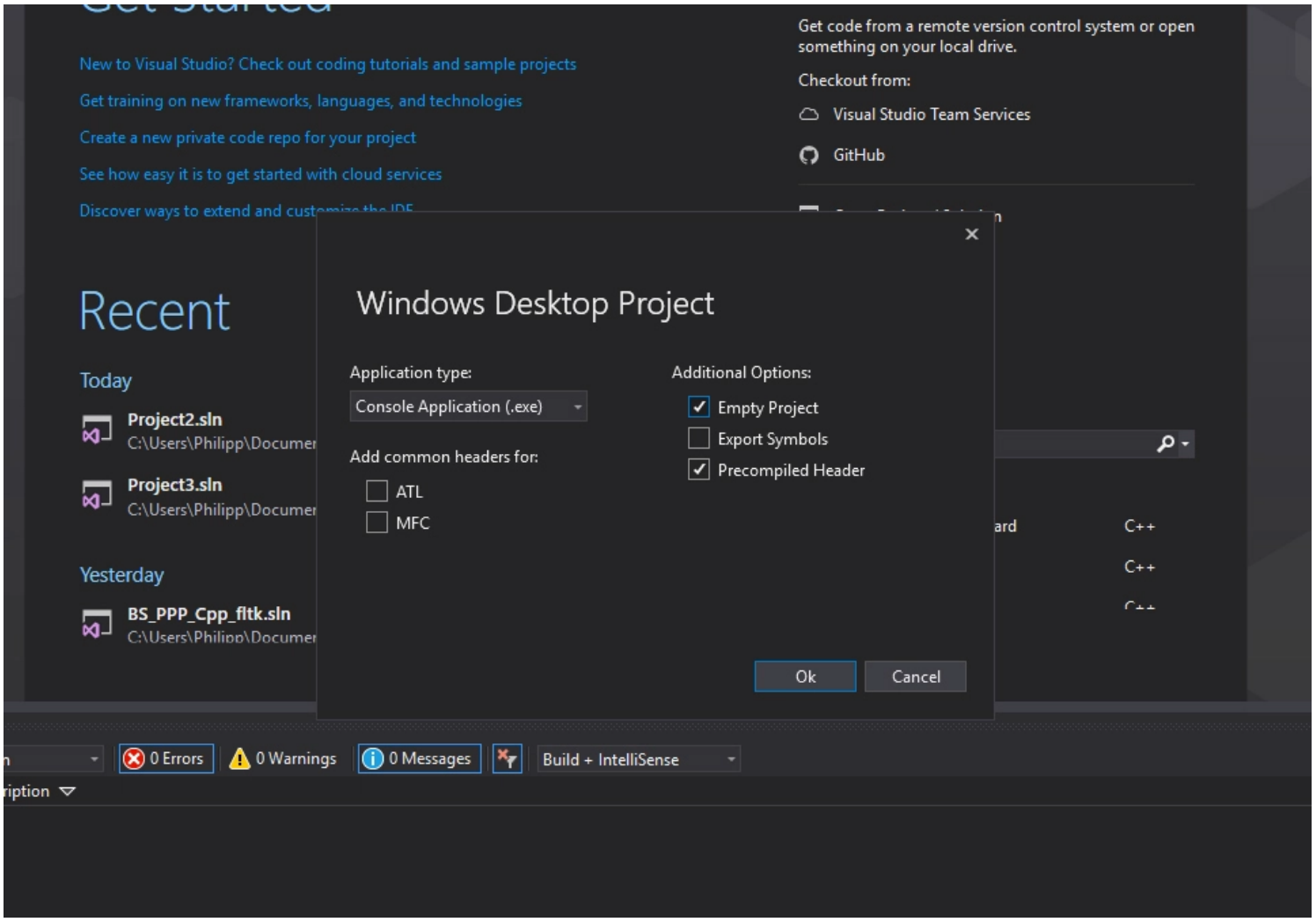


If you got to this point: **Congratulations, you made it!**
For who ever was interested in just getting the **fltk-1.3.4** GUI library running and applying it on a test project, this is the end of your journey and this guide. If you are working through Bjarne Stroustrup's Programming Principles and Practice Using C++ and you are in Chapter 12 starting to use fltk keep on reading for instructions on how to use it within the "Stroustrup"-environment:

Step 4: Using fltk-1.3.4 with Bjarne Stroustrup's header and .cpp files.
Safe your fltk test-project and create a new project. Under the **Visual C++ – Windows Desktop** tab create a new **Windows Desktop Wizard** project, just like you did before (I will go through this a bit faster as you should be familiar by now).

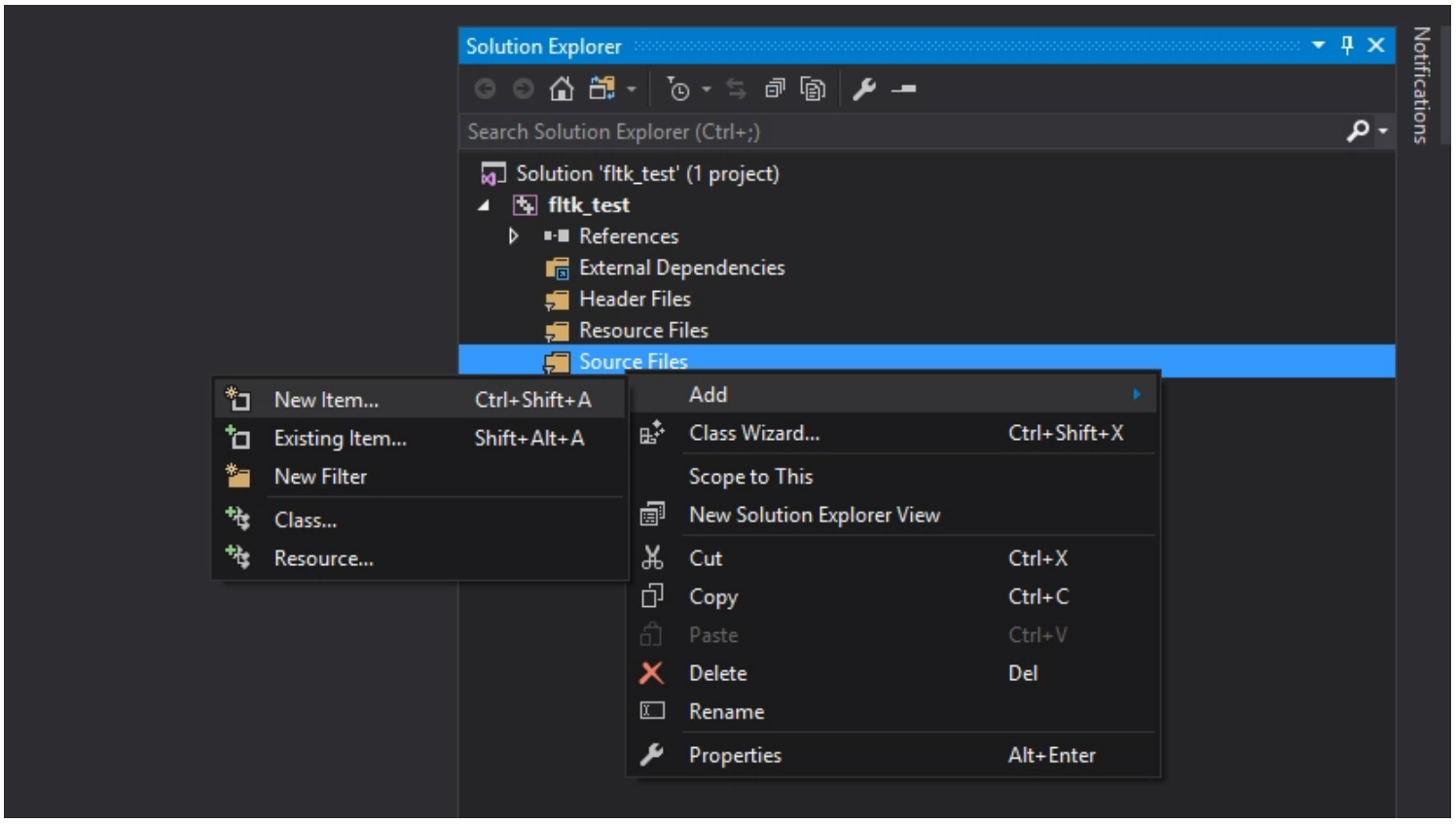

Get Started
Open
Developer News
New Project
Recent
Installed
Visual C#
Visual Basic
Visual C++
Windows Desktop
Windows Universal
General
ATL
Test
JavaScript
Game
Other Project Types
Online
.NET Framework 4.6.1
Sort by: Default
Windows Console Application
Windows Desktop Application
Windows Desktop Wizard
Visual C++
Type: Visual C++
A wizard for creating a Windows desktop applications
Not finding what you are looking for?
Open Visual Studio Installer
Name: Project4
Location: C:\Users\Philipp\Documents\Visual Studio 2017\Projects
Solution name: Project4
Browse...
Create directory for solution
Create new Git repository
OK
Cancel

Tick the box **Empty project** (very important).



I called mine BS_PPP_Cpp_fltk.

Right-click the **Source Files** folder in your **Solution Explorere** and **Add** a **New Item**. Create a new .cpp file, I called mine **p437_d_1_.cpp**.



Copy and paste the following code from Bjarne Stroustrup's Programming Principles and Practice Using C++, Drill 1 page 437:

```
#include "Graph.h"
#include "Simple_window.h"

int main()
{
    using namespace Graph_lib;

    Point tl(150, 150);
    Simple_window win(tl, 600, 400, "My window");
    win.wait_for_button();
}
```

Go to **Project** and **Properties** and setup your project in the exact same way we did with our fltk test-project above.

