

Setting Up Visual Studio

Jacob Kneibel

Executive Summary

In this document, a tutorial is given for setting up Visual Studio, a windows programming environment provided by Microsoft. The tutorial will guide the reader through the setup process from choosing the version of Visual Studio, downloading and installing, and creating a new project, and at the end step through a hello world example. Some explanation will also be given of the various types of C++ project visual studio has to offer. It is expected that the reader has knowledge of the C++ language to begin with.

Keywords

Windows, Visual Studio, C++

Introduction

C++ is a ubiquitous programming language throughout platforms. Microsoft has created a programming environment, Visual Studio, that allows a user to write C++ code, create a hierarchy of header and c++ files, and debug the program. Some versions of the environment are freely available from microsoft, the 2013 community version being the most recent one to be free. Visual Studio additionally have the capability to program in languages Python, HTML5, JavaScript, C#, Visual Basic, and F#. This tutorial will be focused on the C++ aspect of it however. Visual Studio is an environment specifically designed to make developing on the Windows operating system easier. It comes with tools for the creation of console applications as well as windows applications.

For the 2013 Community Visual Studio:

Hardware requirements:

- 1.6 GHz or faster processor
- 1 GB of RAM (1.5 GB if running on a virtual machine)
- 20 GB of available hard disk space
- 5400 RPM hard disk drive
- DirectX 9-capable video card that runs at 1024 x 768 or higher display resolution

Supported Operating Systems:

- Windows 8.1 (x86 and x64)
- Windows 8 (x86 and x64)
- Windows 7 SP1 (x86 and x64)
- Windows Server 2012 R2 (x64)
- Windows Server 2012 (x64)
- Windows Server 2008 R2 SP1 (x64)

Installation

The Installer for Visual Studio can be downloaded from [here](#)^[1]. Various versions of Visual Studio are available. Some are 90-day trials of commercial versions, and some are free versions of the environment. Visual Studio Express 2010 is also available from here, which is kept for legacy code that does not work on the later versions. The version that we want for this tutorial is the Community 2013 with Update 4. This one provides a full environment, but is limited for non-commercial applications. To download the installer select the version and click on Install Now under Installation options.

Once the installer is downloaded, run it. It will ask you to provide a path for the installation the default path is acceptable, so you can either keep it or change it to suit your needs. Additionally you must agree to the License Terms and Privacy Policy and click on Next. It will then ask you to choose Optional features to install, leave the default settings and hit Install. Visual Studio will now install on your computer. Once the Installer is finished, click on Launch to open up Visual Studio. It will ask you to sign in, either do so or click on not now. Then in the next window select general for Development Settings and choose a theme. Once done Visual Studio show be open and look like Figure 1.

