Introduction to Visual Studio

General Information

Visual Studio (VS) is an integrated development environment (IDE) for creating and testing software applications.

Examples of the applications that we can build are:

- Console applications
- Graphics applications
- Desktop applications
- Websites
- Mobile applications

To accomplish this, a number of different tools is available for application development, such as a source code editor, a compiler/interpreter, a linker, a debugger, etc.

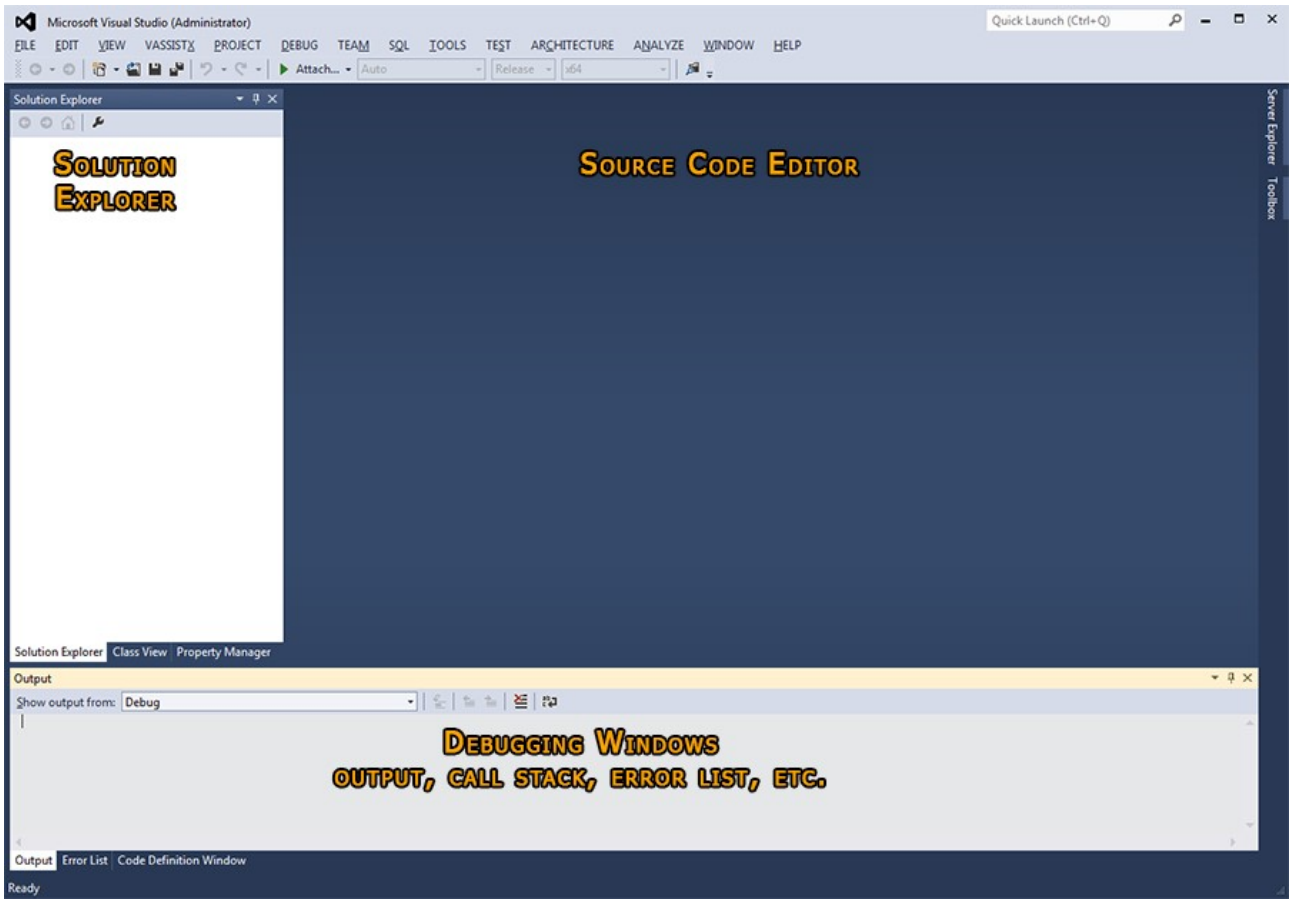
Visual Studio supports several programming languages, such as C, C++, C# and Visual Basic.