Now this time, we have to add a couple of header files from Bjarne Stroustrup, and also some additional .cpp files.
Let's begin with the **header files**:
Most of the people reading, working through Programming Principles and Practice Using C++ included the **std_lib_facilities.h** already many times so for who ever does not know where to get it from, this is the link (http://www.stroustrup.com/Programming/PPP2code/). Download it and copy it in your project folder (in my case: C:\Users\Philipp\Documents\Visual Studio 2017\Projects\BS_PPP Cpp_fltk\BS_PPP_Cpp_fltk\BS_PPP_Cpp_fltk).
As well as the following header files:

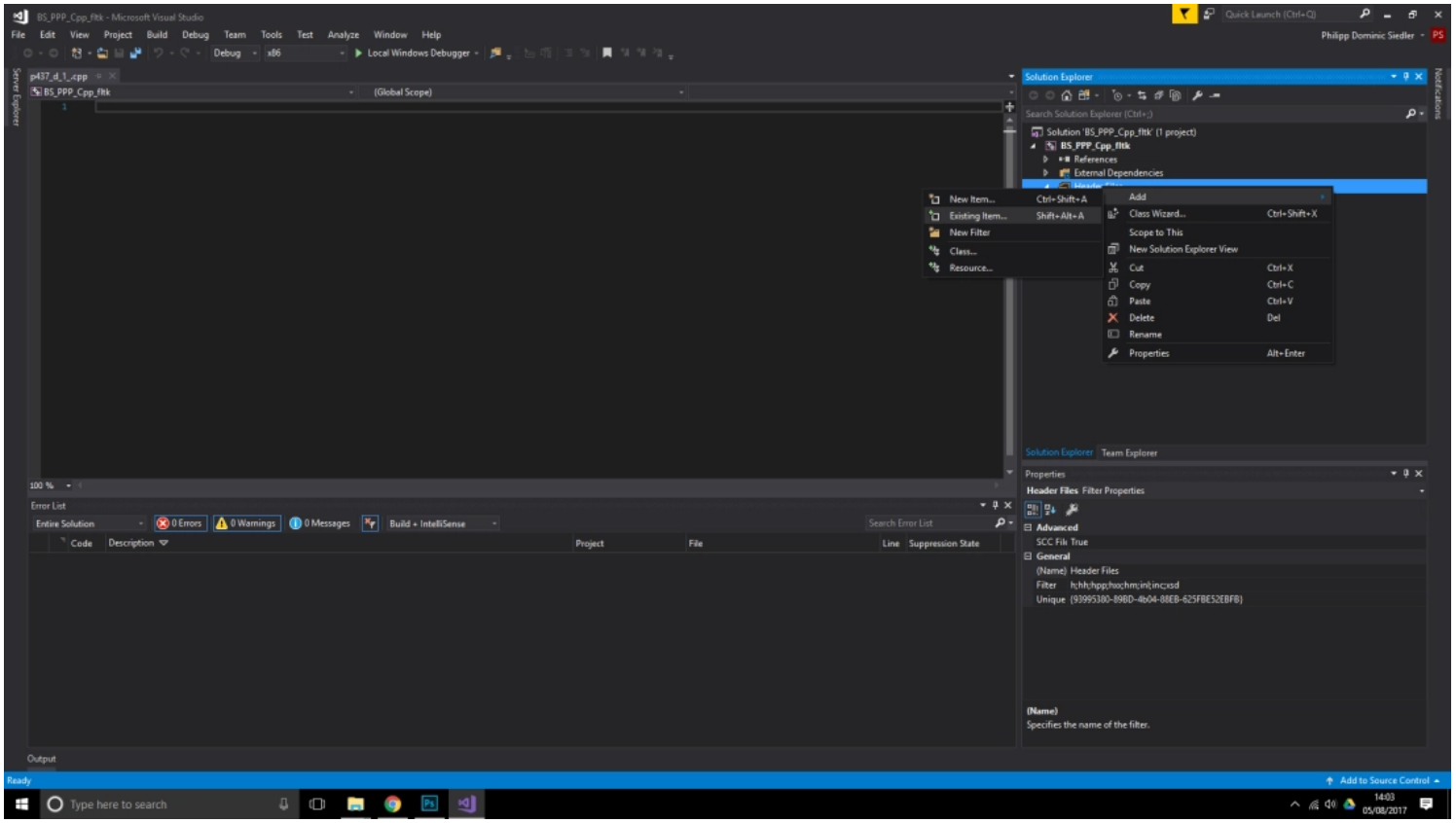
std_lib_facilities.h

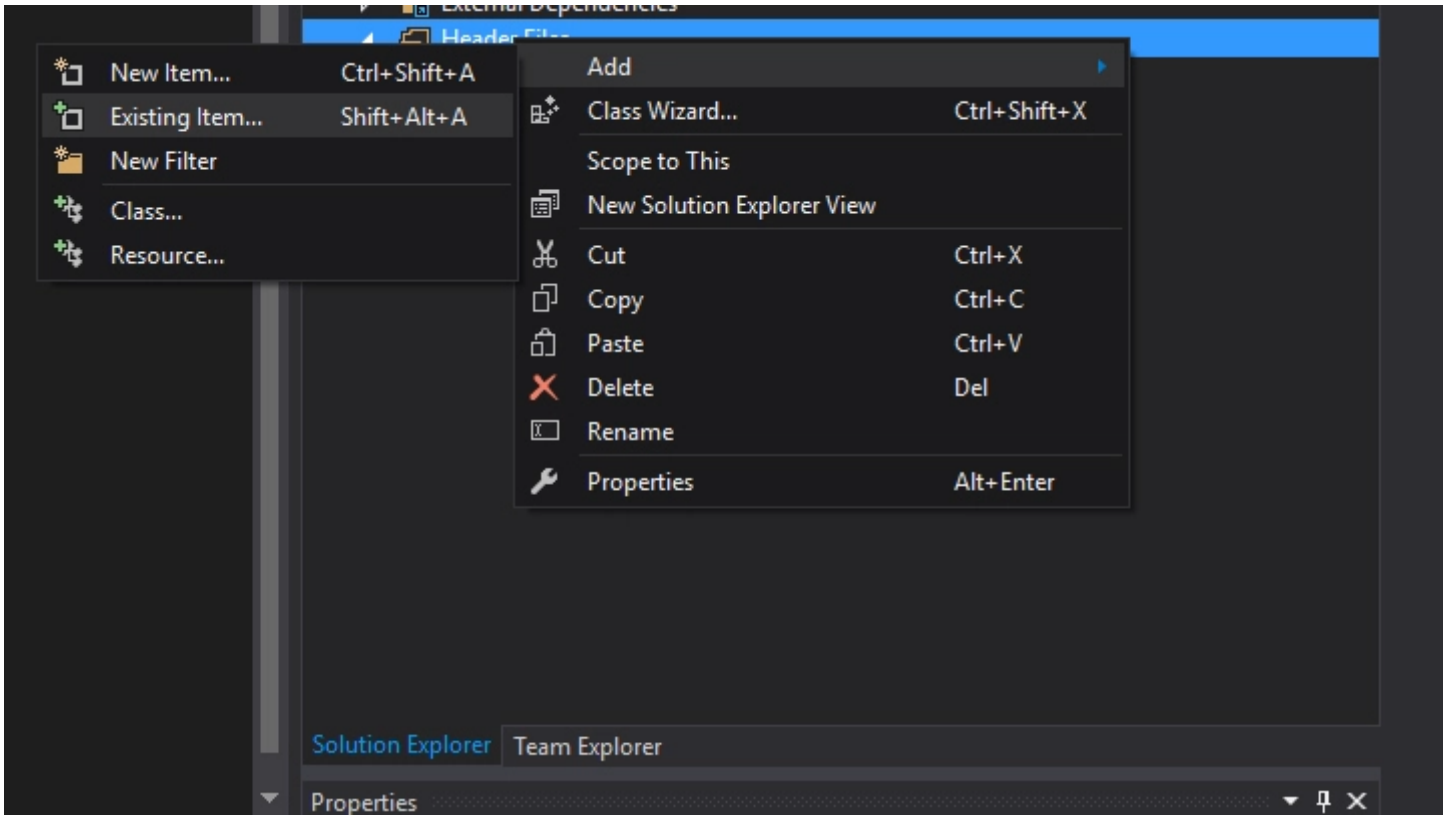
fltk.h
Graph.h
GUI.h
Point.h
Simple_window.h
Window.h

and while your on it also download and copy the following .cpp files into your project directory (as mentioned above):

