

Introduction to Visual Studio

Last Updated : 22 Sep, 2023

Visual Studio is an **Integrated Development Environment(IDE)**

developed by Microsoft to develop Desktop applications, GUI(Graphical User Interface), console, web applications, mobile applications, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

Evolution of Visual Studio

The first version of VS(Visual Studio) was released in 1997, named as Visual Studio 97 having version number 5.0. The latest version of Visual Studio is 15.0 which was released on March 7, 2017. It is also termed as Visual Studio 2017. The supported *.Net Framework Versions* in latest Visual Studio is 3.5 to 4.7. Java was supported in old versions of Visual Studio but in the latest version doesn't provide any support for Java language.

Visual Studio Editions

There are 3 editions of Microsoft Visual Studio as follows:

1. Community

It is a **free** version which is announced in 2014. *All other editions are paid.* This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like *.Net*

applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than \$1 Million(US Dollars) then you are not permitted to use this edition. In a non-enterprise organization, up to five users can use this edition. Its main purpose is to provide the Ecosystem(Access to thousands of extensions) and Languages(You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.

2. Professional

It is the commercial edition of Visual Studio. It comes in Visual Studio 2010 and later versions. It provides the support for XML and XSLT editing and includes the tool like Server Explorer and integration with Microsoft SQL Server. Microsoft provides a free trial of this edition and after the trial period, the user has to pay to continue using it. Its main purpose is to provide Flexibility(Professional developer tools for building any application type), Productivity(Powerful features such as CodeLens improve your team's productivity), Collaboration(Agile project planning tools, charts, etc.) and Subscriber benefits like Microsoft software, plus Azure, Pluralsight, etc.

3. Enterprise

It is an integrated, end to end solution for teams of any size with the demanding quality and scale needs. Microsoft provides a 90-days free trial of this edition and after the trial period, the user has to pay to continue using it. The main benefit of this edition is that it is highly scalable and deliver high-quality software.

Getting Started with Visual Studio 2017

- First, you have to download and install the Visual Studio. For that, you can refer to [Downloading and Installing Visual Studio 2017](#). Don't forget to select the .NET core workload during the installation of VS 2017. If you forget then you have to [modify](#) the installation.

- You can see a number of tool windows when you will open the Visual Studio and start writing your first program as follows:

The screenshot shows the Microsoft Visual Studio interface. The title bar reads "GeekforGeeks - Microsoft Visual Studio". The menu bar includes "File", "Edit", "View", "Project", "Build", "Debug", "Team", "Tools", "Test", "Analyze", "Window", and "Help". The toolbar shows various icons for file operations and debugging. The main editor window displays the following C# code:

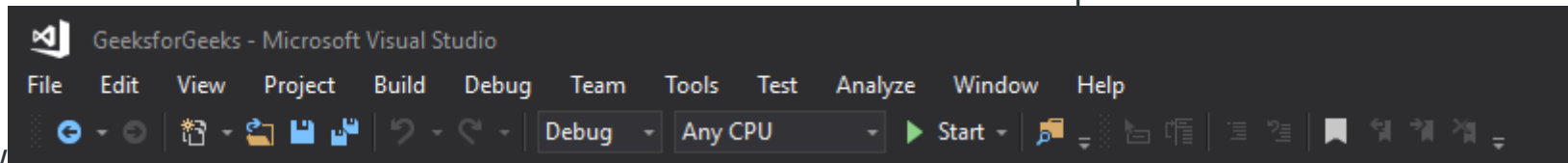
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace GeekforGeeks
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.WriteLine("Hello Geeks!");
14            Console.ReadLine();
15        }
16    }
17 }
18
```

CODE EDITOR

The screenshot shows the Output window in Visual Studio. The title bar reads "100 %". The window contains the text "Output" and "Show output from:" followed by a dropdown menu and several icons. The window is currently empty.

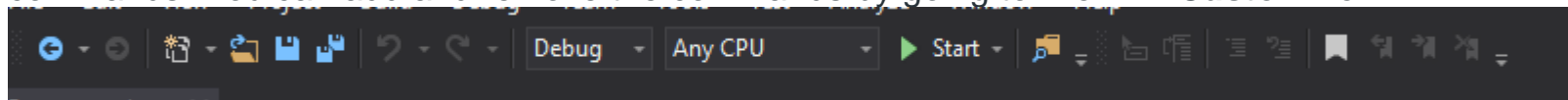
OUTPUT WINDOW

1. **Code Editor:** Where the user will write code.
 2. **Output Window:** Here the Visual Studio shows the outputs, compiler warnings, error messages and debugging information.
 3. **Solution Explorer:** It shows the files on which the user is currently working.
 4. **Properties:** It will give additional information and context about the selected parts of the current project.
- A user can also add windows as per requirement by choosing them from **View** menu. In Visual Studio the tool windows are customizable as a user can add more windows, remove the existing open one or can move windows around to best suit.
 - **Various Menus in Visual Studio:** A user can find a lot of menus on the top screen of Visual Studio as shown



below

1. Create, Open and save projects commands are contained by **File** menu.
 2. Searching, Modifying, Refactoring code commands are contained by the **Edit** menu.
 3. **View** Menu is used to open the additional tool windows in Visual Studio.
 4. **Project** menu is used to add some files and dependencies in the project.
 5. To change the settings, add functionality to Visual Studio via extensions, and access various Visual Studio tools can be used by using **Tools** menu.
- The below menu is known as the **toolbar** which provide the quick access to the most frequently used commands. You can add and remove the commands by going to **View → Customize**



Advantages of using Visual Studio IDE

- A full-featured programming platform for several operating systems, the web, and the cloud, Visual Studio IDE is available. Users can easily browse the UI so they can write their code quickly and precisely.
- To help developers quickly identify potential errors in the code, Visual Studio offers a robust debugging tool.
- Developers can host their application on the server with confidence because they have eliminated anything that could lead to performance issues.
- No matter what programming language developers are using, users of Visual Studio can get live coding support. For faster development, the Platform offers an autocomplete option. The built-in intelligent system offers descriptions and tips for APIs.
- Through Visual Studio IDE you can easily collab with your teammates in a same project. This IDE helps the developers to share, push and pull their code with their teammates.
- Every user of Visual Studio has the ability to customize it. They have the option to add features based on their needs. For example, they can download add-ons and install extensions in their IDE. Even programmers can submit their own extensions.

Note:

- Support for different programming languages in Visual Studio is added by using a special **VSPackage** which is known as *Language Service*.
- When you will install the Visual Studio then the functionality which is coded as VSPackage will be available as Service.
- Visual Studio IDE provides the three different types of services known as **SVsSolution**, **SVsUIShell**, and **SVsShell**.
- SVsSolution service is used to provide the functionality to enumerate solutions and projects in Visual Studio.
- SVsUIShell service is used to provide User Interface functionality like toolbars, tabs etc.
- SvsShell service is used to deal with the registration of VSPackages.