**Building Winsock Project Using Microsoft Visual Studio 2017**

Preparation: Install Microsoft Visual Studio Community 2017 from

* 1. <https://www.visualstudio.com/zh-hans/downloads/>

1. **Creating a Windows Socket project**

Now let us construct an executable using the Visual C++ development environment.

**Making the Project**

Setting up a project is the first step in working with Windows sockets. There are several important steps in creating a Windows socket project. First, open Microsoft Visual Studio and click on *File* and then *New*, *Project* (Figure 1).

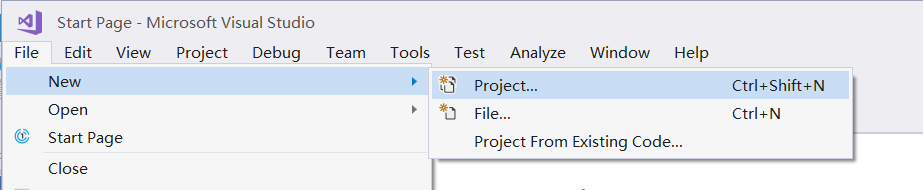


Figure 1

When the New window pops up (Figure 2), do the following:

1) select *Windows Console Application*

2) type a name in the project **Name** field

3) pick a location to store your project

4) click *OK*.

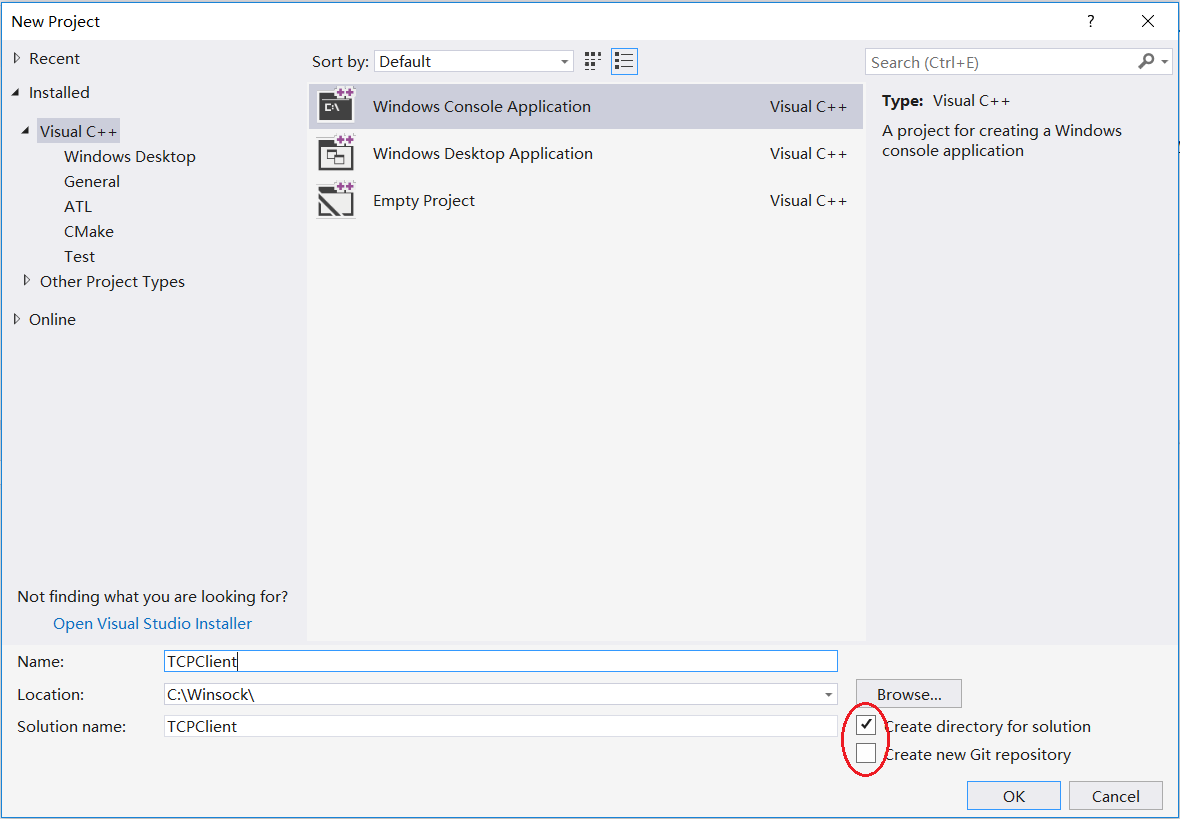


Figure 2

You can see the information of your project by clicking on *View* then *Solution Explorer*. (Figure 3)

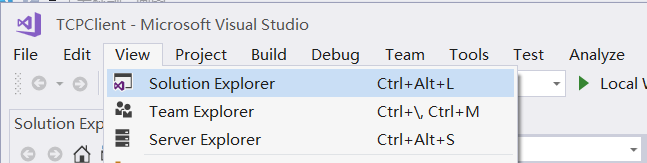


Figure 3

A Solution Explorer window will pop up, and it would be convenient to drag the Solution Explorer window to the left panel like in Figure 4.

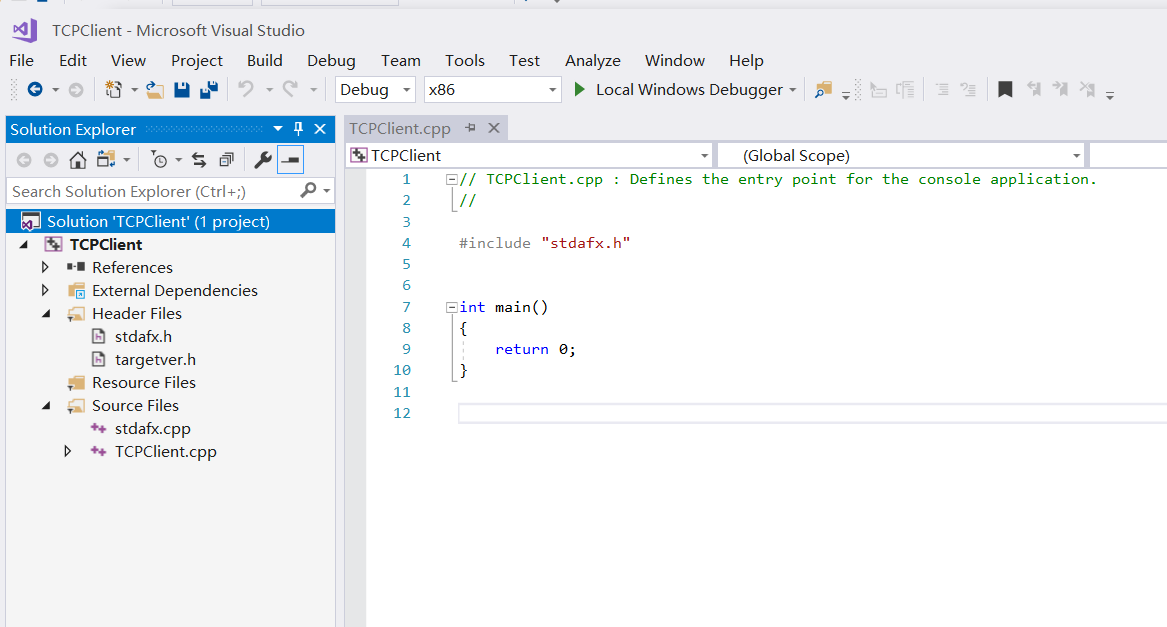


Figure 4

1. **Adding Windows Sockets Library to the Project**

Now in the Solution Explorer window, right click on your project name *(TCPClient* in this example)and then click *Properties*. (Figure 5)