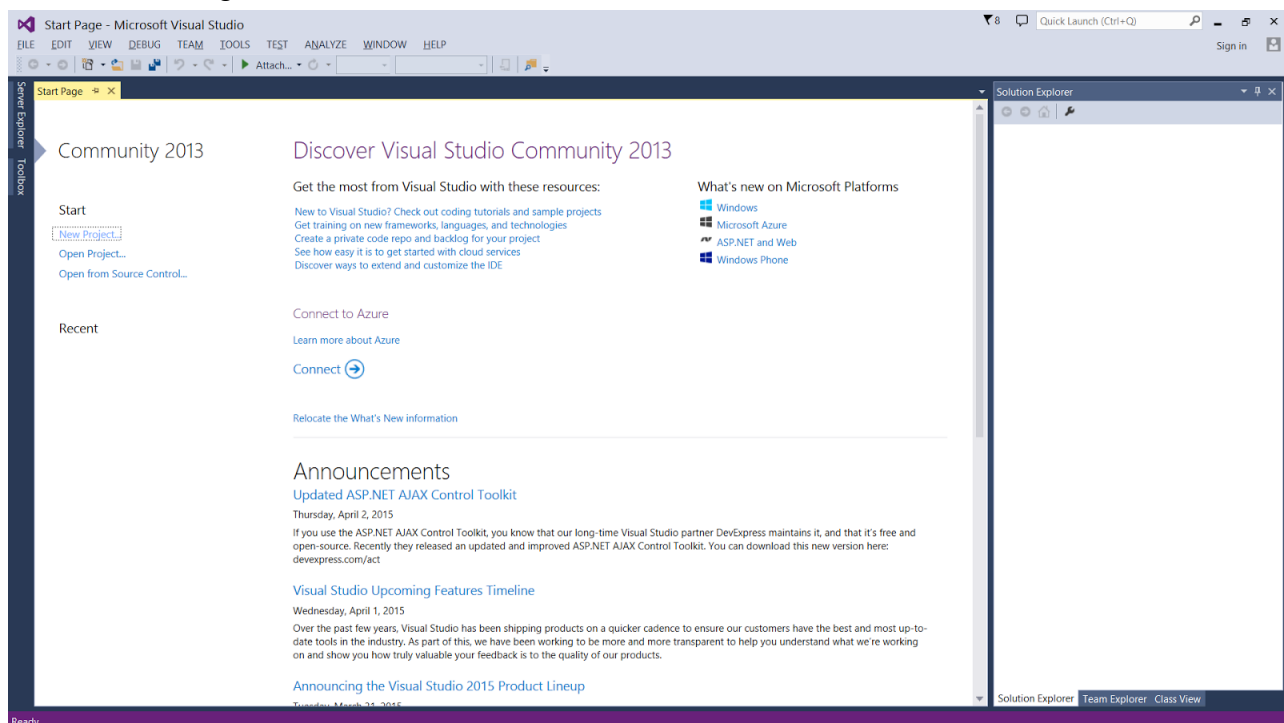


Figure 1: First Open of Visual Studio

Creating a New Project

From the Starting window go to File in the upper left and then New and Project. (Notation of navigating tabs will use -> so here it would be File->New->Project) A new window will pop up. For the first project, on the left side of the window select Visual C++, indicating that we want to create a C++ project. Then in the middle of the screen select Win32 Console Application. This project will be an application that runs in the command prompt of windows. Since this is the first project we are creating in Visual Studio, let's name the project HelloWorld. In Figure 2, this window is shown with the changes we made boxed with red. Now click OK. A wizard will appear, click Finish to create the project.

