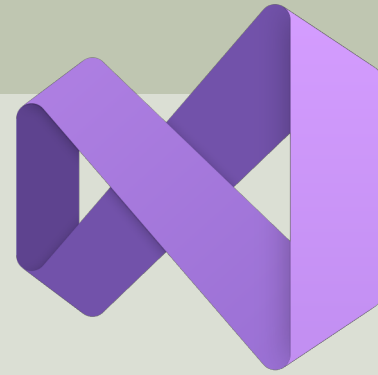


<https://www.halvorsen.blog>



Visual Studio and C#

Hans-Petter Halvorsen



Contents

- Introduction to Visual Studio, C# and .NET
- How to create a basic Windows Desktop Application
- Step-by-step Code Examples using the “Windows Forms App” template will be provided
- You will learn to use the Solution Explorer, use the Toolbox, use the Designer to create User Interface, set Properties, create and use Event Handlers, create and use Variables, use built-in Methods in your code, create a Method and finally create a Class with Methods
- This will give you the foundation for creating any kind of Application using Visual Studio and C#

<https://www.halvorsen.blog>

Introduction



Hans-Petter Halvorsen

[Table of Contents](#)

Visual Studio

- Visual Studio is an Integrated Development Environment (IDE) from Microsoft
- You can use it to create Desktop Applications, Web Applications, etc.
- You can use different Programming Languages (C#, VB.NET and F#), but C# is the default option
- Visual Studio comes in 3 different editions; Professional, Enterprise and Community (free)
- <https://visualstudio.microsoft.com/>

Visual Studio

The screenshot displays the Visual Studio IDE interface for a Windows Forms application. The main window shows a form titled "Form1" with a text box labeled "label1" and a button labeled "button1". The interface includes a menu bar (File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help), a toolbar, and a search bar (Search (Ctrl+Q)). The left sidebar contains the "Toolbox" with a search box and a list of controls such as Pointer, BackgroundWorker, BindingSource, Button, CheckBox, etc. The right sidebar shows the "Solution Explorer" with a tree view of the project "WinFormsApp1" containing files like Dependencies, Form1.cs, Form1.Designer.cs, Form1.resx, and Program.cs. Below the Solution Explorer is the "Properties" window showing "WinFormsApp1 Project Properties" with fields for File Name, Full Path, and Project Folder. At the bottom, the "Error List" window shows "0 Errors", "0 Warnings", and "0 Messages". The status bar at the very bottom indicates "Ready" and includes options for "Add to Source Control" and "Select Repository".

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Win...pp1 Live Share

Debug Any CPU WinFormsApp1

Form1...[esig]*

Form1

label1

button1

Toolbox

Search Toolbox

All Windows Forms

- Pointer
- BackgroundWorker
- BindingSource
- Button
- CheckBox
- CheckedListBox
- ColorDialog
- ComboBox
- ContextMenuStrip
- DataGridView
- DateTimePicker
- DomainUpDown
- ErrorProvider
- FileSystemWatcher
- FlowLayoutPanel
- FolderBrowserDialog
- FontDialog
- GroupBox
- HelpProvider
- HScrollBar
- ImageList
- Label
- LinkLabel
- ListBox
- ListView

Solution Explorer

Search Solution Explorer (Ctrl+...)

Solution 'WinFormsApp1' (1 of 1 projects)

- WinFormsApp1
 - Dependencies
 - Form1.cs
 - Form1.Designer.cs
 - Form1.resx
 - Program.cs

Properties

WinFormsApp1 Project Properties

File Name	WinFormsApp1.csproj
Full Path	C:\Temp\WinFormsAp
Project Folder	C:\Temp\WinFormsAp

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSe

Search Error List

Co...	Description	Project	File	L...
-------	-------------	---------	------	------