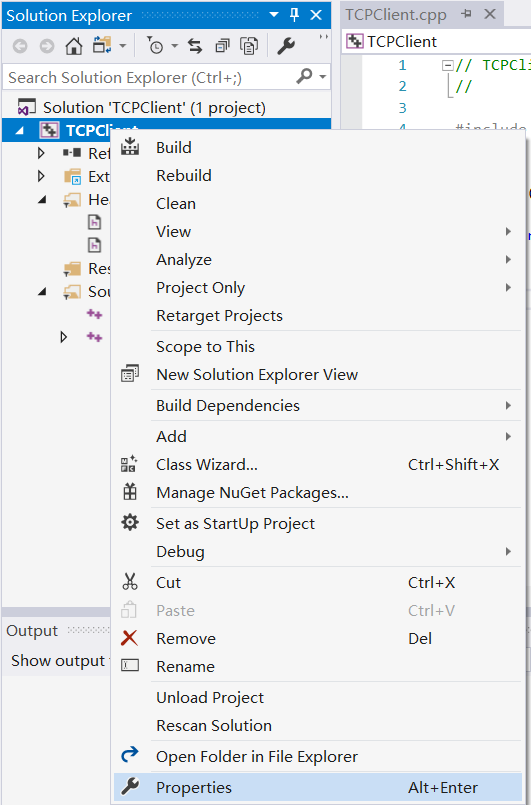


Figure 5

When the Property Pages window pops up (Figure 6), double click *Configuration Properties*, choose *Linker→* *Input* on the left panel, find *Additional Dependencies* on the right panel, and add *Ws2\_32.lib* to the list, click *OK*.

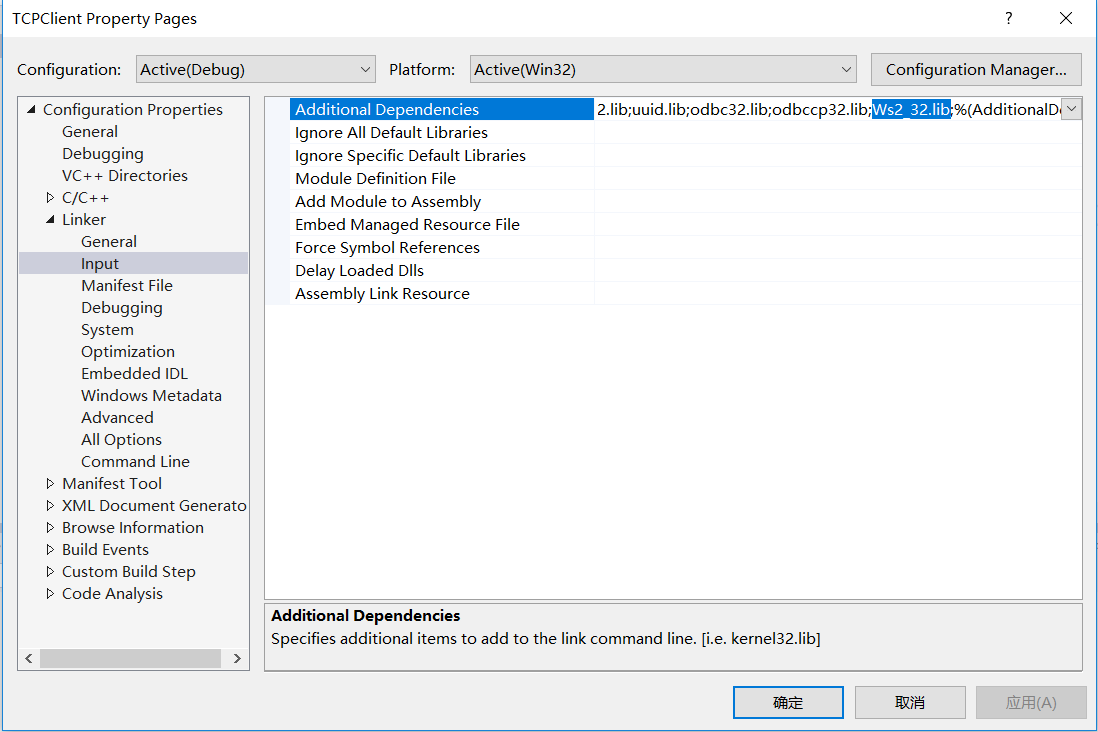


Figure 6

1. **Editing/adding source files to the Project**

An empty source file is created by default. You can add your own code to the file. (Figure 7)