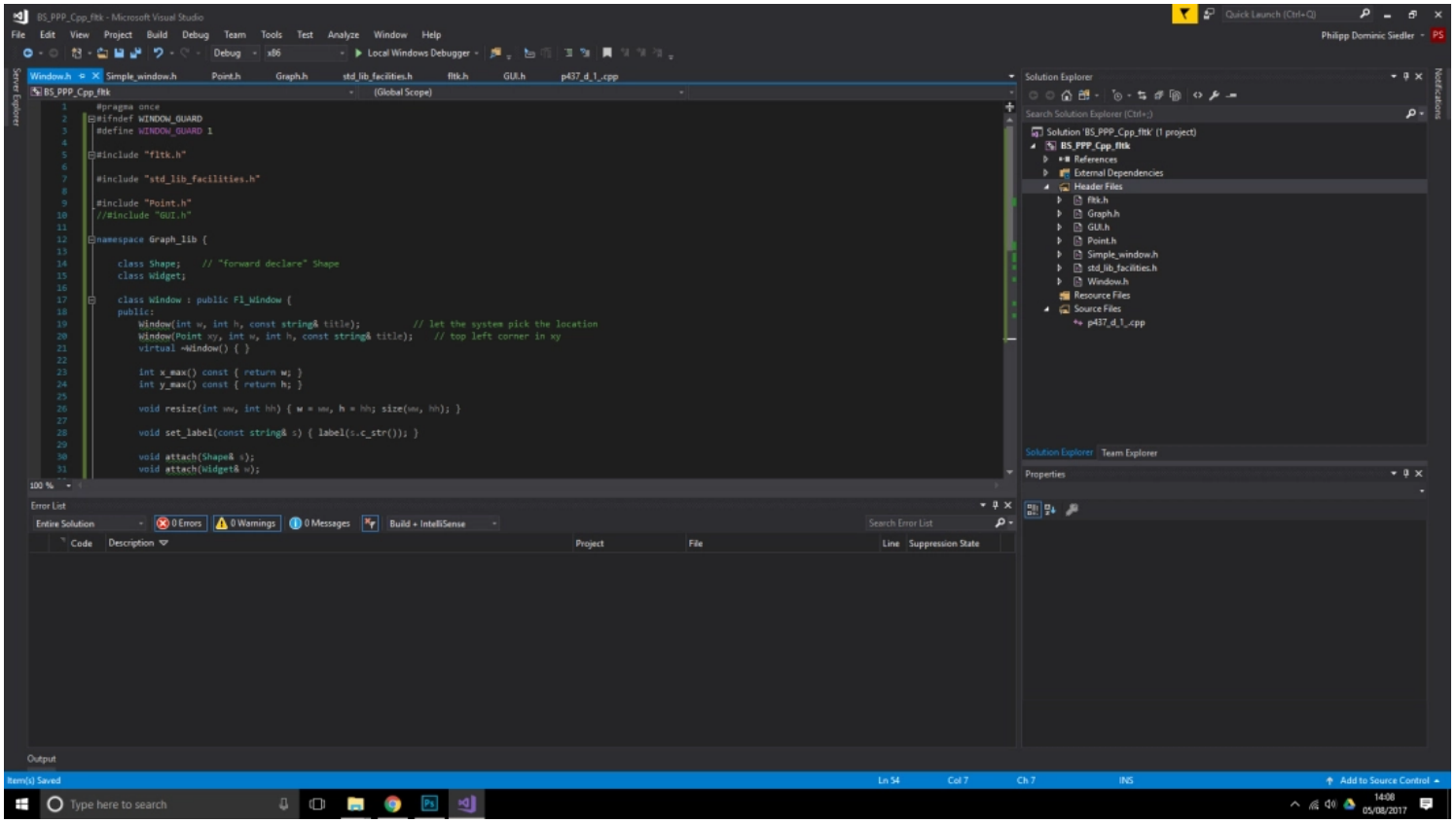
Graph.cpp
GUI.cpp
Window.cpp

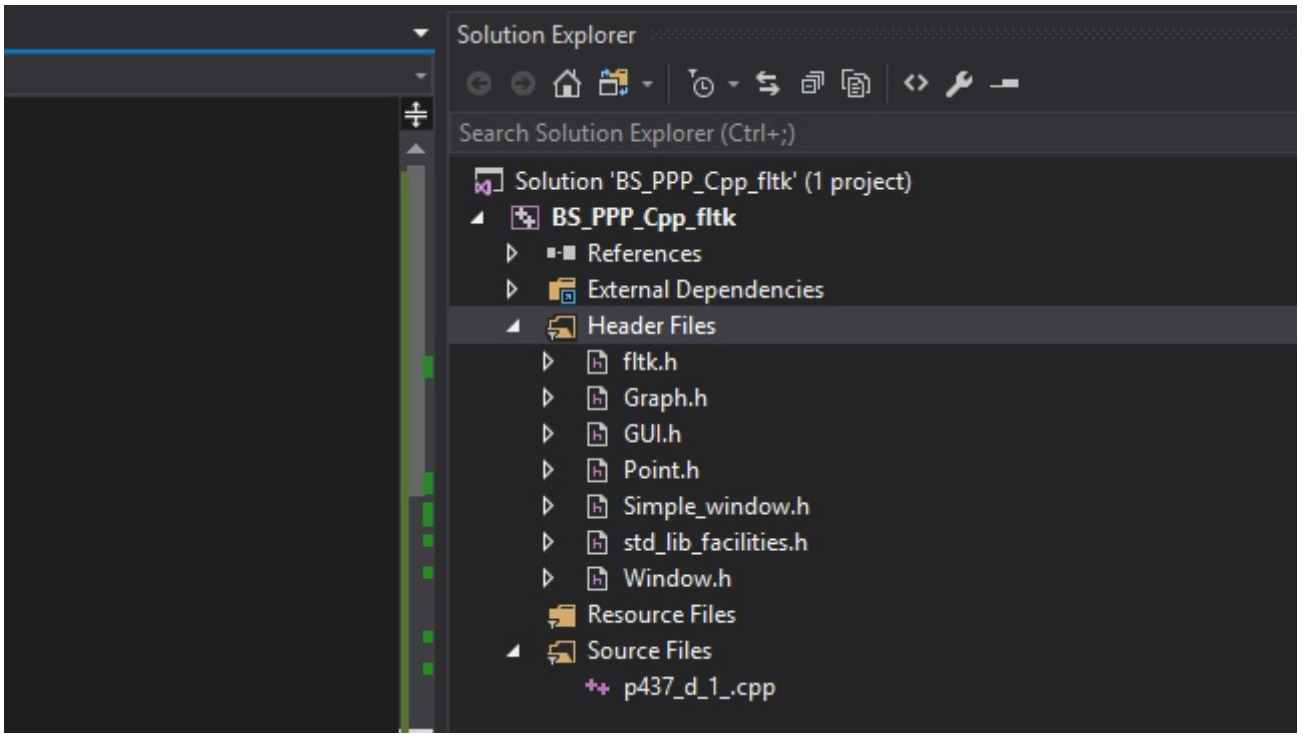
Right-click the **Header Files** folder in your project's solution explorer. **Add** an **Existing Item**.