Ready

Add to Source Control Select Repository

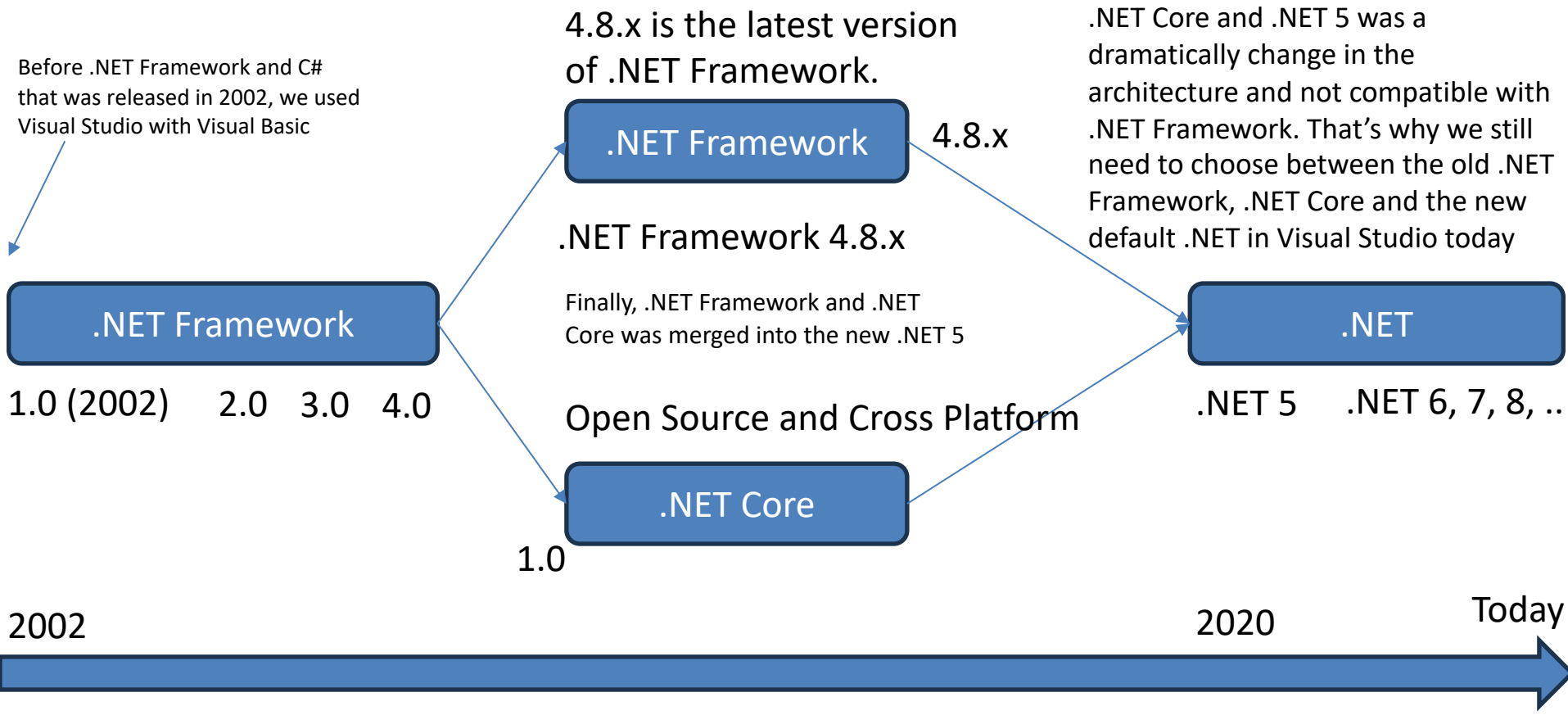
C#

- C# (pronounced “C-Sharp”) is a Programming Language
- Developed by Microsoft
- It runs on the .NET Framework
- C# is an Object-oriented Programming (OOP) Language
- C# is one of the most popular Programming Languages today
- Flexible Language:
 - Can be used for many different types of Applications; Desktop Applications, Web Applications, Mobile Apps, ..
 - All integrated into the Visual Studio IDE

.NET

- .NET is a free, open-source and cross-platform application platform supported by Microsoft
- In .NET you can choose between different Programming Languages, but **C# is the most used and recommended .NET language today**
- **Basically, .NET is just a huge code library** hiding all the “dirty work” to make it easy to make different types of Applications, either it is Desktop Applications, Web Applications or Mobile Applications

.NET from the beginning till today



Windows Forms

- As mentioned, with .NET you can create all kind of Applications, including Windows Desktop Applications, Web Applications, Mobile Applications, etc. using the same tool/IDE (Visual Studio) and the same Programming Language
- When it comes to Windows Desktop Applications you also have many choices, like Windows Forms Applications, WPF Applications, Windows Store Applications, etc.
- Windows Forms Applications is probably the most used of alle these alternatives
- So, here in this tutorial we will focus on **Windows Forms Applications**

Visual Studio - Create New Project


Create a new project

Recent project templates


A list of your recently accessed templates will be displayed here.