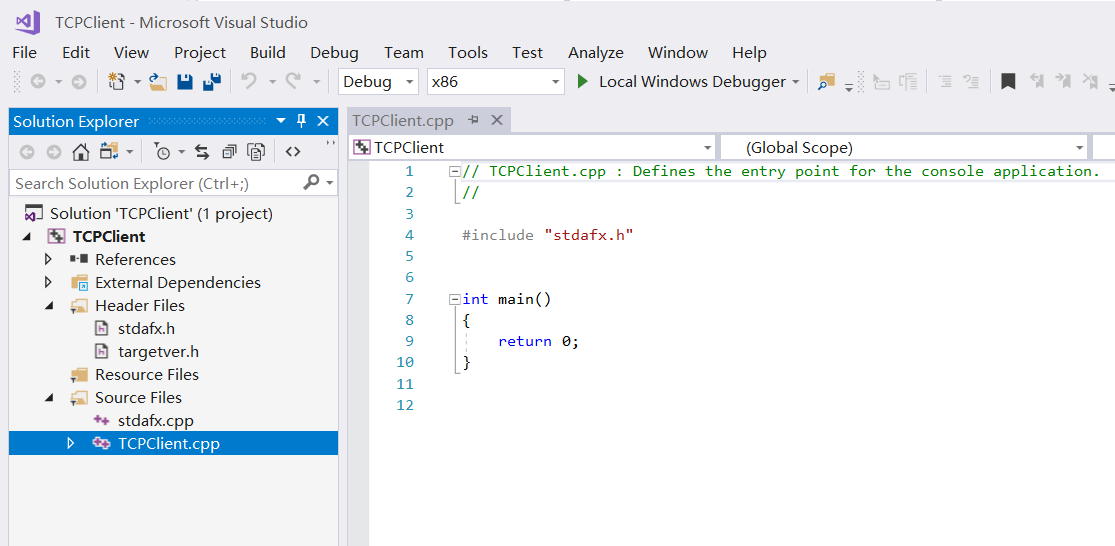


Figure 7

To add an existing source file to your project, first save the file in the project directory, then from the Solution Explorer window, right click on *Source Files*, then choose *Add*, *Existing Item…* (Figure 8)