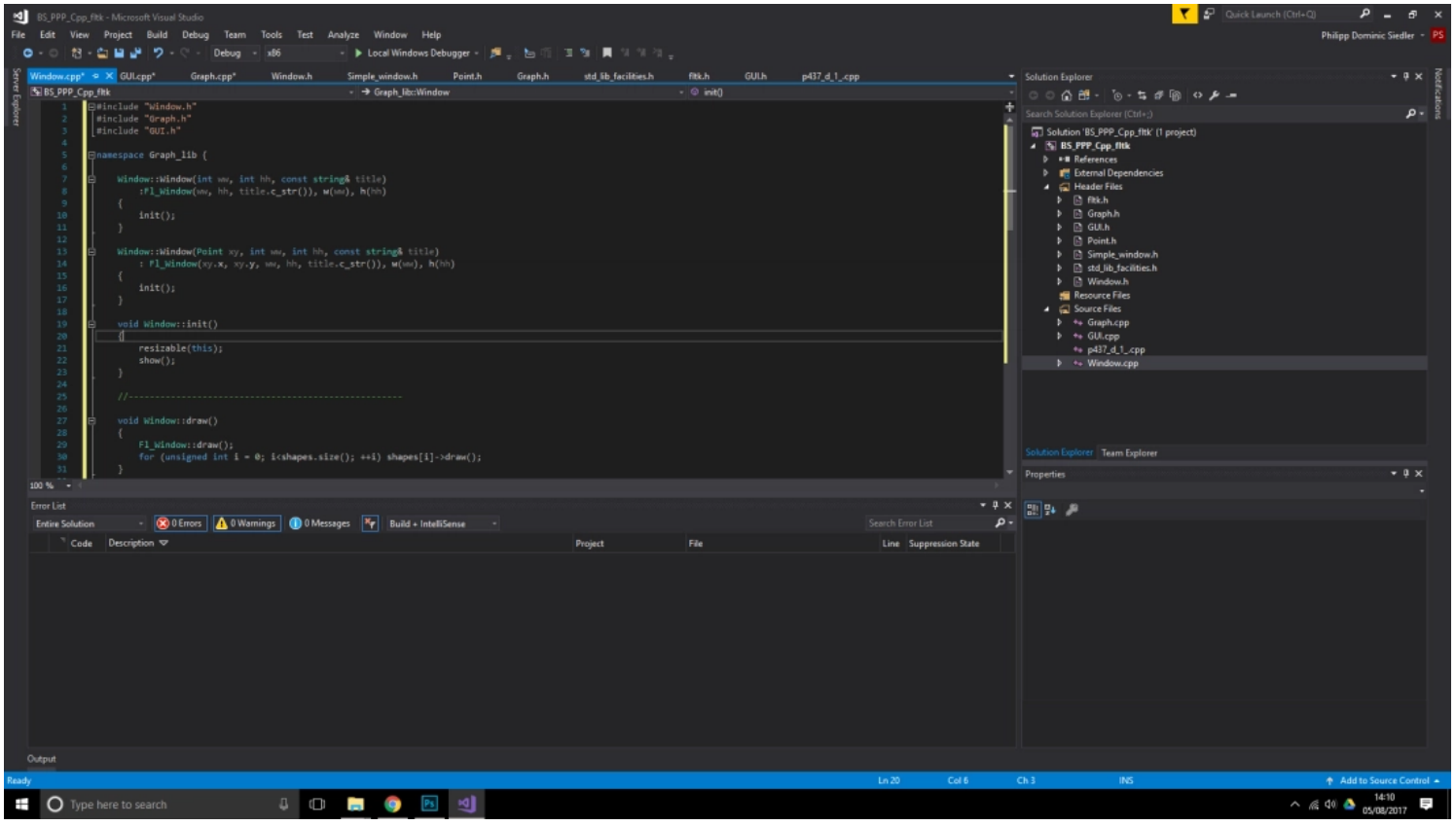


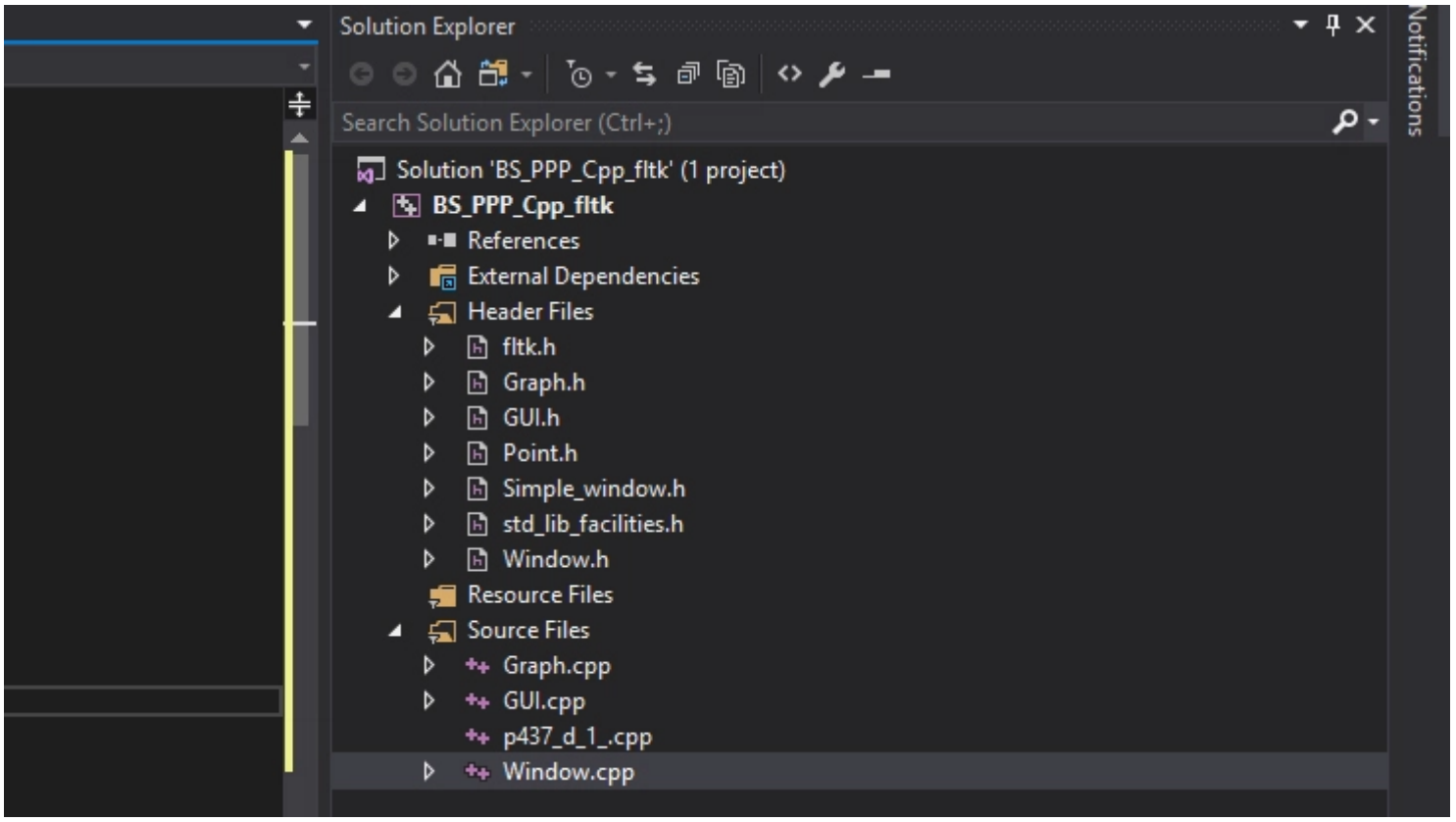
Once you added all your header files the **Header Files** folder should look like this:



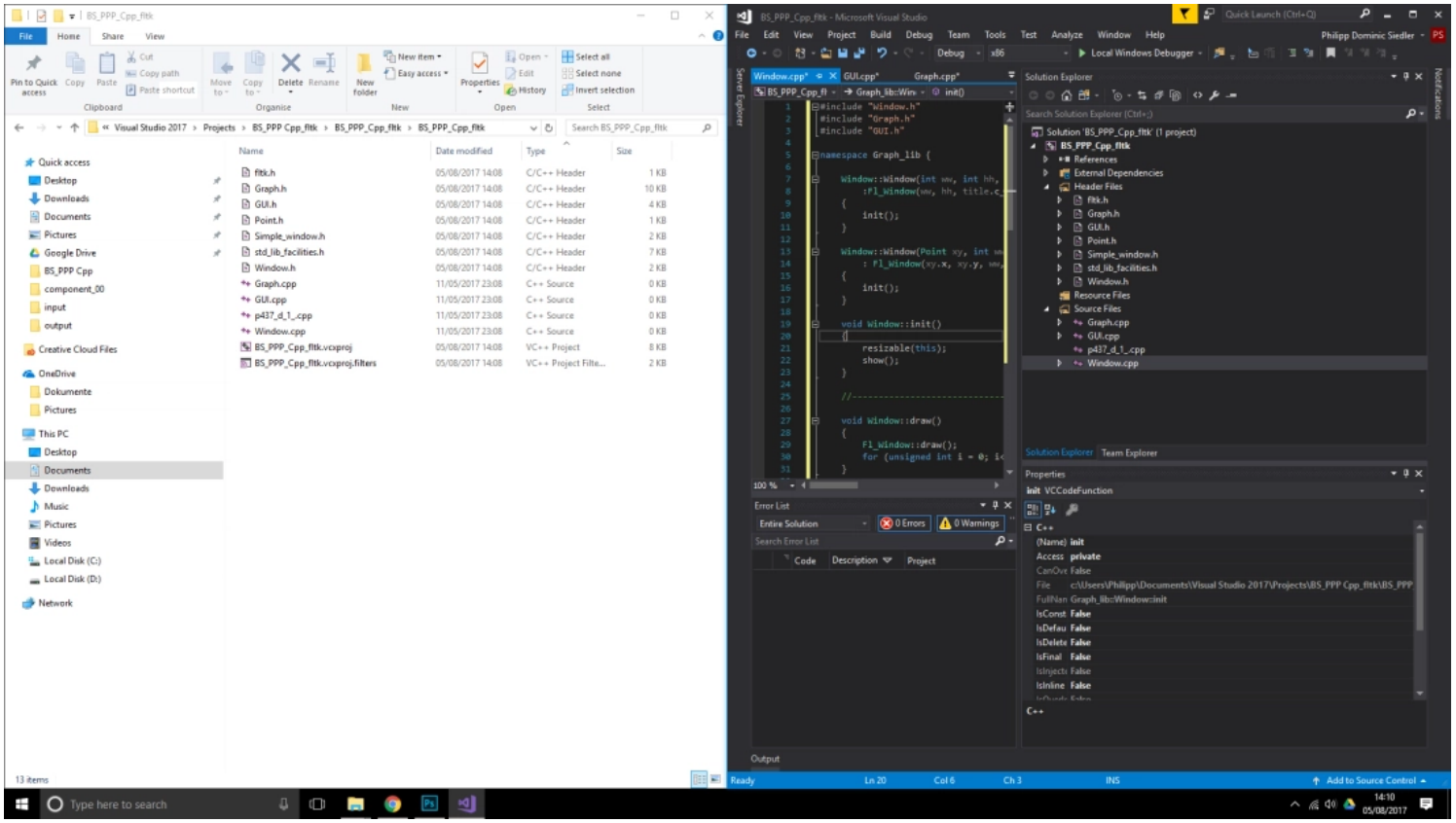


Now also add the .cpp files: **Graph.cpp**, **GUI.cpp** and **Window.cpp** to your **Source Files** folder. Your Source Files folder should look like this:



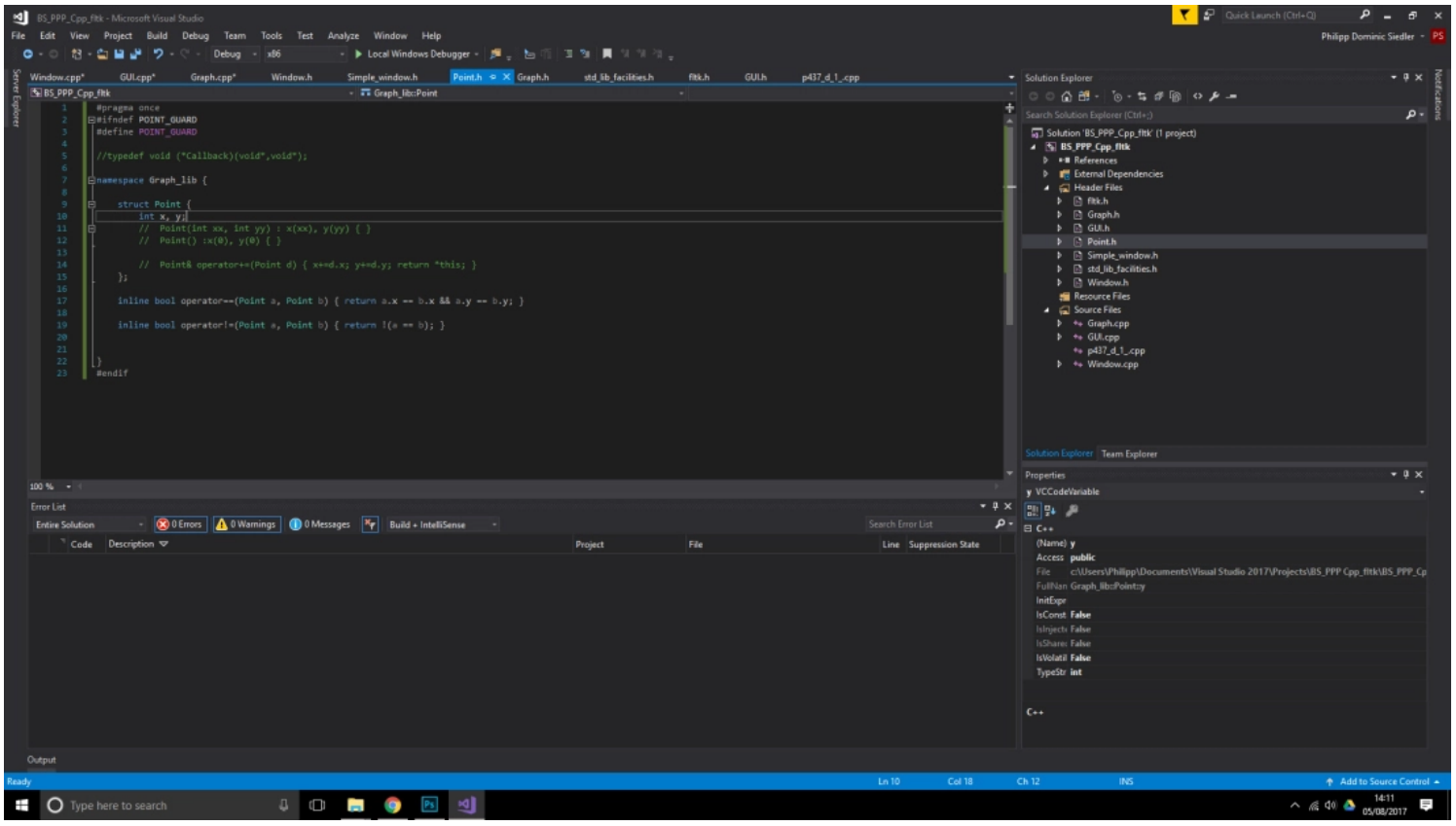


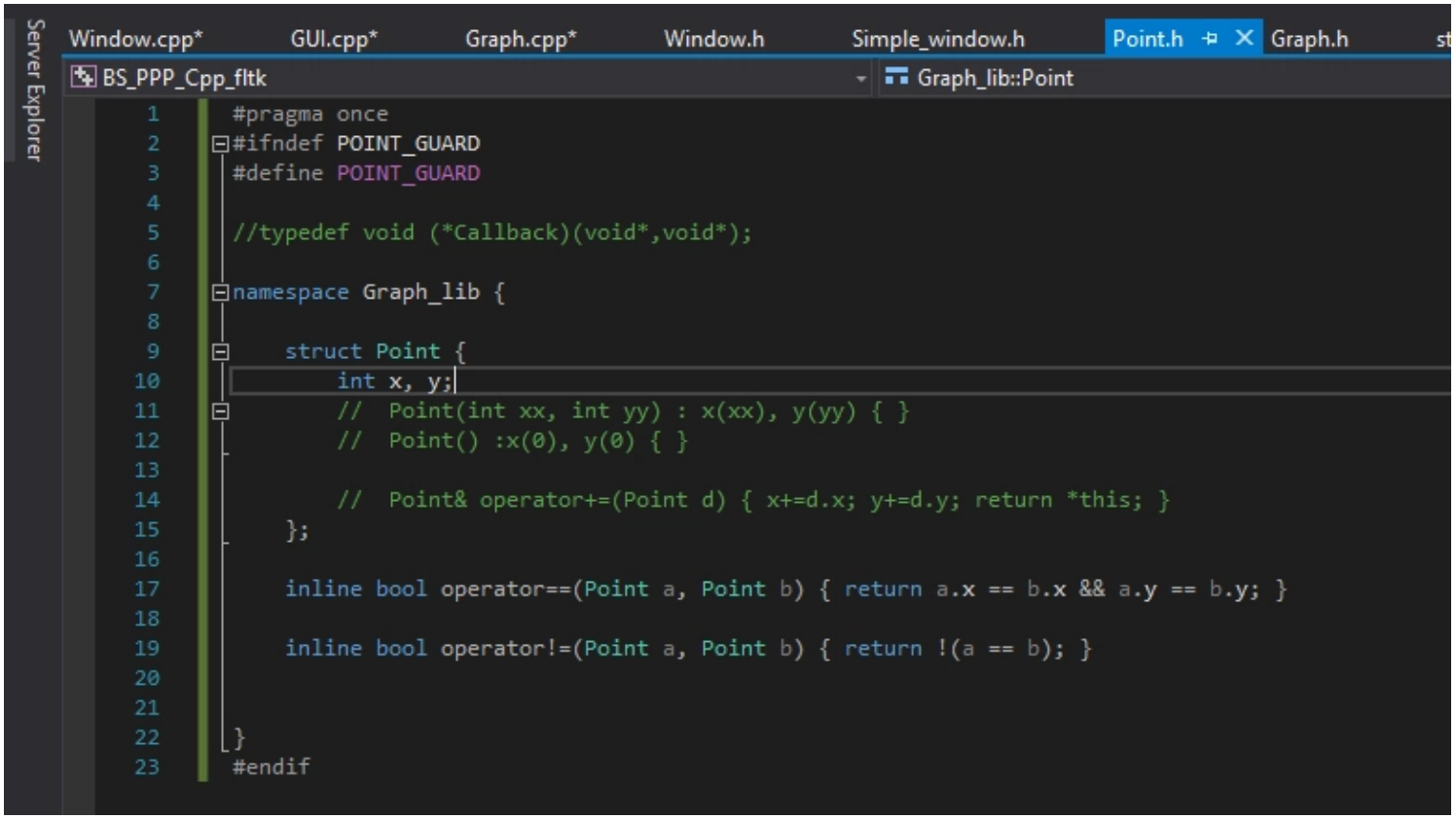
Your project directory folder and your **Solution Explorer** in Visual Studio 2017 should look like this:



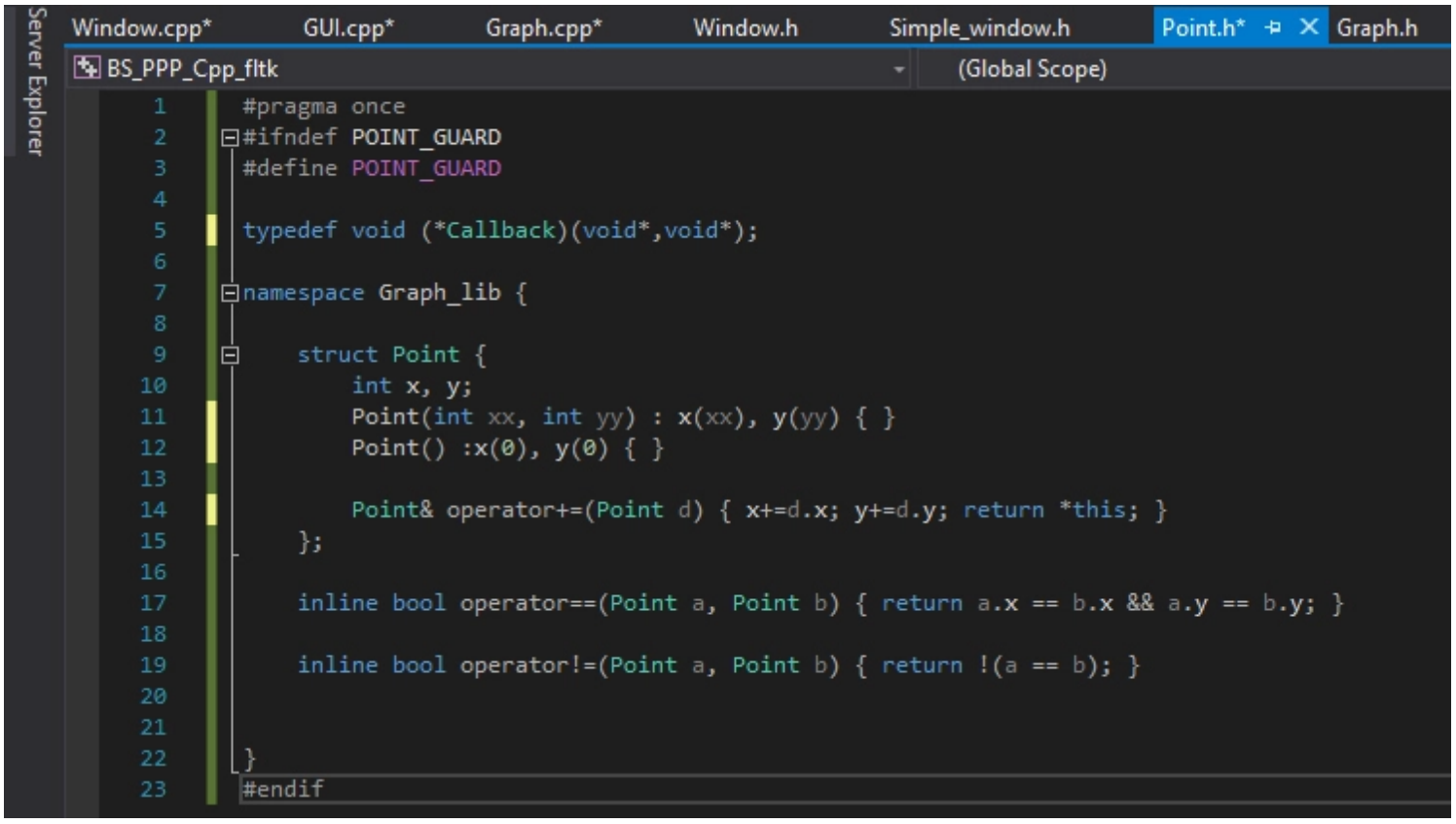
Now you are almost there. We just have to make a couple of changes to those Added files.

Open the **Point.h** header file by double clicking it:

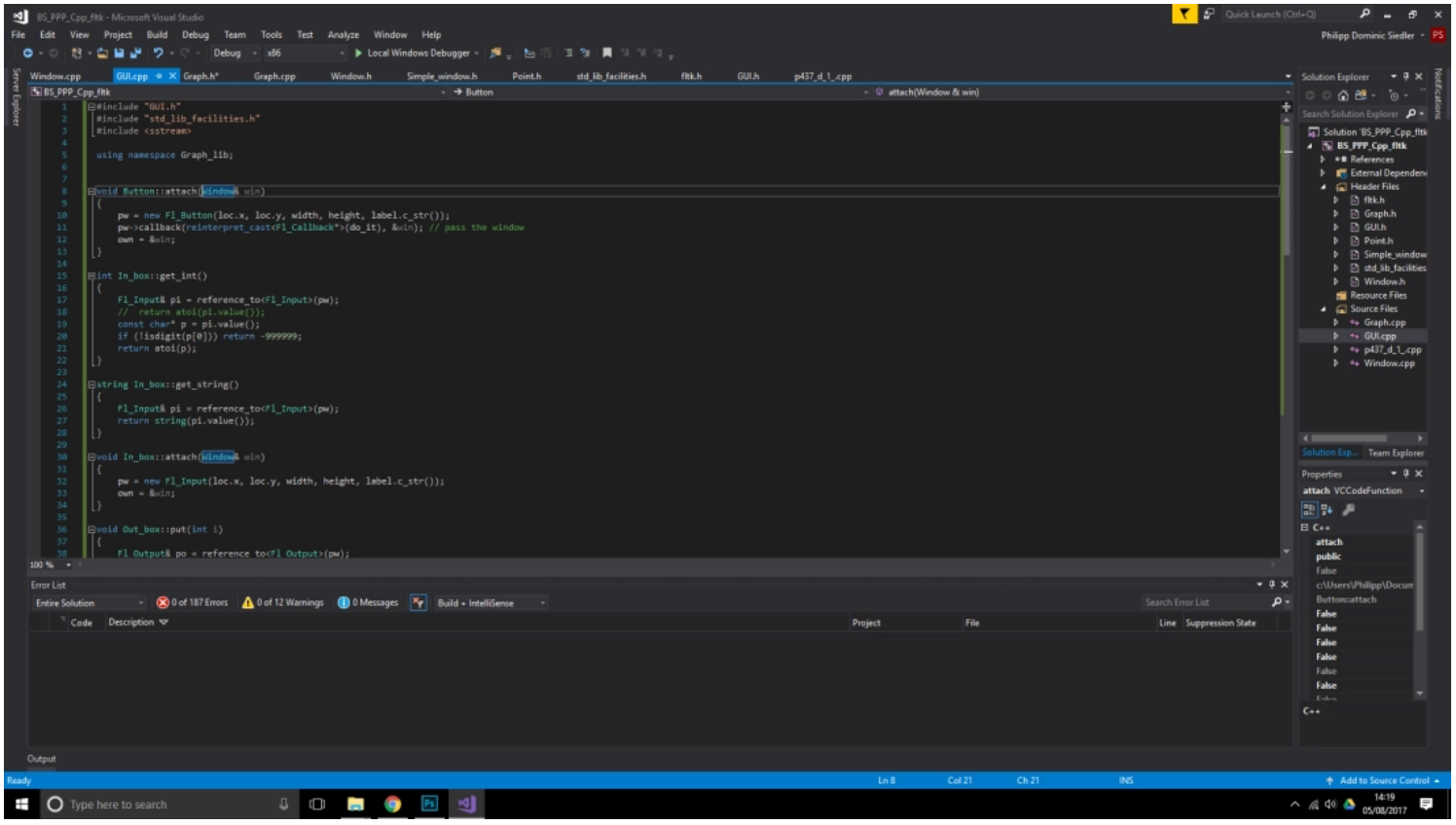




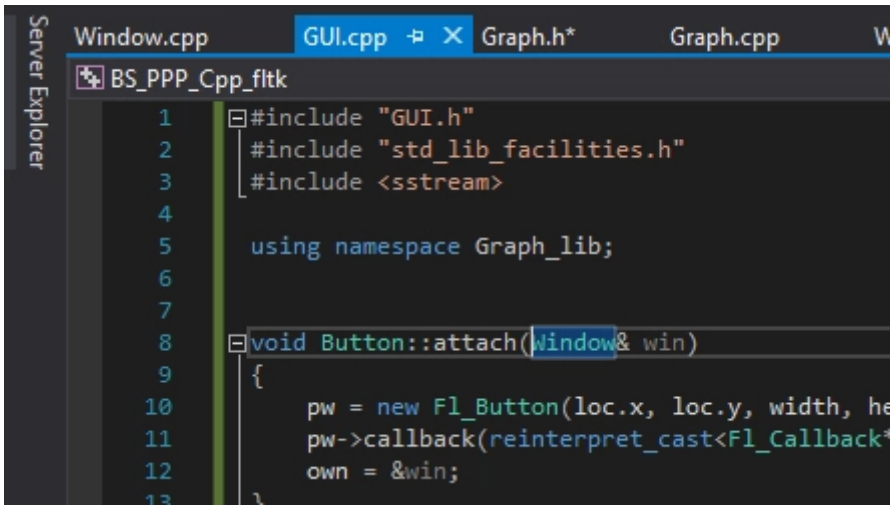
Remove all the notation double forward-slashes:



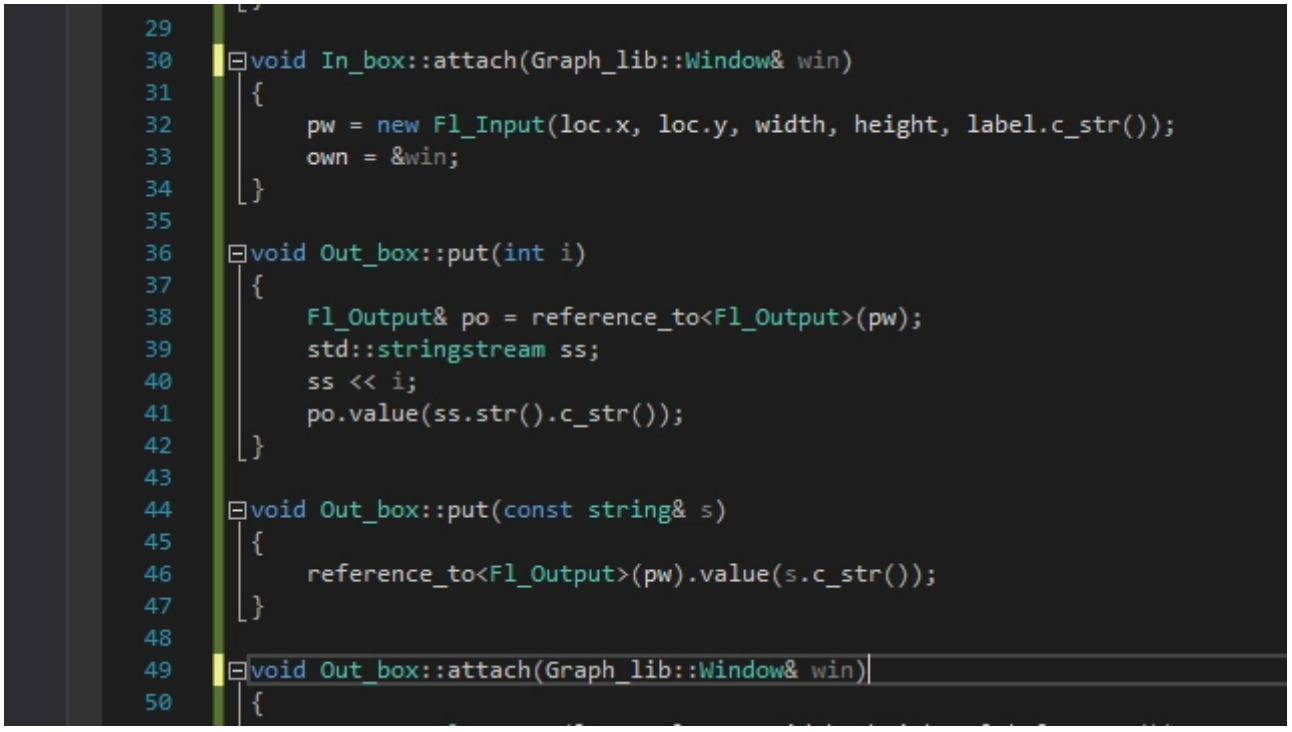
Open the **GUI.cpp** file by double clicking it:



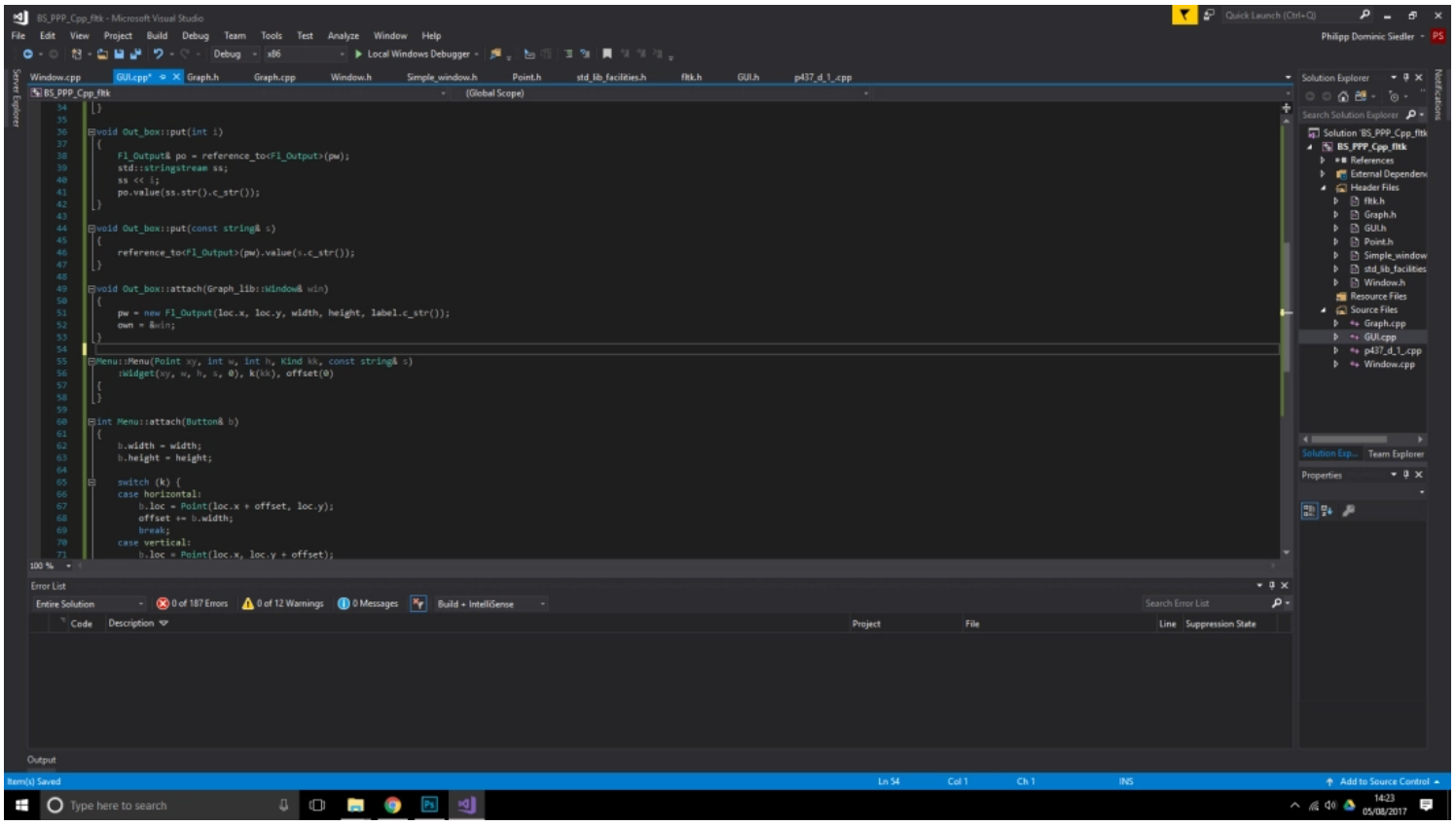
Add **Graph_lib::** in front of **Window&** of **Button::attach** (line 8)



In_box::attach (line 30) and **Out_box::attach** (line 49).