Search for templates (Alt+S) Clear all


C# Windows Desktop

 **JUnit Test Project**
A project that contains NUnit tests that can run on .NET Core on Windows, Linux, and MacOS.


C# Linux macOS Windows Desktop Test

 **Windows Forms App (.NET Framework)**
A project for creating an application with a Windows interface

C# Windows Desktop

 **Windows Forms App**
A project template for creating a .NET Windows Forms (WinForms) App.


C# Windows Desktop

 **WPF Application**
A project for creating a .NET WPF Application

C# Windows Desktop

 **WPF Class Library**
A project for creating a class library that targets a .NET WPF Application

C# Windows Desktop Library

 **WPF Custom Control Library**
A project for creating a custom control library for .NET WPF Applications

C# Windows Desktop Library

Use this if you depends on 3.party libraries and backward combability

The old .NET Framework Windows Forms Application Template

The new default .NET Windows Forms Application Template

Use this for all new Applications

Back

Next

.NET Core and .NET 5 was a dramatically change in the architecture and not compatible with .NET Framework. That's why we still need to choose between the old .NET Framework, .NET Core and the new default .NET in Visual Studio today

<https://www.halvorsen.blog>

Windows Forms Application



Hans-Petter Halvorsen

[Table of Contents](#)

Windows Forms Example

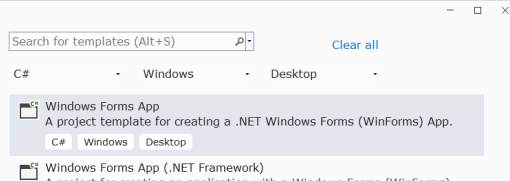
- Use the Toolbox
- Use the Designer to create User Interface
- Set Properties
- Event Handlers
- Create and use Variables
- Use built-in Methods in your code
- Create a Method
- Create a Class with Methods

Windows Forms App

Create a new project

Recent project templates

- Windows Forms App C#
- Windows Forms App (.NET Framework) C#
- Python Application Python
- MSTest Test Project C#
- ASP.NET Core Web App (Razor Pages) C#
- NUnit Test Project C#
- Unit Test Project (.NET Framework) C#
- .NET MAUI App C#
- Blazor WebAssembly Standalone App C#



Configure your new project

Windows Forms App C# Windows Desktop

Project name

Location

 ...

Solution name

 Place solution and project in the same directory

Project will be created in "C:\Users\hansha\OneDrive\Programming\Visual Studio Examples\Introduction to Visual Studio and C#\Code\HelloWorld\HelloWorld"

Additional information

Windows Forms App C# Windows Desktop

Framework

- .NET 8.0 (Long Term Support)
- .NET Core 3.1 (Out of support)
- .NET 5.0 (Out of support)
- .NET 6.0 (Long Term Support)
- .NET 7.0 (Standard Term Support)
- .NET 8.0 (Long Term Support)

Back

Create

Windows Forms App

The screenshot displays the Visual Studio IDE with a Windows Forms application in design mode. The main window is titled "Form1" and is currently empty. The interface includes a menu bar (File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help), a toolbar, and a search bar. The "Toolbox" on the left lists various controls, with "All Windows Forms" selected. The "Solution Explorer" on the right shows the project structure for "HelloWorld", including "Form1.cs", "Form1.Designer.cs", "Form1.resx", and "Program.cs". The "Properties" window at the bottom right shows the properties for the selected "Form1" control, including "Size" (816, 489), "StartPosition" (WindowsDefaultLocation), "Text" (Form1), and "TopMost" (False). The "Error List" at the bottom shows 0 errors, 0 warnings, and 0 messages.