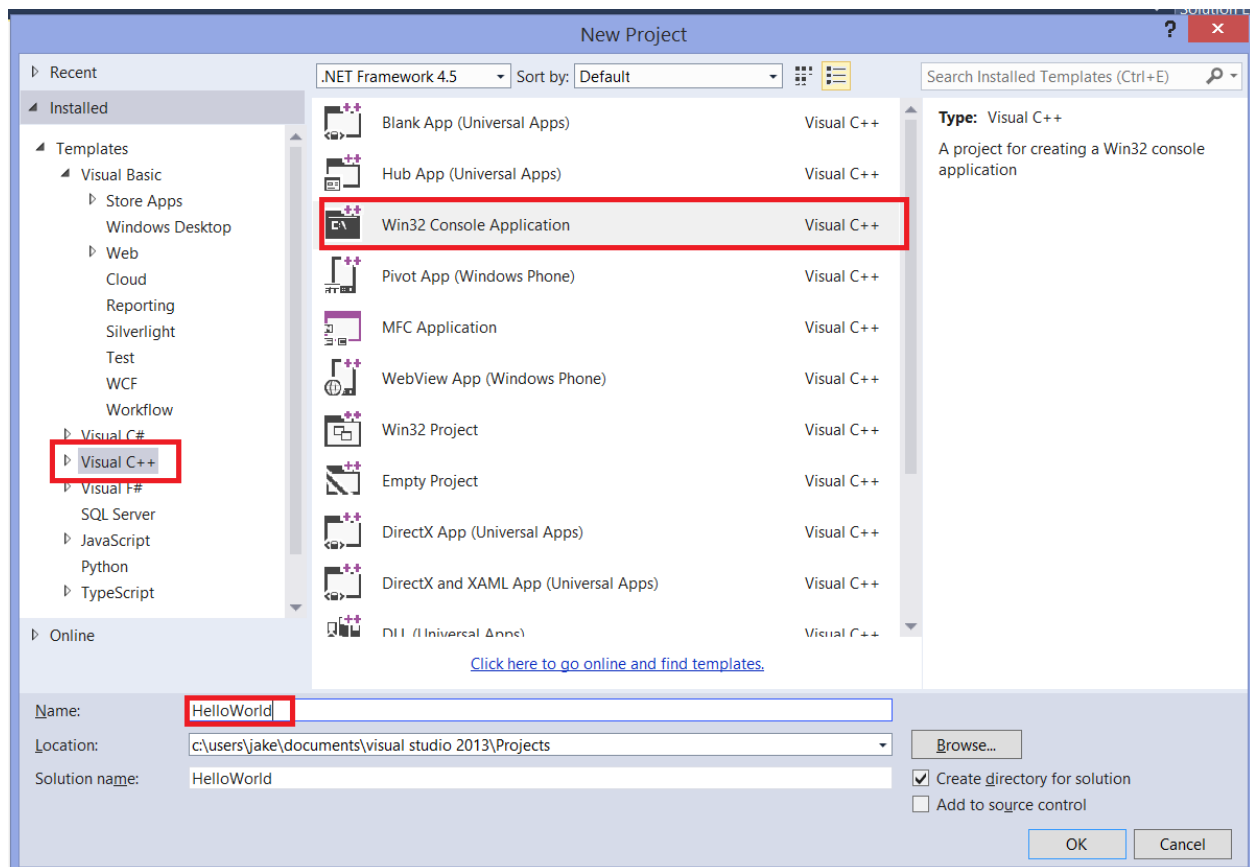


Figure 2: Creating a new C++ Console project

The project is now open and the project files are displayed in the Solution Explorer, which is boxed with red in Figure 3.