It also includes the .NET Framework for web and desktop application development. However, we do not need this for computer graphics based applications.

The version we will work on is Visual Studio 2012 which is available from your DreamSpark account. The latest version is Visual Studio 2015.

In computer graphics, we mainly use C++ as the main programming language and a graphics API such as OpenGL or Direct3D.

Visual Studio User Interface



Visual Studio allows us to create solutions, which are containers for projects. Each project can be an executable, a dynamic or a static library, etc. and we can create more than one project (e.g. one executable and several DLL libraries) and store them in the solution.

When we create a new solution and/or project(s), the Solution Explorer is a hierarchical view of projects, files, and code. The solution files have extension .sln and the project files have extension .vcxproj.

The Debugging Windows at the bottom provide a number of windows mainly used for debugging purposes. The most common windows are:

- Output (for printing text, compilation and linking status, etc.)

- Error List (compile and linking errors and warnings)

- Call Stack (shows the functions or procedure calls that are currently on the stack)

- Watch (used for evaluating variables and expressions at runtime)

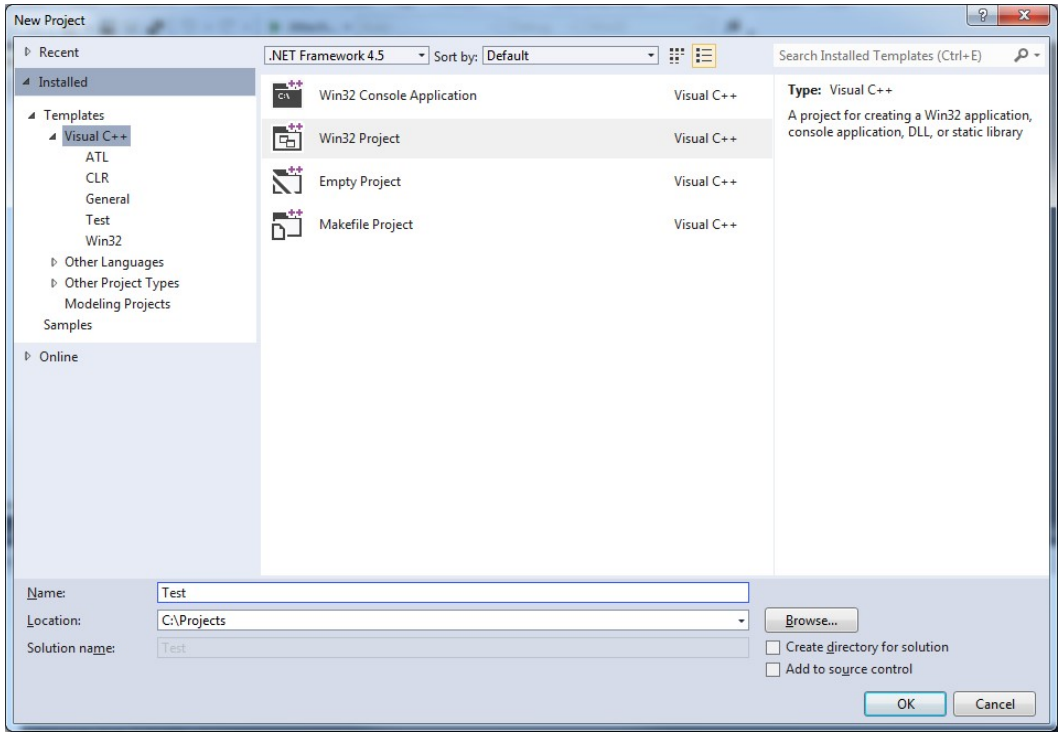
- Locals (displays variables local to the current context)

If not shown, these windows are available from the menu bar (View->Windows or Debug->Windows).