Toolbox

- Common Controls
 - Pointer
 - Button
 - CheckBox
 - CheckedListBox
 - ComboBox
 - DateTimePicker
 - Label
 - LinkLabel
 - ListBox
 - ListView
 - MaskedTextBox
 - MonthCalendar
 - NotifyIcon
 - NumericUpDown
 - PictureBox
 - ProgressBar
 - RadioButton
 - RichTextBox
 - TextBox
 - ToolTip
 - TreeView
- All Windows Forms
 - Pointer
 - BackgroundWorker
 - BindingSource
 - Button
 - CheckBox
 - CheckedListBox
 - ColorDialog
 - ComboBox
 - ContextMenuStrip
 - DataGridView

Solution Explorer

- Solution 'HelloWorld' (1 of 1 project)
 - Dependencies
 - Form1.cs
 - Form1.Designer.cs
 - Form1.resx
 - Program.cs

Properties

Form1 System.Windows.Forms.Form

Size	816, 489
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	Form1
TopMost	False

Text

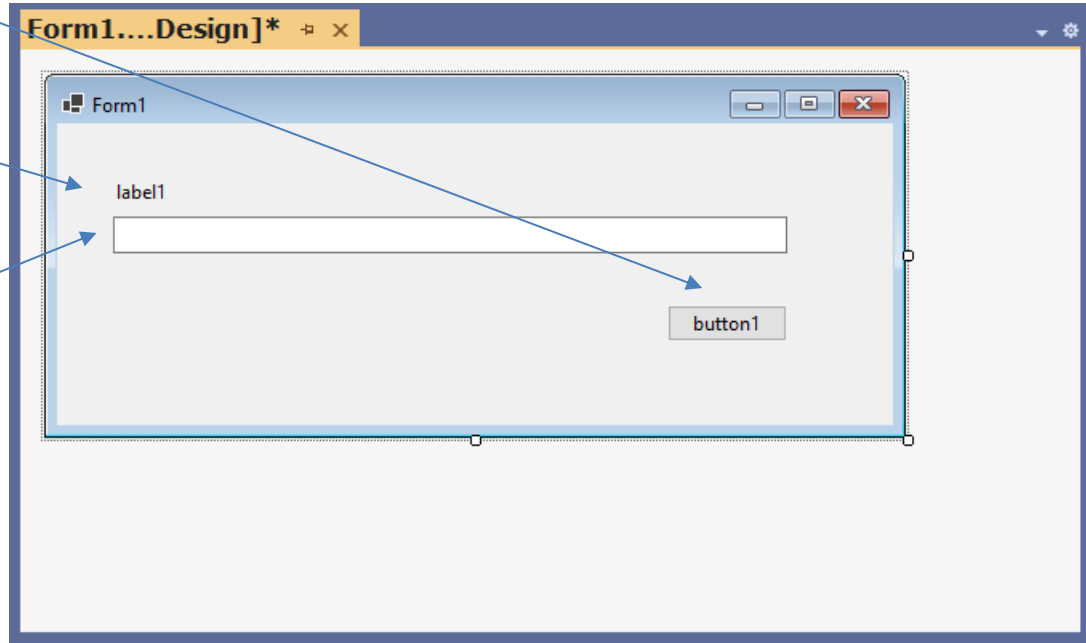
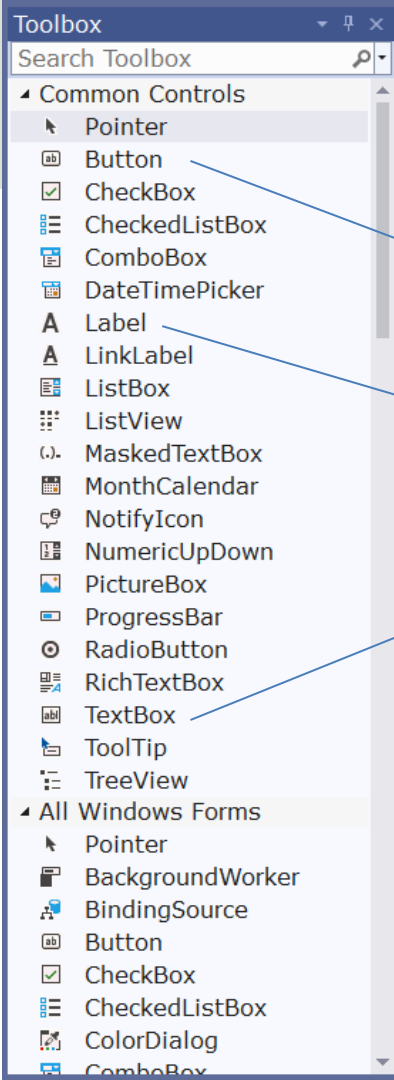
The text associated with the control.