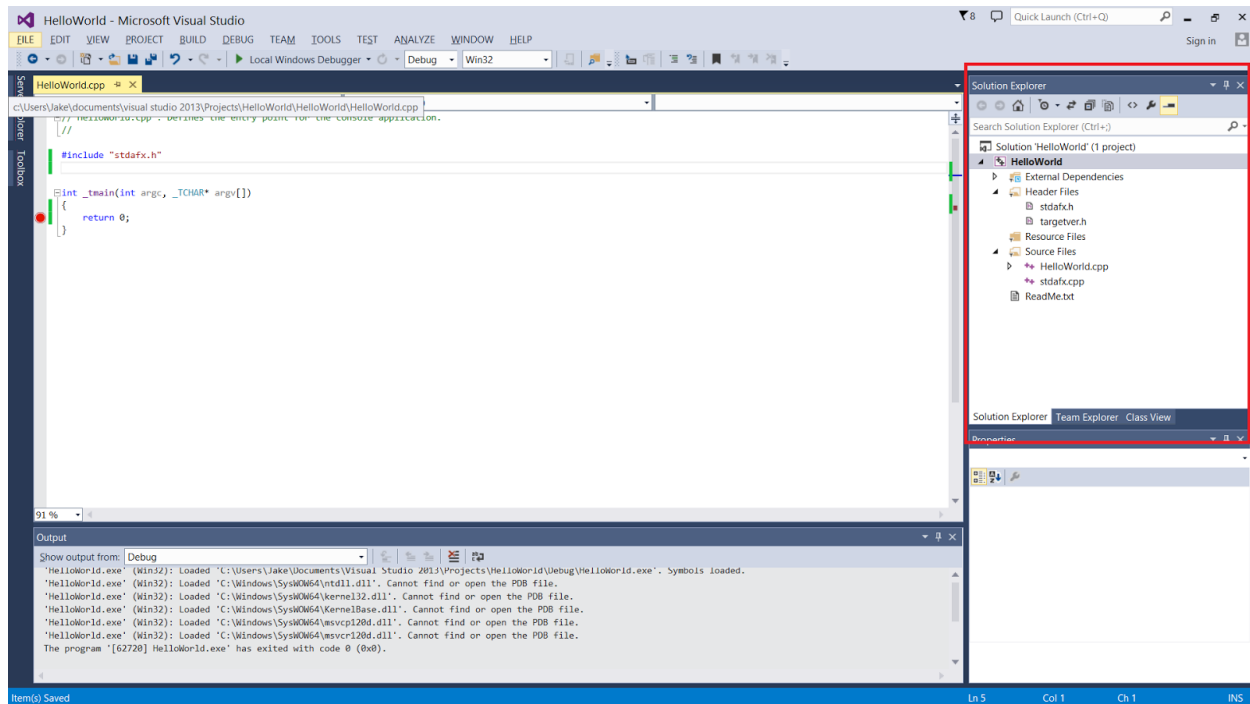


Figure 3: Project Environment, Solution Explorer is boxed in red

The HelloWorld.cpp file is open in a window. This is the main C++ file that will run when the code is compiled. Currently it has a few lines of code in it:

```
#include "stdafx.h"
```

```
int _tmain(int argc, _TCHAR* argv[])  
{  
    return 0;  
}
```

The include is a call to a header file created by Visual Studio that contains a few includes to header files that are standard and used by the window console application. Other header files can be included in this file. To open this file, the solution explorer on the right side of the window has the file under Header Files. Back in the HelloWorld.cpp file the function _tmain is the main function and has inputs for command line passing of parameters. argc is the number of arguments passed in and argv is an array of the arguments. Now Let's add two lines to HelloWorld.cpp to print out Hello World. Add the line:

```
#include <iostream>
```

right below `#include "stdafx.h"` and add the line:

```
std::cout << "Hello World" << std::endl;
```

in the function `_tmain` before the `return 0;`.

Now to compile and run the code click on "Local Windows Debugger" at the top of the window, it'll have a green arrow next to it. Once done compiling it will run. The console window will open for a brief second and display "Hello World". The program will end and the window will close. To be able to keep the console open to read the output, right click on the line `"return 0;"` and select Breakpoint->Insert Breakpoint. Now again compile and run the code. The console window should stay open though it will move to the background so you'll have to navigate to it and bring it to the front. Figure 4 shows what the console should look like. To finish running the code after the breakpoint navigate back to Visual Studio and select "Continue" located where "Local Windows Debugger" button was before.