The first time you run Visual Studio, you will be asked to choose a default environment. Choose Visual C++. You can change the default environment by going to Tools->Import and Export Settings selecting Reset All Settings.

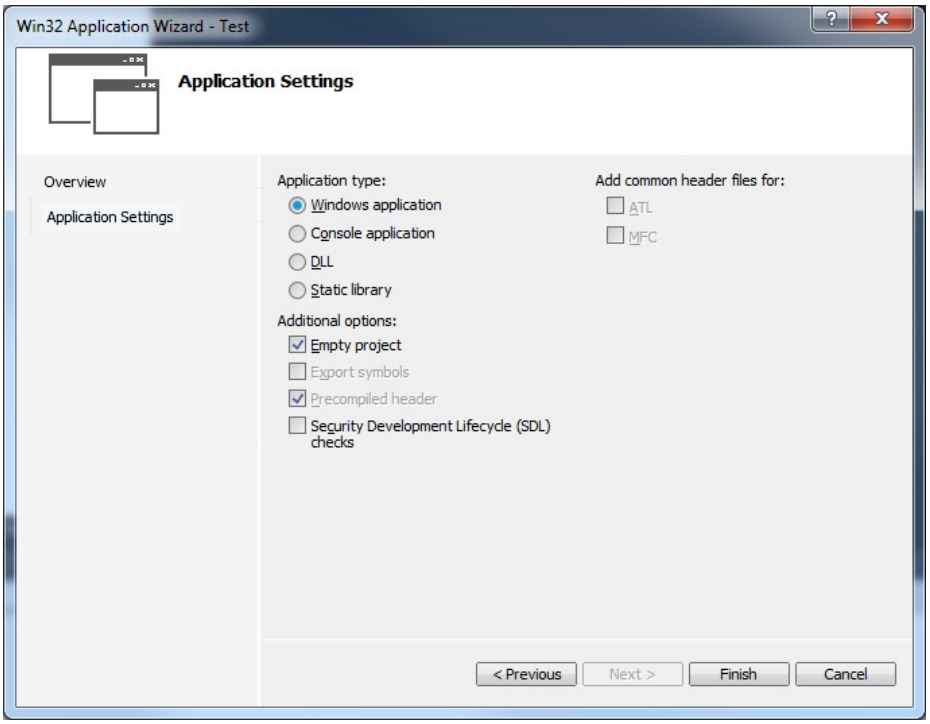
Creating a New Solution

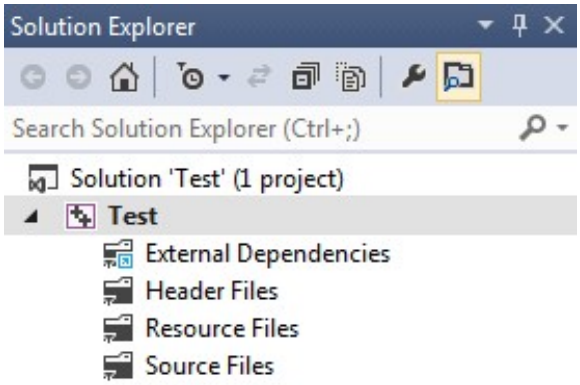
Select File->New Project and then select Visual C++ in the Project Types. Finally, select the type of project you want to create.



We can create different types of projects for different programming languages. In graphics we use Win32 console application or Win32 Project (where the console is disabled). In the field Name we write the name of the solution and the name of the project (initially one solution will be created with one project).

Next, we select Empty Project in the Application Settings and click Finish.