Error List

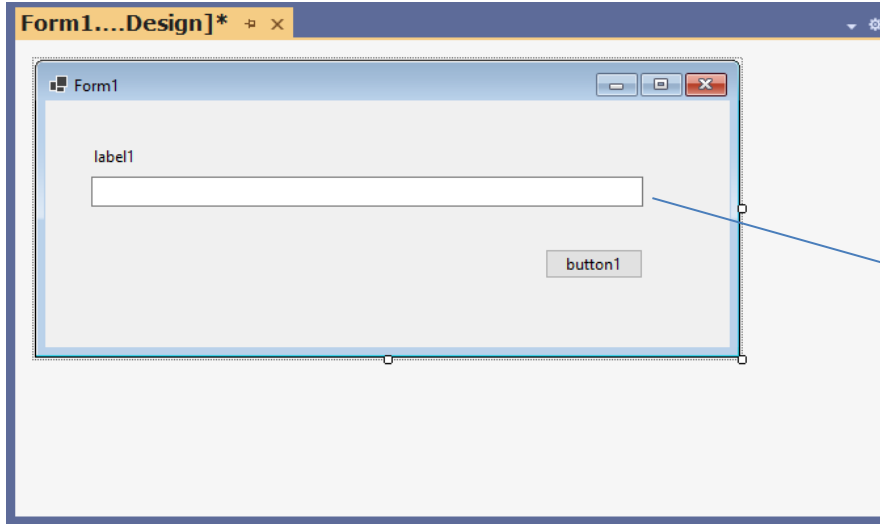
Entire Solution | 0 Errors | 0 Warnings | 0 Messages | Build + IntelliSense | Search Error

Code	Description	Project	File	Suppression...
------	-------------	---------	------	----------------

Toolbox and Designer



Properties



Properties

textBox1 System.Windows.Forms.TextBox

(DataBindings) (ControlBindings)

(Name)	textBox1
AcceptsReturn	False
AcceptsTab	False
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left

(Name)
Indicates the name used in code to identify th...

Create GUI

The screenshot displays the Visual Studio IDE with a Windows Forms application in design mode. The main window, titled "Form1.cs [Design]", shows a form with a text box labeled "Enter your Name:" and a "Click Me" button. The interface includes a menu bar (File, Edit, View, Project, Build, Debug, Format, Test, Analyze, Tools, Extensions, Window, Help), a toolbar, and several docked panels: Toolbox, Solution Explorer, Properties, and Error List.

Toolbox: Lists common controls such as Pointer, Button, CheckBox, CheckedListBox, ComboBox, DateTimePicker, Label, LinkLabel, ListBox, ListView, MaskedTextBox, MonthCalendar, NotifyIcon, NumericUpDown, PictureBox, ProgressBar, RadioButton, RichTextBox, and TextBox.

Solution Explorer: Shows the project structure for "HelloWorld", including "Form1.cs", "Form1.Designer.cs", "Form1.resx", and "Program.cs".

Properties: Displays the properties for the selected "Form1" control, including "StartPosition" (WindowsDefaultLocation), "Tag", "Text" (Form1), and "TopMost" (False). The "Text" property is expanded to show the description: "The text associated with the control."

Error List: Shows 0 Errors, 0 Warnings, and 0 of 1 Message.

Notifications: A vertical bar on the right side of the IDE contains "Notifications" and "Diagnostic Tools".

Create C# Code

The screenshot displays the Visual Studio IDE with the following components:

- Menu Bar:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search, HelloWorld
- Toolbox:** Search Toolbox, General. Text: "There are no usable controls in this group. Drag an item onto this text to add it to the toolbox."
- Code Editor:** Form1.cs [Design].

```
1 namespace HelloWorld
2 {
3     3 references
4     public partial class Form1 : Form
5     {
6         1 reference
7         public Form1()
8         {
9             InitializeComponent();
10        }
11
12        1 reference
13        private void btnClick_Click(object sender, EventArgs e)
14        {
15            string name = txtName.Text;
16            MessageBox.Show(name);
17        }
18    }
```
- Solution Explorer:** Solution 'HelloWorld' (1 of 1 project).
 - Dependencies
 - Form1.cs
 - Form1.Designer.cs
 - Form1.resx
 - Program.cs
- Properties:** Empty.
- Error List:** 0 Errors, 0 Warnings, 0 of 1 Message.
- Output:** Error List ... Output