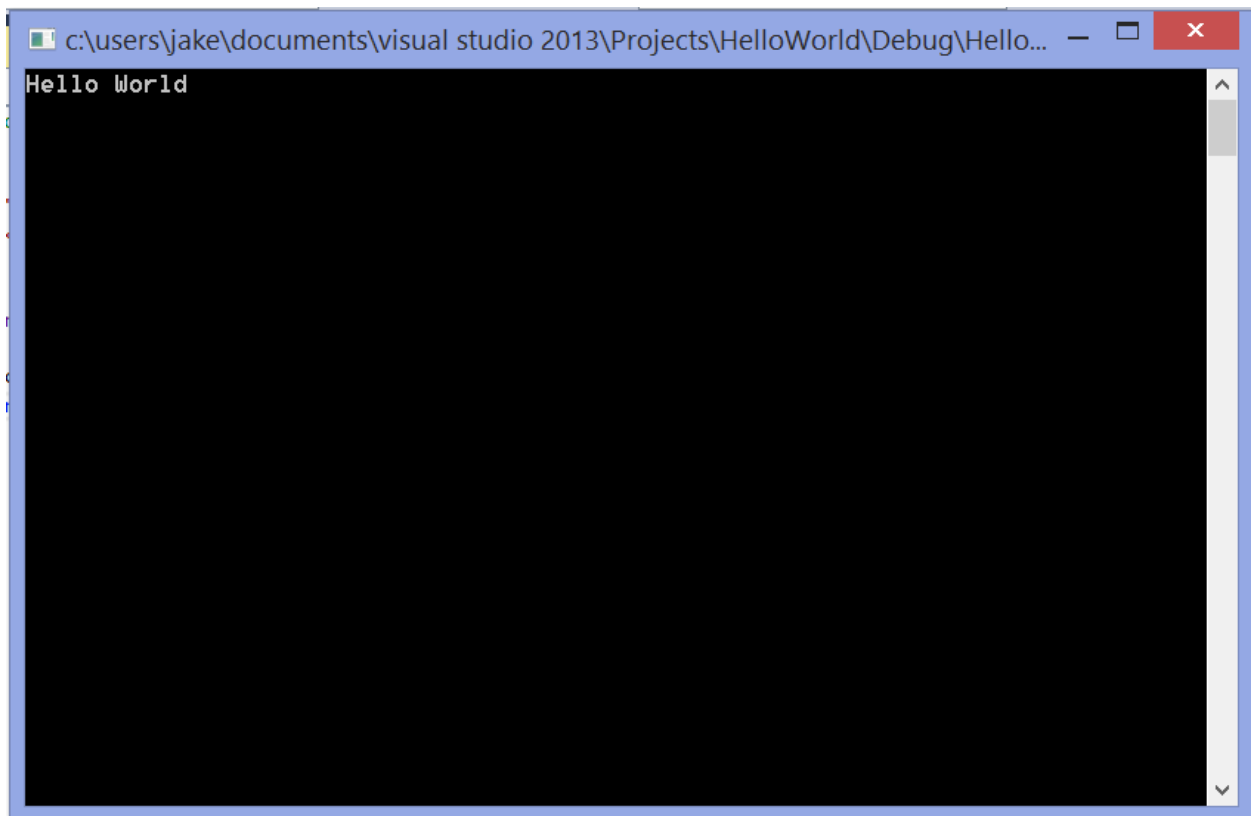


Figure 4: Hello World Created for Console Application

To add additional files to the project, right click the project in the solution explorer and go to Add, from which you can either add a new file or an existing one. Add and new file and a window pops up. A header file or C++ file can be chosen to be created among other options that are on the left side of the window.

Project Properties

Various Project Properties can be accessed from the Solution explorer by right clicking on the project and selecting project properties. This opens up a window as seen in Figure 5. This window contains many different properties of the project and the different property categories are in a selection box on the left side. The one property which can cause the most errors is in the Configuration Properties->C/C++-> general tab. It is Additional Include Directories. If in your program an include is declared for something that can not be found by Visual Studio it could be that that include isn't on the computer or the directory it is in is not present in Additional Include Directories. To add directories to this list the path can be specified here and for more than one directory they should be separated by semicolons. Directories specified here will be checked by Visual Studio when compiling for the various files needed. To showcase this, create a new header file by going to File->New File and selecting C++ and header file. add the line

```
//Test Header
```

and save this file as test.h somewhere besides the project directory for example "C:\Users\YourName\Documents\ . now in the main project add

```
#include "test.h"
```

If you try to compile this now it will not work because Visual Studio can not locate test.h. To fix this go into project properties and C/C++->general and add the directory where test.h is located to Additional Include Directories and click OK. Now the program will be able to compile. While this header file does not do anything, Other header files located throughout your computer can be useful, so it is important to know how to add directories to this list.