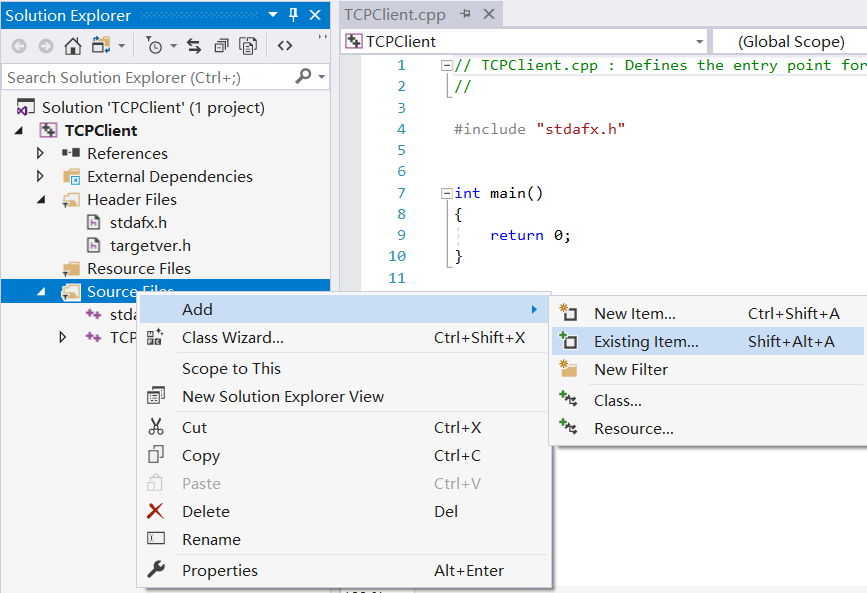


Figure 8

From the pop-up window, select the file or files you want to add and click *Add* (Figure 9).

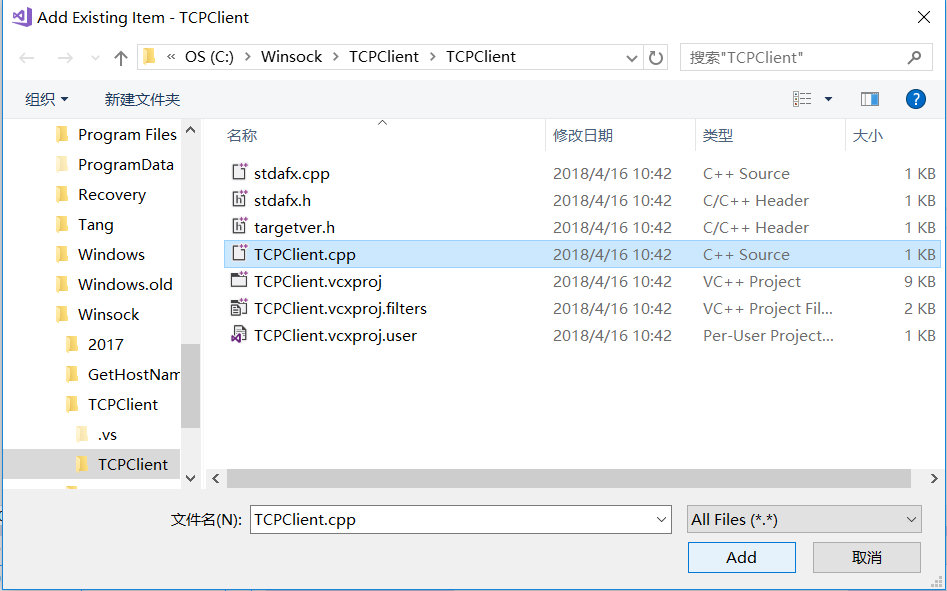


Figure 9

1. **Compile and build your project**

Right-click on a source file, click *Compile* to compile an individual source file. (Figure 10)

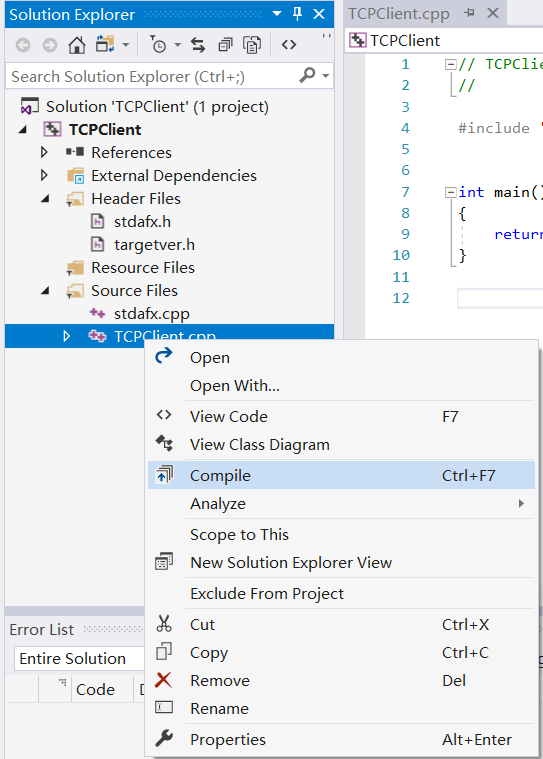


Figure 10

Click *Build* then *Build Solution* to build the executable file. (Figure 11)