Running Application

The screenshot displays the Visual Studio IDE in a debug configuration. The main window shows the source code for `Form1.cs` in the `HelloWorld` namespace. The code defines a `Form1` class that inherits from `Form`. It includes an `InitializeComponent()` method and a `btnClick_Click` event handler that reads the text from a text box and displays it in a message box.

```
namespace HelloWorld
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnClick_Click(object sender, EventArgs e)
        {
            string name = txtName.Text;

            MessageBox.Show(name);
        }
    }
}
```

The `Diagnostic Tools` window is open, showing a 1:00 minute session. It displays performance metrics for the application:

- Process Memory (MB):** 17 MB
- CPU (% of all processors):** 0%

The `Solution Explorer` on the right shows the project structure for `HelloWorld`, including `Form1.cs`, `Form1.Designer.cs`, `Form1.resx`, and `Program.cs`.

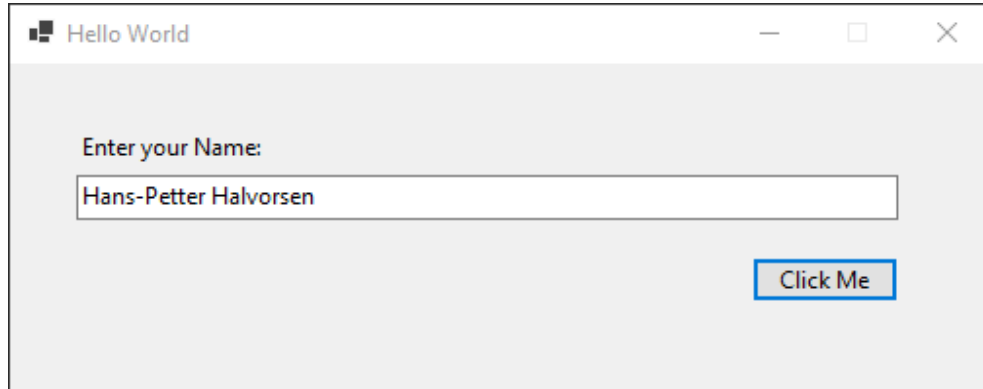
The `Error List` at the bottom left shows no errors or warnings, with a message count of 1 of 1.

Overlaid on the IDE is a `Form1` window titled "Enter your Name:" with a text box containing "Hans-Petter Halvorsen" and a "Click Me" button. A smaller message box is also visible, displaying the name "Hans-Petter Halvorsen" with an "OK" button.

Improvements

- Change Windows Title from “Form1” to something meaningful in Properties window
- Disable “Maximize” window in Properties window
- Enable Button if TextBox is empty
- Use If .. Else
- Split into First Name and Last Name
- .. (lots of more improvements can be made)

Updated Application

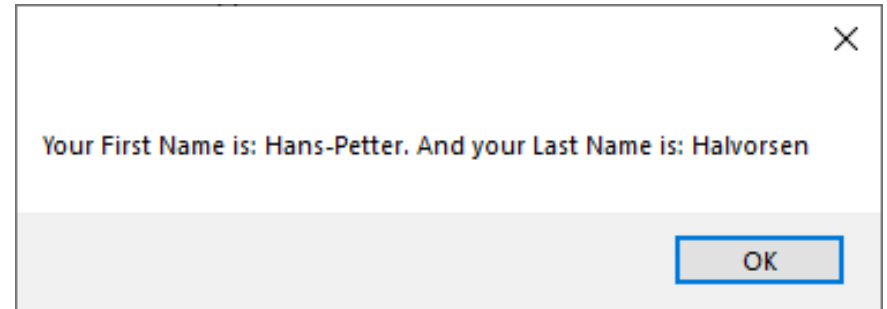


Hello World

Enter your Name:

Click Me

This screenshot shows a standard Windows-style application window titled "Hello World". It features a light gray background and a white title bar with standard minimize, maximize, and close buttons. The main content area contains a label "Enter your Name:" followed by a text input field containing the name "Hans-Petter Halvorsen". Below the input field is a blue button with the text "Click Me".



Your First Name is: Hans-Petter. And your Last Name is: Halvorsen

OK

This screenshot shows a dialog box with a white background and a close button in the top right corner. The main text reads "Your First Name is: Hans-Petter. And your Last Name is: Halvorsen". At the bottom right, there is a blue button with the text "OK".