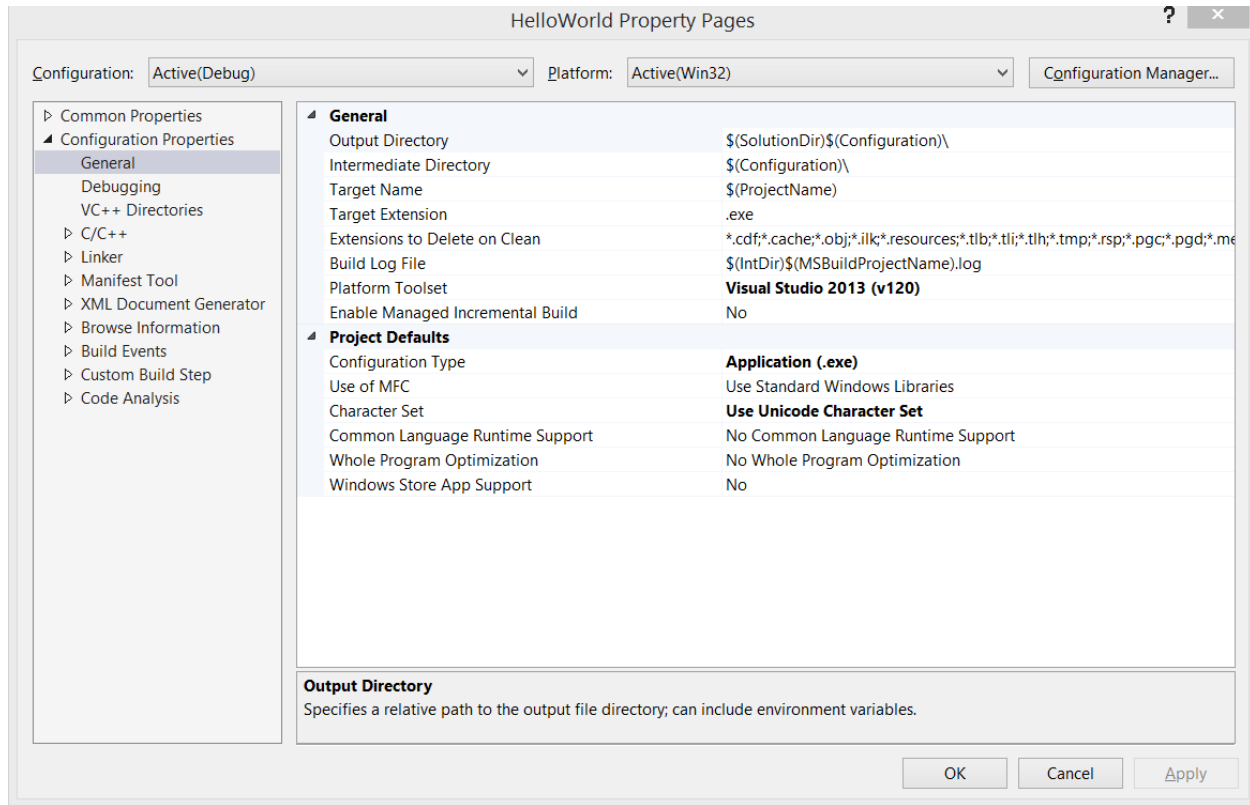


Figure 5: Project Properties Window

Running the program from the executable file

To run the program without the Visual Studio environment, we need to find the executable file produced from compiling the code. To locate the folder it is in, right click the solution in the solution explorer and click “Open Folder in File Explorer”. This will open up the File Explorer in the Hello World solution folder. Go into the Debug folder. The executable file is here, double click to run it. Again it only shows up briefly. Another fix for this is to create a batch file. right click in the folder and go to New->Text Document. Rename the file run.bat, changing the extension from .txt to .bat. Now right click the file and Edit. Add the following text and save:

```
cmd /k HelloWorld.exe
```

This will run the HelloWorld program in the command prompt and open the command prompt for further commands, when you double click the run.bat file. to exit the command prompt either click the red X in the corner or type exit and enter.

Conclusion

In this tutorial, the reader was guided through the process of Installing Visual Studio, and creation of a console application. Additionally some features of Visual Studio were explored. While this is only a small fraction of the functionality of Visual Studio, the reader should now be able to get a project started and build some C++ code in Visual Studio and explore the environment on their own.

References

[1] Visual Studio Download page

https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs#DownloadFamilies_2