Tutorial: Create a WinForms app with Visual Basic

Article • 04/03/2024

In this tutorial, you create a Visual Basic application that has a Windows Forms user interface. The Visual Studio integrated development environment (IDE) includes all the tools you need to create a Windows Forms app.

In this tutorial, you learn how to:

✔ Create a project
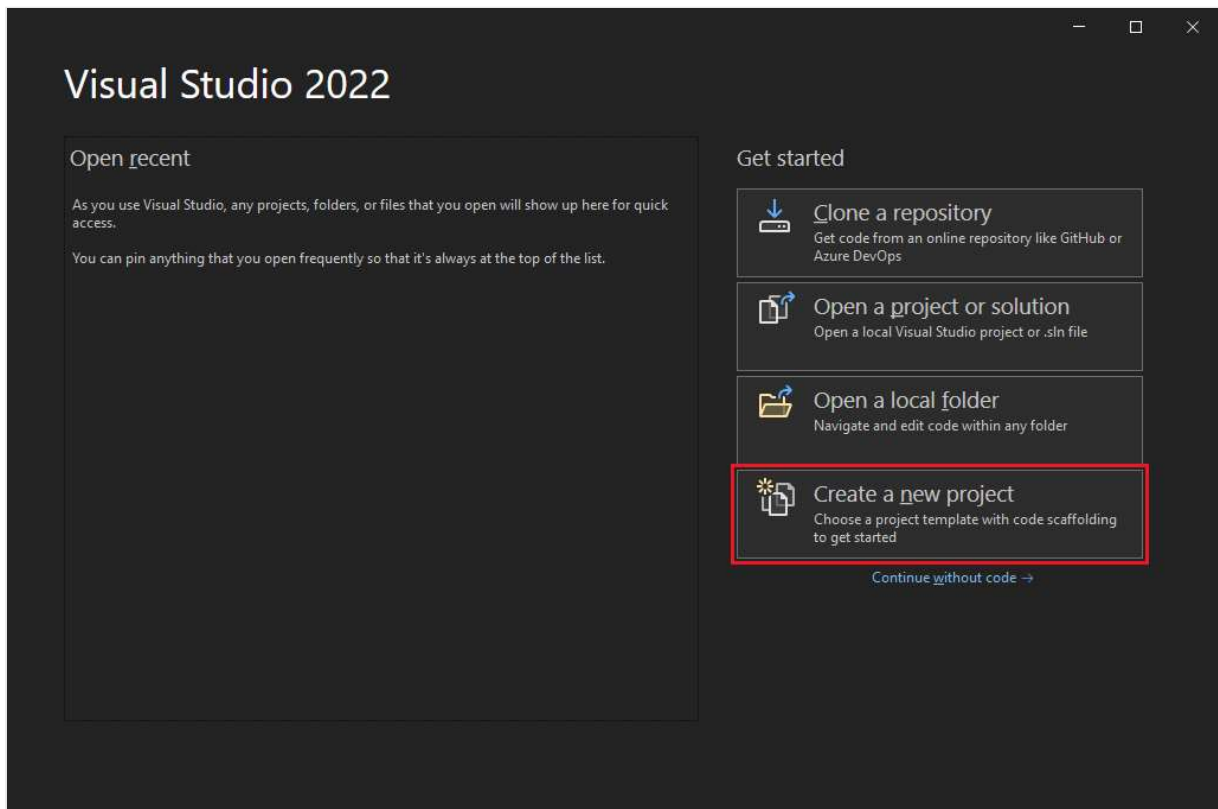✔ Add a button to the form
✔ Add a label and code
✔ Run the application

Prerequisites

- You need Visual Studio to complete this tutorial. Visit the Visual Studio downloads page for a free version.
- The **.NET desktop development** workload. To verify or install this workload in Visual Studio, select **Tools** > **Get Tools and Features**. For more information, see Change workloads or individual components.
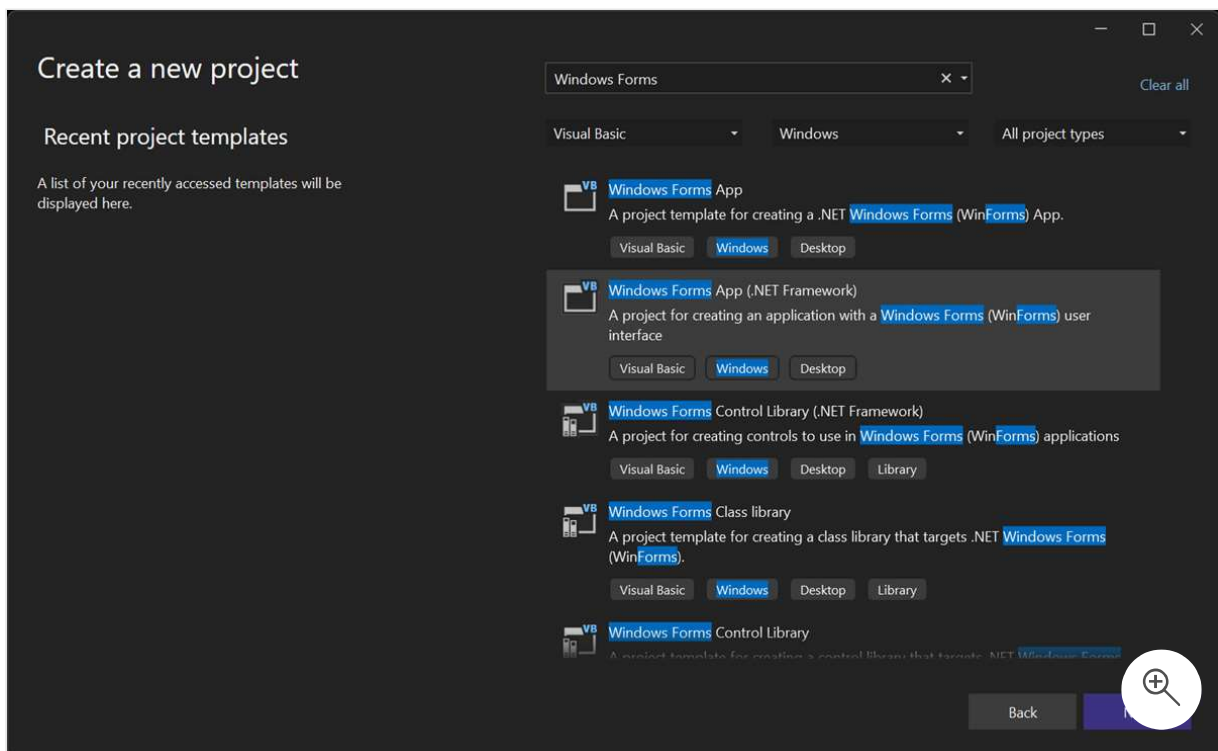
Create a project