Updated Code

The screenshot displays the Visual Studio IDE with the following components:

- File Explorer:** Shows the project structure for 'HelloWorld', including 'Form1.cs', 'Form1.Designer.cs', 'Form1.resx', and 'Program.cs'.
- Code Editor:** Contains the following C# code for `Form1`:

```
using System.Windows.Forms;

namespace HelloWorld
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            btnClick.Enabled = false;
        }

        private void btnClick_Click(object sender, EventArgs e)
        {
            string fullName = txtName.Text;

            var subNames = fullName.Split(' ');
            string firstName = subNames[0];
            string lastName = subNames[1];

            string message = "Your First Name is: " + firstName + ". And your Last Name is: " + lastName;

            MessageBox.Show(message);
        }

        private void txtName_TextChanged(object sender, EventArgs e)
        {
            string name = txtName.Text;

            if (name.Length > 0)
                btnClick.Enabled = true;
            else
                btnClick.Enabled = false;
        }
    }
}
```
- Properties Window:** Currently empty.
- Toolbox:** Shows a message: "There are no usable controls in this group. Drag an item onto this text to add it to the toolbox."
- Status Bar:** Indicates "No issues found" and shows the current line and column: "Ln: 37 Ch: 2 SPC CRLF".

```
using System.Windows.Forms;
```

```
namespace HelloWorld
```

```
{  
    public partial class Form1 : Form
```

```
{  
    public Form1()
```

```
{  
        InitializeComponent();
```

```
  
        btnClick.Enabled = false;
```

```
    }  
  
    private void btnClick_Click(object sender, EventArgs e)
```

```
{  
    string fullName = txtName.Text;
```

```
  
    var subNames = fullName.Split(' ');
```

```
    string firstName = subNames[0];
```

```
    string lastName = subNames[1];
```

```
  
    string message = "Your First Name is: " + firstName + ". And your Last Name is: " + lastName;
```

```
  
    MessageBox.Show(message);
```

```
    }  
  
    private void txtName_TextChanged(object sender, EventArgs e)
```

```
{  
    string name = txtName.Text;
```

```
  
    if (name.Length > 0)
```

```
        btnClick.Enabled = true;
```

```
    else
```

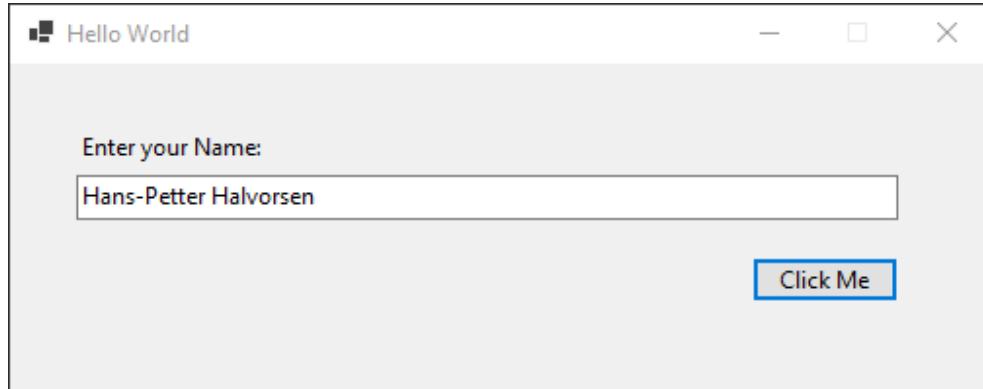
```
        btnClick.Enabled = false;
```

```
    }
```

```
}
```

```
}
```

Updated Application

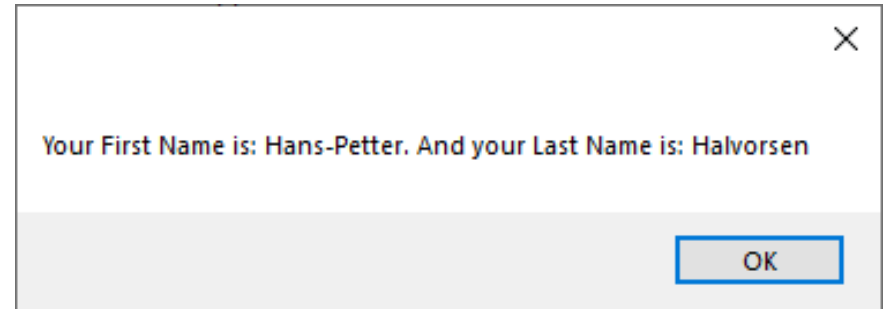


Hello World

Enter your Name:

Click Me

This screenshot shows a standard Windows-style window titled "Hello World". It features a light gray header bar with a close button (X) on the right. The main content area is white and contains a label "Enter your Name:" followed by a text input field containing the name "Hans-Petter Halvorsen". Below the input field is a blue button with the text "Click Me".



Your First Name is: Hans-Petter. And your Last Name is: Halvorsen

OK

This screenshot shows a dialog box with a white background and a light gray footer bar. The text "Your First Name is: Hans-Petter. And your Last Name is: Halvorsen" is centered in the main area. A blue button with the text "OK" is located in the bottom right corner of the footer bar. A close button (X) is visible in the top right corner.

Classes and Methods

We will update our Application:

- We will create a separate Method
- We will create a separate Class and put the Method inside the Class

Method

Creating the Method:

```
string SplitFullName(string fullName)
{
    var subNames = fullName.Split(' ');
    string firstName = subNames[0];
    string lastName = subNames[1];

    string message = "Hello! Your First Name is " + firstName + " and your Last Name is " + lastName;

    return message;
}
```

Using the Method:

```
private void btnClick_Click(object sender, EventArgs e)
{
    string fullName = txtName.Text;

    string message = SplitFullName(fullName);

    MessageBox.Show(message);
}
```

```
namespace HelloWorld
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            btnClick.Enabled = false;
        }

        private void btnClick_Click(object sender, EventArgs e)
        {
            string fullName = txtName.Text;

            string message = SplitFullName(fullName);

            MessageBox.Show(message);
        }

        private void txtName_TextChanged(object sender, EventArgs e)
        {
            string name = txtName.Text;

            if (name.Length > 0)
                btnClick.Enabled = true;
            else
                btnClick.Enabled = false;
        }

        string SplitFullName(string fullName)
        {
            var subNames = fullName.Split(' ');
            string firstName = subNames[0];
            string lastName = subNames[1];

            string message = "Hello! Your First Name is " + firstName + " and your Last Name is " + lastName;

            return message;
        }
    }
}
```

Class

Creating the Class and Method:

```
namespace HelloWorld
{
    public class Person
    {
        public string SplitFullName(string fullName)
        {
            var subNames = fullName.Split(' ');
            string firstName = subNames[0];
            string lastName = subNames[1];

            string message = "Hello! Your First Name is " + firstName + " and your Last Name is " + lastName;

            return message;
        }
    }
}
```

Using the Class and Method:

```
private void btnClick_Click(object sender, EventArgs e)
{
    string fullName = txtName.Text;

    Person person = new Person();
    string message = person.SplitFullName(fullName);

    MessageBox.Show(message);
}
```

```
using System.Windows.Forms;
```

```
namespace HelloWorld
```

```
{  
    public partial class Form1 : Form
```

```
    {  
        public Form1()
```

```
        {  
            InitializeComponent();
```

```
            btnClick.Enabled = false;
```

```
        }  
  
        private void btnClick_Click(object sender, EventArgs e)
```

```
        {  
            string fullName = txtName.Text;
```

```
            Person person = new Person();
```

```
            string message = person.SplitFullName(fullName);
```

```
            MessageBox.Show(message);
```

```
        }  
  
        private void txtName_TextChanged(object sender, EventArgs e)
```

```
        {  
            string name = txtName.Text;
```

```
            if (name.Length > 0)  
                btnClick.Enabled = true;
```

```
            else  
                btnClick.Enabled = false;
```

```
        }
```

```
    }
```

```
}
```

Summary

- We have used Visual Studio and C# and created a basic Windows Desktop Application.
- Step-by-step Code Examples using the “Windows Forms App” template have been provided.
- You have learned to use the Solution Explorer, use the Toolbox, use the Designer to create User Interface, set Properties, create and use Event Handlers, create and use Variables, use built-in Methods in your code, create a Method and finally create a Class with Methods.
- This will give you the foundation for creating any kind of Applications using Visual Studio and C#.

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