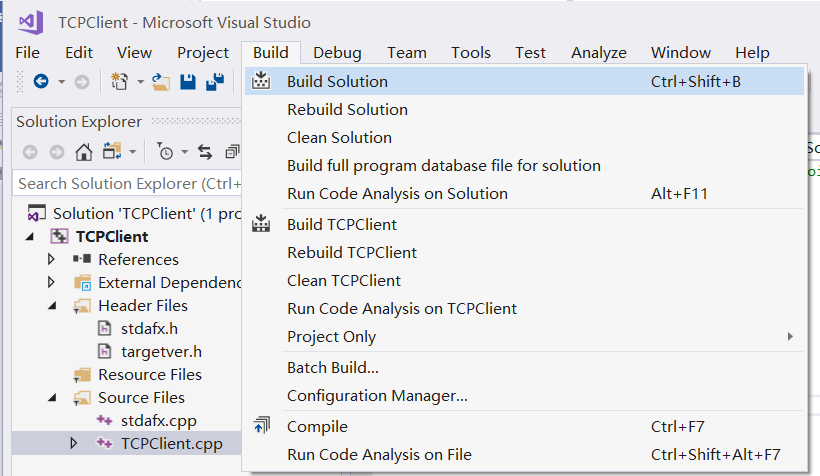


Figure 11

If the build process succeeded, an executable file (in this example TCPClient.exe) will be put in the Debug subdirectory under your project directory. (Figure 12)

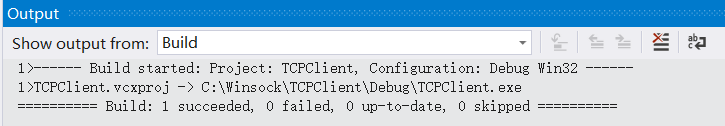


Figure 12

1. **Running the Project from the command line**

You can run applications from the command line by opening the MS-DOS prompt in Windows.

Once you have opened the MS-DOS prompt, you are ready to locate your file. When the project was created you selected the directory to store your project in the *Location,* and *Project Name* fields on the *New* window screen. To get to your project directory, you need to type *cd \directory* where directory is the *location\project name* of your project. (Figure 13)

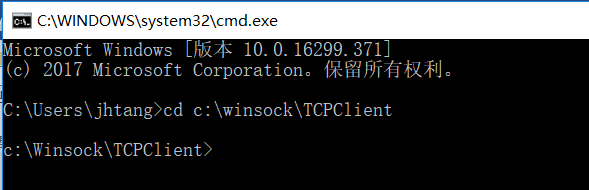


Figure 13

The executable is located in the Debug directory of the project directory.

To execute the application, type in the name of the executable file along with any arguments. (Figure 14)

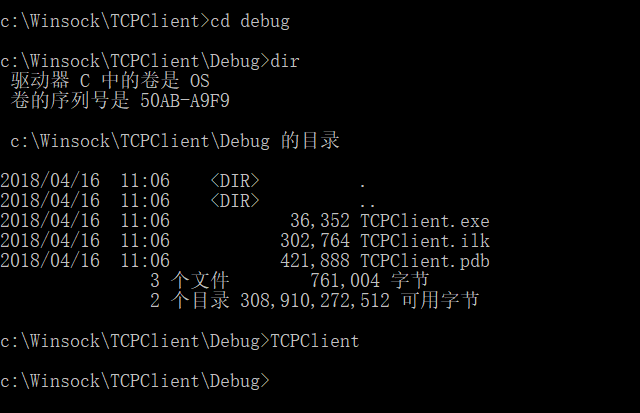


Figure 14