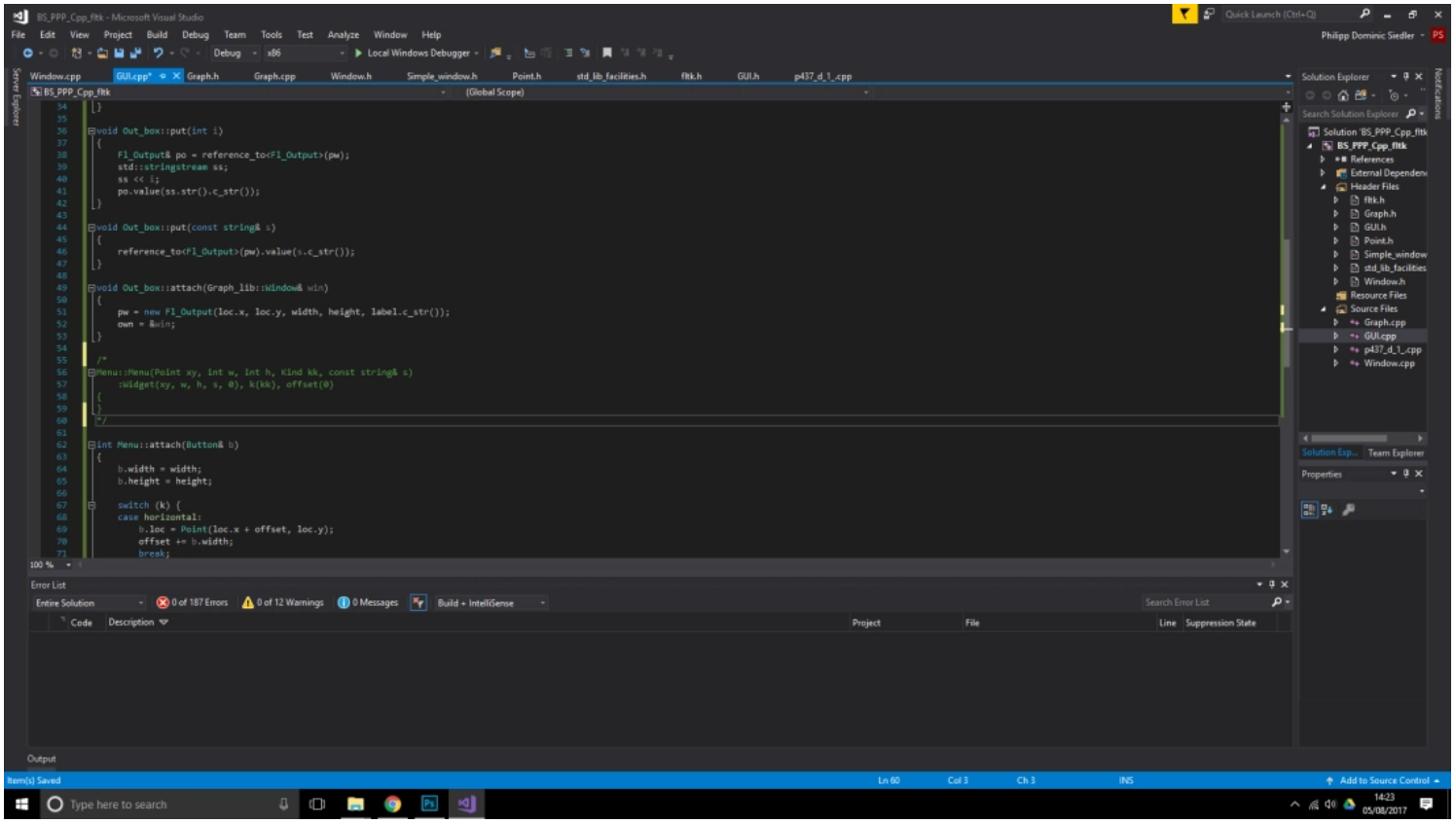


and remove or notate the constructor Menu::Menu as it is already defined in GUI.h.

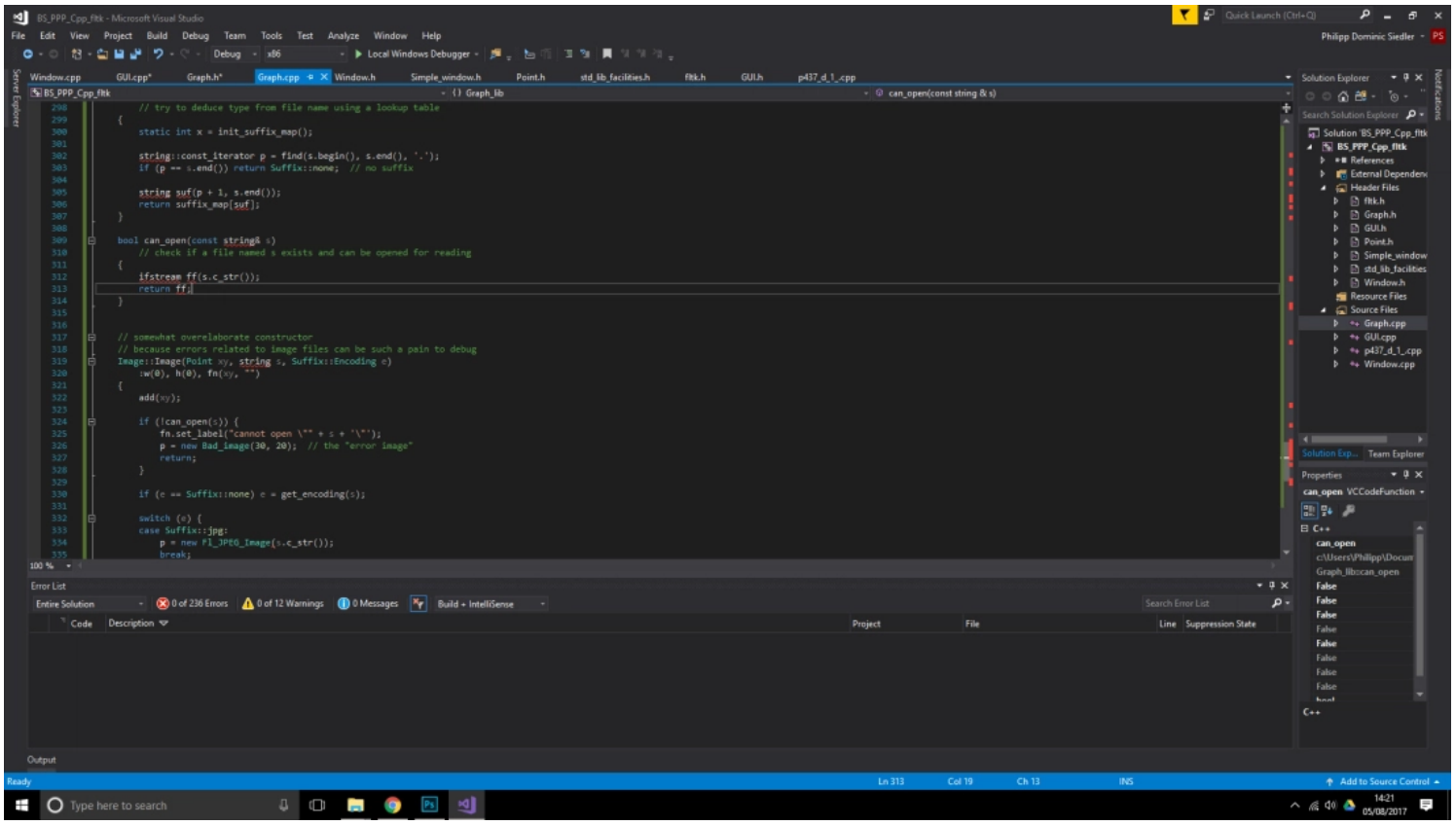


```
45      {
46          reference_to<Fl_Output>(pw).value(s.c_str());
47      }
48
49      void Out_box::attach(Graph_lib::Window& win)
50      {
51          pw = new Fl_Output(loc.x, loc.y, width, height, label.c_str()
52          own = &win;
53      }
54
55      Menu::Menu(Point xy, int w, int h, Kind kk, const string& s)
56          :Widget(xy, w, h, s, 0), k(kk), offset(0)
57      {
58      }
59
60      int Menu::attach(Button& b)
61      {
62          b.width = width;
63          b.height = height;
64
65          switch (k) {
```

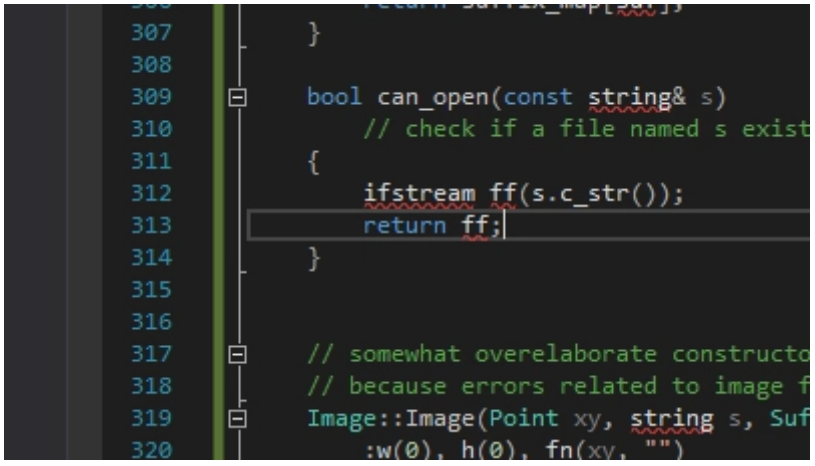
```
47  }
48
49  void Out_box::attach(Graph_lib::Window& win)
50  {
51      pw = new Fl_Output(loc.x, loc.y, width, height, label.c_st
52      own = &win;
53  }
54
55  /*
56  Menu::Menu(Point xy, int w, int h, Kind kk, const string& s)
57      :Widget(xy, w, h, s, 0), k(kk), offset(0)
58  {
59  }
60  */
61
62  int Menu::attach(Button& b)
63  {
64      b.width = width;
65      b.height = height;
66
```

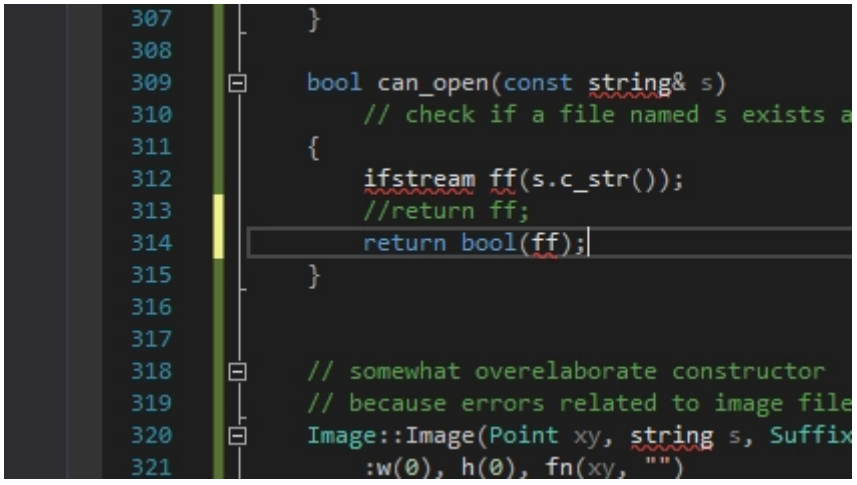


Open the **Graph.cpp** file:

