**INTRODUCTION**

# *Microsoft Visual Studio*

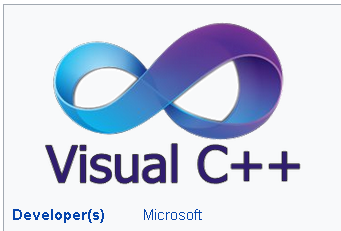
# *C:\Users\susha\Desktop\XYX\VS.PNG*

# Microsoft Visual Studio is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program) for [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), as well as [web sites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Native_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code).

# Visual Studio includes a [code editor](https://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) (the [code completion](https://en.wikipedia.org/wiki/Code_completion) component) as well as [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring). [The integrated debugger](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a [code profiler](https://en.wikipedia.org/wiki/Profiling_(computer_programming)), forms designer for building [GUI](https://en.wikipedia.org/wiki/GUI) applications, [web designer](https://en.wikipedia.org/wiki/Web_designer), [class](https://en.wikipedia.org/wiki/Class_(computing)) designer, and [database schema](https://en.wikipedia.org/wiki/Database_schema) designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for [source control](https://en.wikipedia.org/wiki/Source_control) systems (like [Subversion](https://en.wikipedia.org/wiki/Subversion_(software))) and adding new toolsets like editors and visual designers for [domain-specific languages](https://en.wikipedia.org/wiki/Domain-specific_language) or toolsets for other aspects of the [software development lifecycle](https://en.wikipedia.org/wiki/Software_development_lifecycle) (like the [Team Foundation Server](https://en.wikipedia.org/wiki/Team_Foundation_Server) client: Team Explorer).

# Visual Studio supports 36 different [programming languages](https://en.wikipedia.org/wiki/Programming_language) and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](https://en.wikipedia.org/wiki/C_(programming_language)),[[6]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-6) [C++](https://en.wikipedia.org/wiki/C%2B%2B) and [C++/CLI](https://en.wikipedia.org/wiki/C%2B%2B/CLI) (via [Visual C++](https://en.wikipedia.org/wiki/Visual_C%2B%2B)), [VB.NET](https://en.wikipedia.org/wiki/VB.NET) (via [Visual Basic .NET](https://en.wikipedia.org/wiki/Visual_Basic_.NET)), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)) (via [Visual C#](https://en.wikipedia.org/wiki/Visual_C_Sharp)), [F#](https://en.wikipedia.org/wiki/F_Sharp_(programming_language)) (as of Visual Studio 2010[[7]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-7)) and [Type Script](https://en.wikipedia.org/wiki/TypeScript) (as of Visual Studio 2013 Update 2). Support for other languages such as [Python](https://en.wikipedia.org/wiki/Python_(programming_language)),[[8]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-8) [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), [Node.js](https://en.wikipedia.org/wiki/Node.js), and [M](https://en.wikipedia.org/wiki/MUMPS) among others is available via language services installed separately. It also supports [XML](https://en.wikipedia.org/wiki/XML)/[XSLT](https://en.wikipedia.org/wiki/XSLT), [HTML](https://en.wikipedia.org/wiki/HTML)/[XHTML](https://en.wikipedia.org/wiki/XHTML), [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) (and [J#](https://en.wikipedia.org/wiki/J_Sharp)) were supported in the past.

***Microsoft Visual C++***

******

**Microsoft Visual C++** (often abbreviated to **MSVC**) is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) product from [Microsoft](https://en.wikipedia.org/wiki/Microsoft) for the [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), and [C++/CLI](https://en.wikipedia.org/wiki/C%2B%2B/CLI) [programming languages](https://en.wikipedia.org/wiki/Programming_language). MSVC is [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software); it was originally a standalone product but later became a part of [Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) and made available in both [trial ware](https://en.wikipedia.org/wiki/Trialware) and [freeware](https://en.wikipedia.org/wiki/Freeware) forms. It features tools for [developing](https://en.wikipedia.org/wiki/Software_development) and [debugging](https://en.wikipedia.org/wiki/Debugging) C++ code, especially code written for [Windows API](https://en.wikipedia.org/wiki/Windows_API), [DirectX](https://en.wikipedia.org/wiki/DirectX) and [.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework).

Many [applications](https://en.wikipedia.org/wiki/Application_software) require [redistributable](https://en.wikipedia.org/wiki/Redistributable) Visual C++ packages to function correctly. These packages are often installed independently of applications, allowing multiple applications to make use of the package while only having to install it once. These Visual C++ redistributable and runtime packages are mostly installed for standard [libraries](https://en.wikipedia.org/wiki/Library_(computing)) that many applications use.

***Application***

Creating an Application Page, having browse button, selecting image from any directory/folder. Have a place (box) to display the image selected(Image read) using Visual C++ in Visual Studio.

# WINDOWS FORM APPLICATION

# Step 1: Open Visual Studio -> Start Page.

# C:\Users\susha\Desktop\XYX\STRAT PAGE.jpg

# Step 2: Select File -> New -> Project

# C:\Users\susha\Desktop\XYX\NEW.png

# Step 3:Select Installed -> Visual C++ -> CLR -> CLR Empty Project -> OK

# C:\Users\susha\Desktop\XYX\CLR.png

# Step 4: Select -> Project -> Add New Item

# C:\Users\susha\Desktop\XYX\ADD new item.png

# Step 5: Select Visual C++ -> UI -> Windows Form -> Add

# C:\Users\susha\Desktop\XYX\Windows Form.png

# Application Page Will Appear As

# C:\Users\susha\Desktop\XYX\Design,h.png

# Step 6: Adding Controls

# PictureBox

# Select Toolbox -> Common Control -> PictureBox

# C:\Users\susha\Desktop\XYX\PB.png

# Button

# Select Toolbox -> Common Control -> Button

# C:\Users\susha\Desktop\XYX\Button.png

# 

# CheckBox

# Select Toolbox -> Common Control -> CheckBox

# 

# CODE

# Application Design Page[MyForm.h(Design)]

# C:\Users\susha\Desktop\XYX\MyDesign.png

# MyForm.h Source Code

# C:\Users\susha\Desktop\XYX\MyFormh.png

# MyForm.cpp Source Code

# C:\Users\susha\Desktop\XYX\MyFormcpp.png

# Code For Individual Controls

# PictureBox

private: System::Void pictureBox1\_Click(System::Object^ sender, System::EventArgs^ e) {}

# Browse Button

private: System::Void OpenButton\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK)

{

pictureBox1->Load(openFileDialog1->FileName);

}

}

# Close Button

private: System::Void CloseButton\_Click(System::Object^ sender, System::EventArgs^ e)

{

this->Close();

}

# Clear Button

private: System::Void ClearButton\_Click(System::Object^ sender, System::EventArgs^ e)

{

pictureBox1->Visible = false;

}

# Stretch CheckBox

private: System::Void checkBox1\_CheckedChanged(System::Object^ sender, System::EventArgs^ e)

{

if (checkBox1->Checked)

{

pictureBox1->SizeMode = PictureBoxSizeMode::StretchImage;

}

else

{

pictureBox1->SizeMode = PictureBoxSizeMode::Normal;

}

}