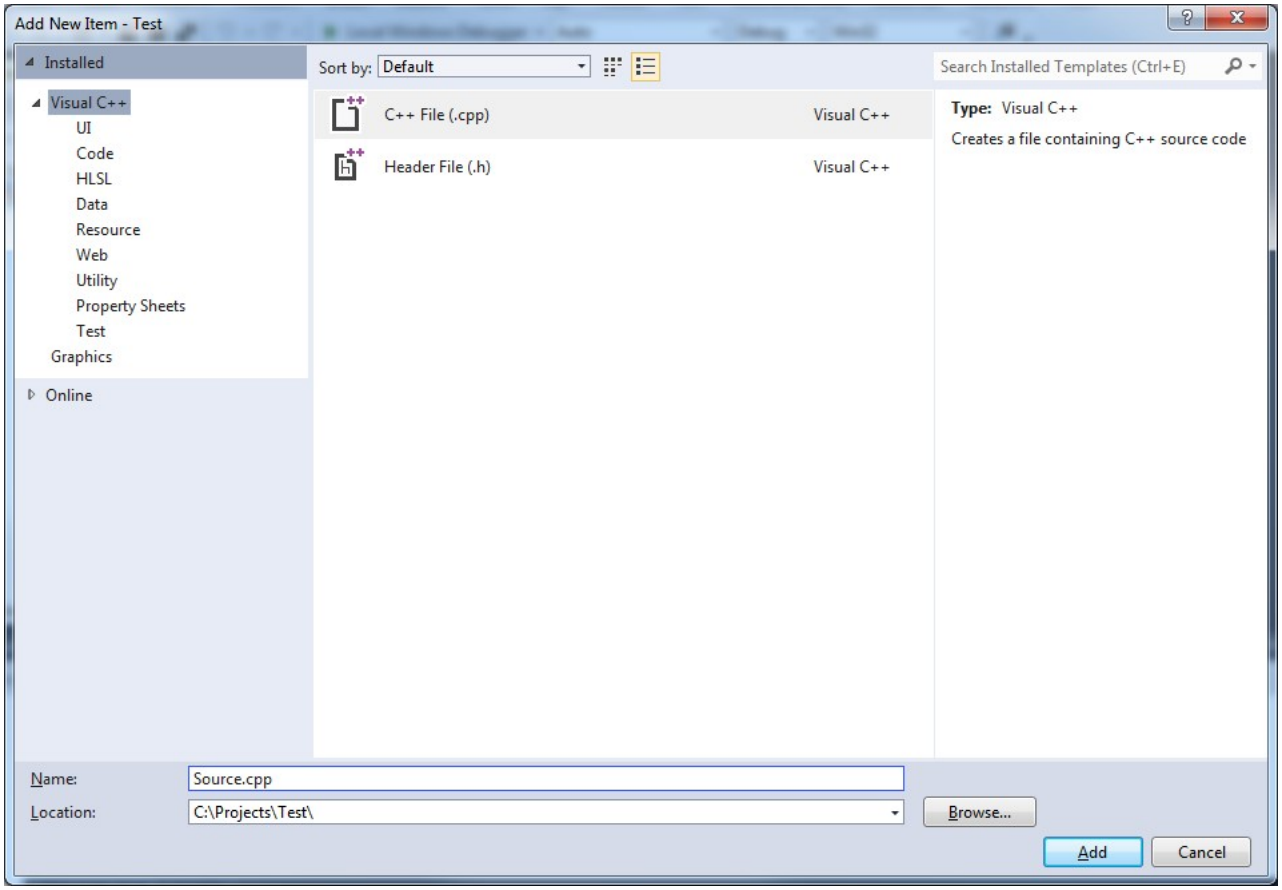


Now, we have created the

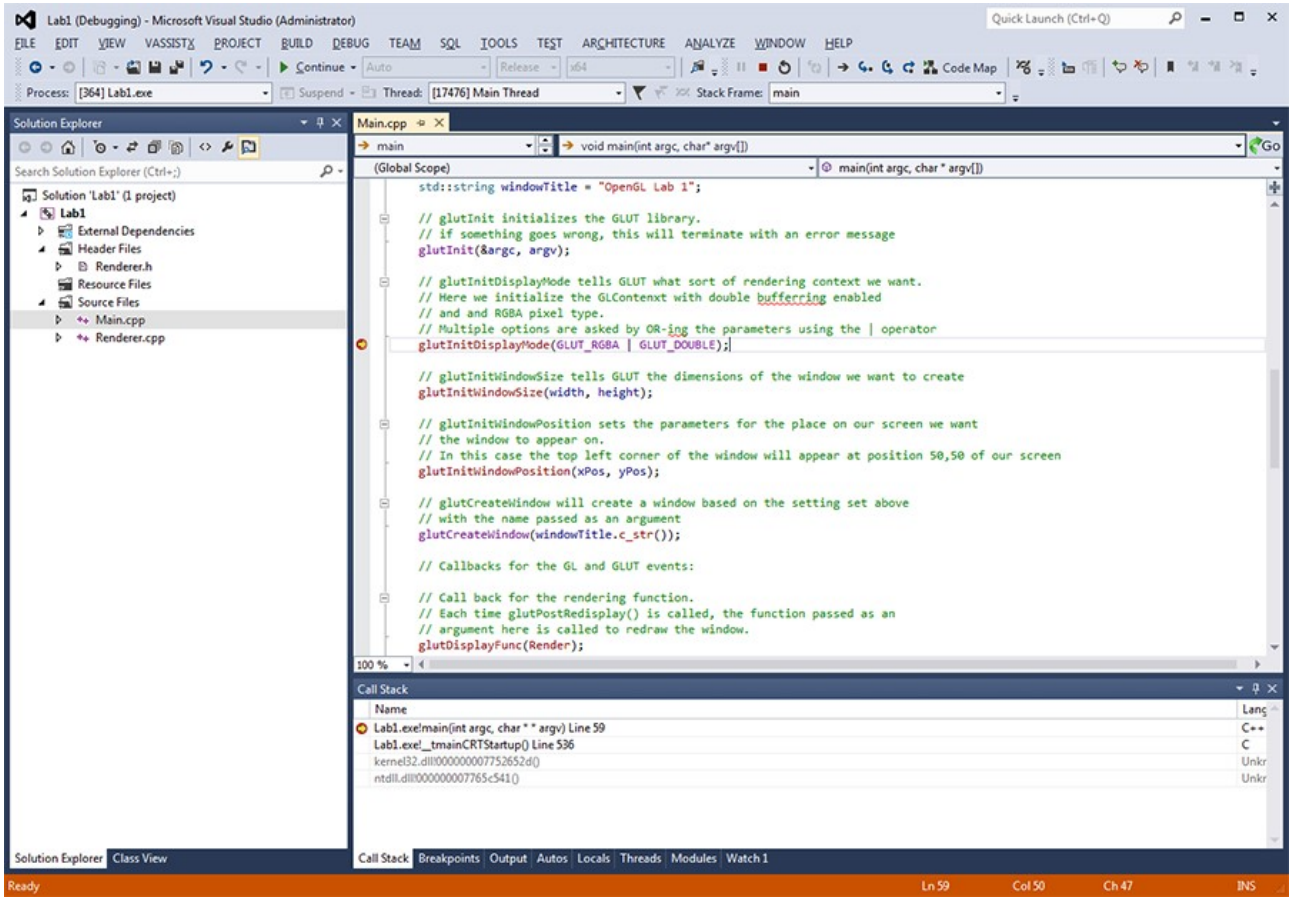
- A Solution named Test
- A Project named Test
- Three filters storing header (.h), source (.cpp), and resource files.

The project is empty and contains no source code.

In order to add a new file, we select and right click on the appropriate filter and choose Add -> New Item and add a new cpp or .h file.



Debugging



We can compile our application by going to Build->Build Solution and execute it by either clicking on the green arrow icon or by pressing F5.

When debugging, we can add a breakpoint by pressing F9 on the line we want the breakpoint to be placed in.

When the debugger stops at a breakpoint, we can use the Debugging Windows to view information about the state of our application at that specific point.

More Information

You can use the following links to get more information regarding Visual Studio and C++.

- http://msdn.microsoft.com/en-us/library/60k1461a.aspx (Visual C++)
- http://www.cplusplus.com/reference/ (C++ reference and tutorials)