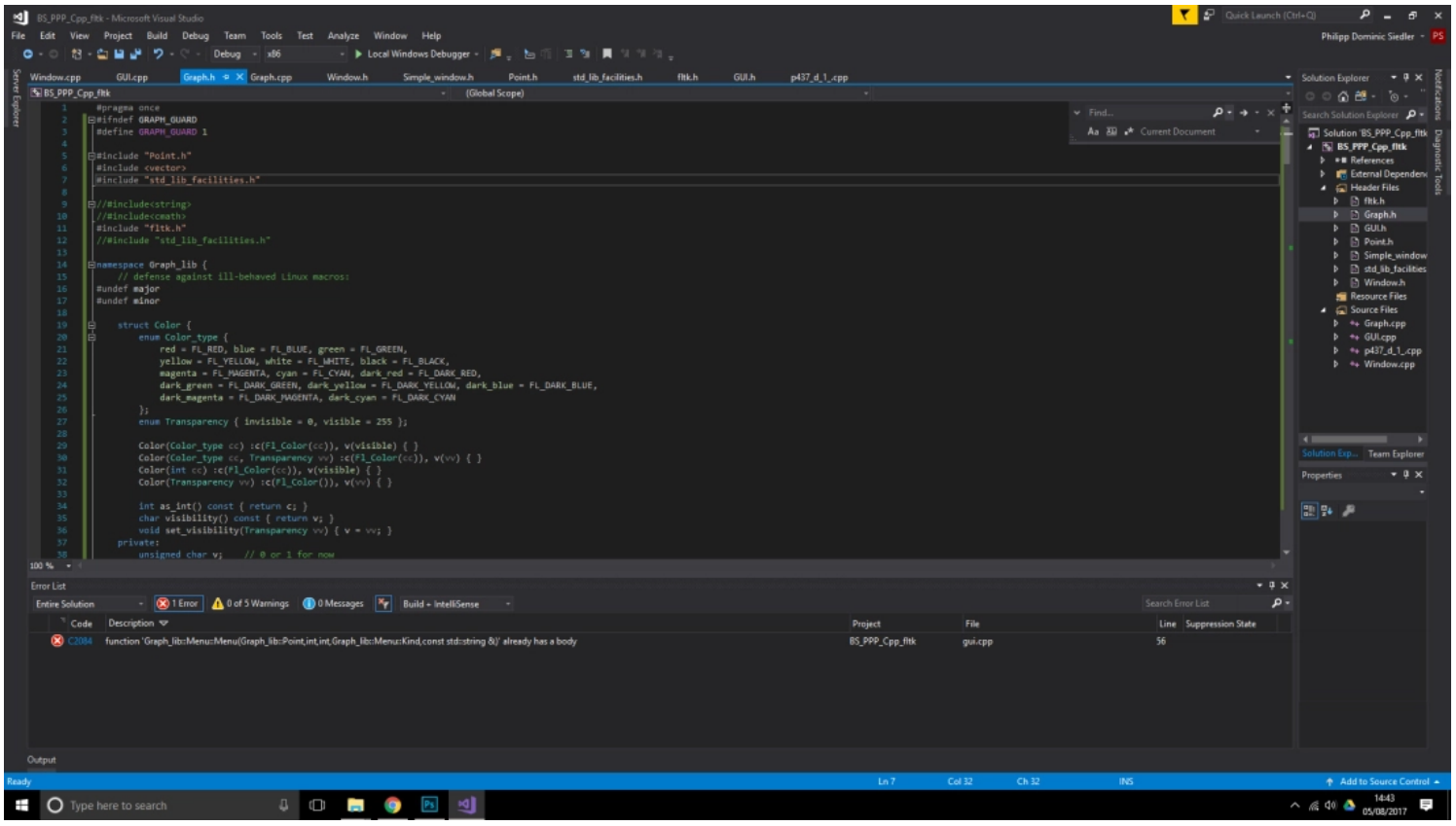


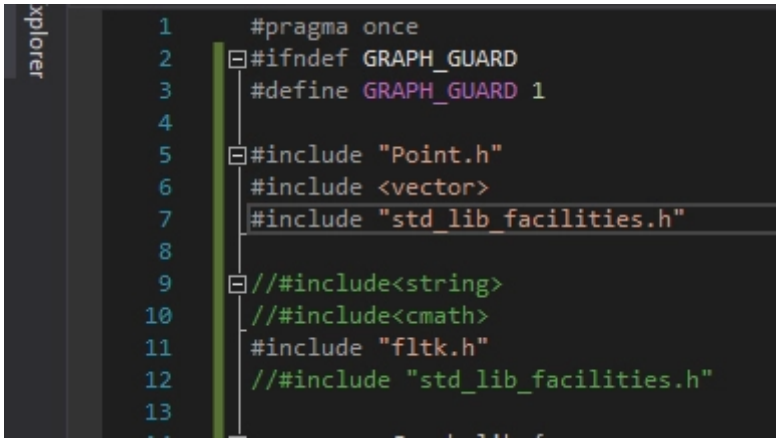
And go to line 313. Change **return ff;** to **return bool(ff);**



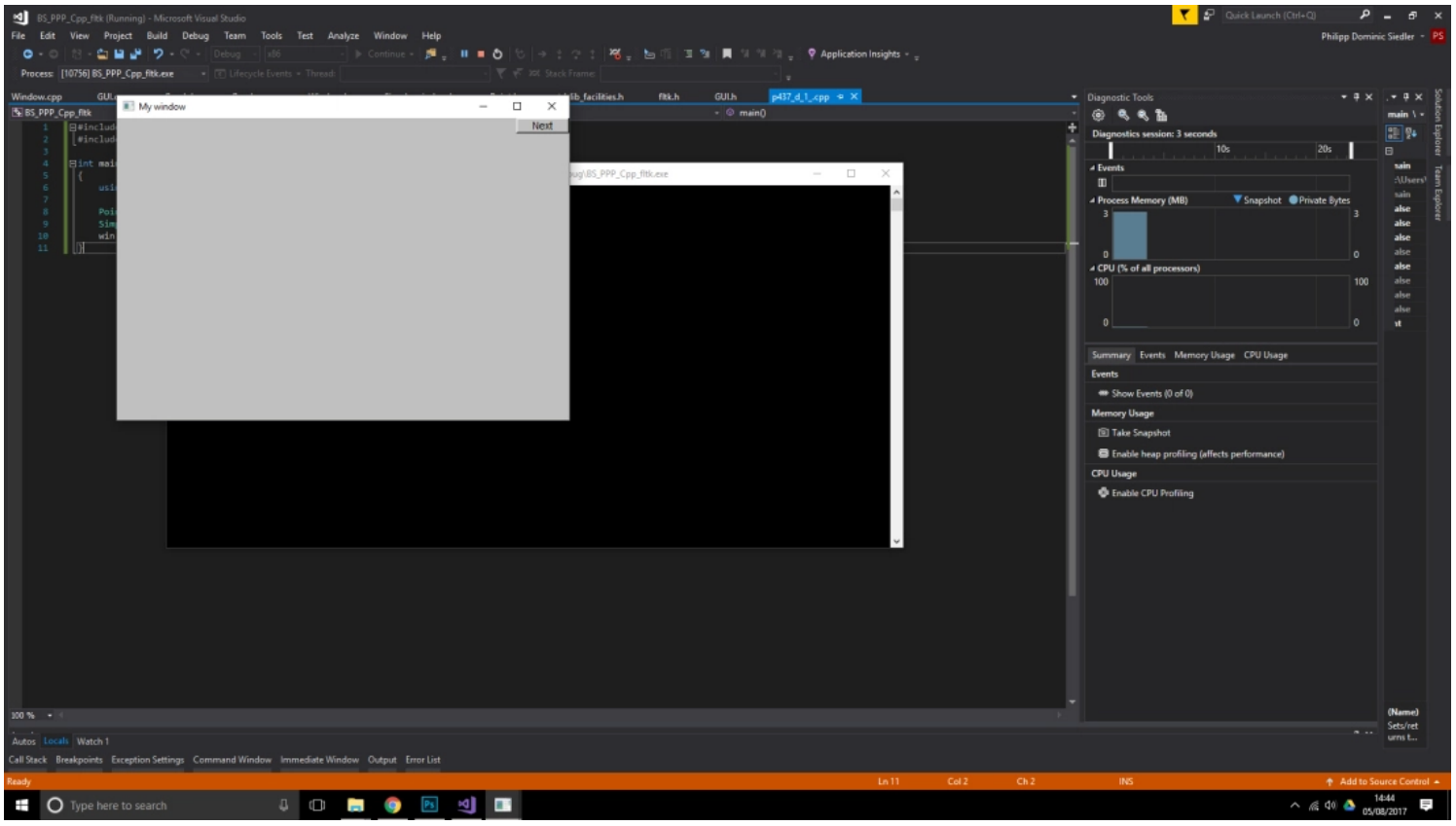


The last change you have to make is in **Graph.h**, open and **#include "std_lib_facilities.h"**:





This is it. Now safe all your files and run your project. I like to just hit **F5**.
If everything goes right and you were able to follow my instructions you should be able to see the following:



CONGRATULATION, ALL DONE!
If you made it so far and everything worked out I am very happy I could help you. At this point I should mention Benjamin Wuethrich. I followed his tutorial on installing and using fltk on Visual Studio 2015 and with the help of a bunch of other peoples forum posts and other informations I was able to transfer the knowledge to make it happen on Visual Studio 2017 Community. So thanks to Benjamin Wuethrich. His tutorial for Visual Studio 2015 can be found here (https://bewuethr.github.io/installing-fltk-133-under-visual-studio/).

Thank you for going through this with me, if you have any questions I am very happy to help. Also if you have any suggestions for improvement, please let me know.

Tagged:
Bjarne Stroustrup,
C++,
FLTK,
Principles and Practice Using C++

Published by bumpyroadtocode

View all posts by bumpyroadtocode