

CHAPTER 2

Compile and Run

Visual Studio Compilation

With the Hello World program completed, the next step is to compile and run it. To do so, open the Debug menu and select Start Without Debugging, or simply press Ctrl+F5. Visual Studio will then compile and run the application, which displays the string in a console window.

Console Compilation

If you did not have an IDE such as Visual Studio, you could still compile the program as long as you have the .NET Framework installed. To try this, open a console window (C:\Windows\System32\cmd.exe) and navigate to the project folder where the source file is located. You then need to find the C# compiler called `csc.exe`, which is located in a path similar to the one shown here. Run the compiler with the source filename as an argument and it will produce an executable in the current folder.

```
C:\MySolution\MyProject>  
\Windows\Microsoft.NET\Framework64\v3.5\  
csc.exe Program.cs
```

If you try running the compiled program from the console window, it will show the same output as the one created by Visual Studio.

```
C:\MySolution\MyProject> Program.exe  
Hello World
```

Language Version

As of Visual Studio 2019, the default version of C# used is determined by which version of .NET the project targets. You can change this version by right-clicking the project node in the Solution Explorer and selecting Properties. From there, under the Application tab, there is a drop-down list labeled Target framework. To use the latest features from C# 8.0, the project needs to target .NET Core 3.0 or later so make sure it is selected. For this option to be available, the .NET Core workload must have been selected when installing Visual Studio 2019 and the project must be created using one of the .NET Core templates.

Comments

Comments are used to insert notes into the source code. C# uses the standard C++ comment notations, with both single-line and multi-line comments. They are meant only to enhance the readability of the source code and have no effect on the end program. The single-line comment begins with `//` and extends to the end of the line. The multi-line comment may span multiple lines and is delimited by `/*` and `*/`.

```
// single-line comment
```

```
/* multi-line  
   comment */
```

In addition to these, there are two documentation comments. There is one single-line documentation comment that starts with `///` and one multi-line documentation comment that is delimited by `/**` and `*/`. These comments are used when producing class documentation.

```
/// <summary>Class level documentation.</summary>
class MyApp
{
    /** <summary>Program entry point.</summary>
        <param name="args">Command line arguments.</param>
    */
    static void Main(string[] args)
    {
        System.Console.WriteLine("Hello World");
    }
}
```