Create a Visual Basic application project. The project type comes with all the template files that you need.

1. Open Visual Studio.

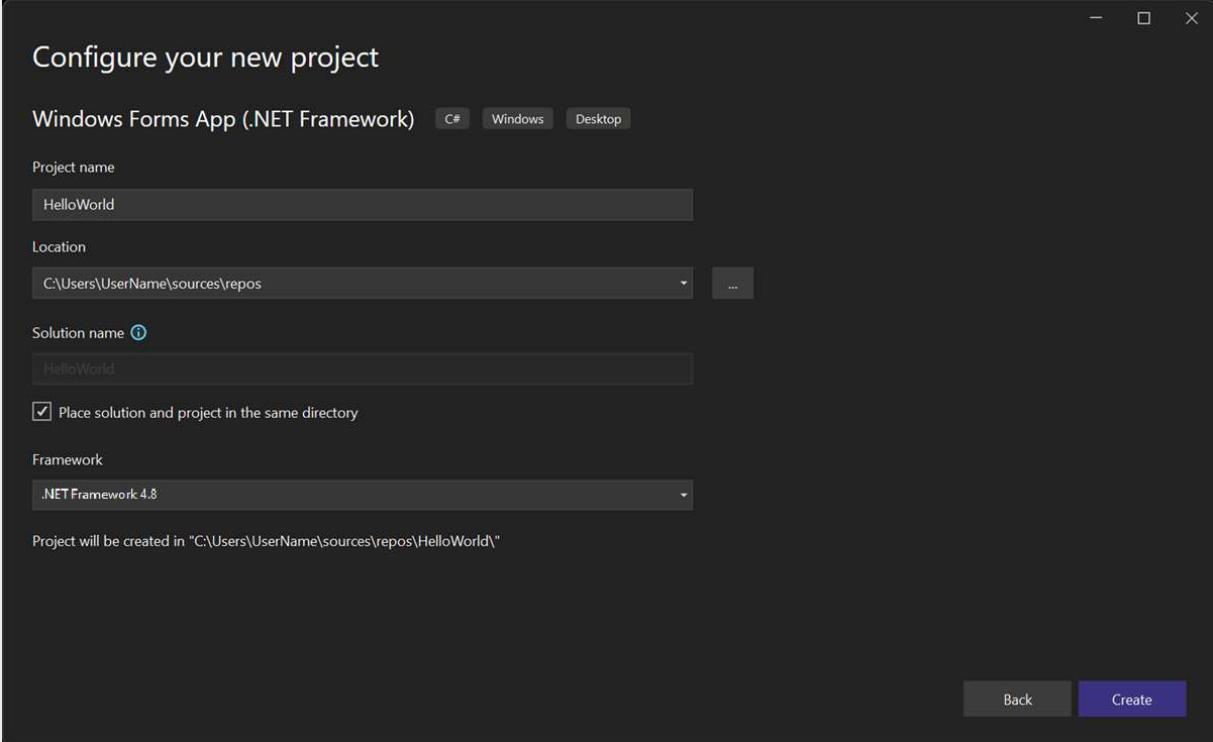2. On the start window, choose **Create a new project**.

3. On the **Create a new project** window, select the **Windows Forms App (.NET Framework)** template for Visual Basic.

You can refine your search to quickly get to the template you want. For example, enter *Windows Forms App* in the search box. Next, select **Visual Basic** from the language list, and then **Windows** from the platform list.

4. In the **Configure your new project** window, enter *HelloWorld* as the **Project name**. Then, select **Create**.
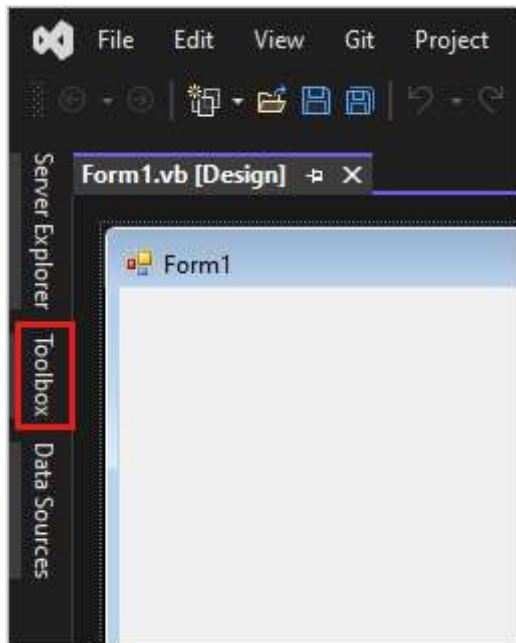

Visual Studio opens your new project.

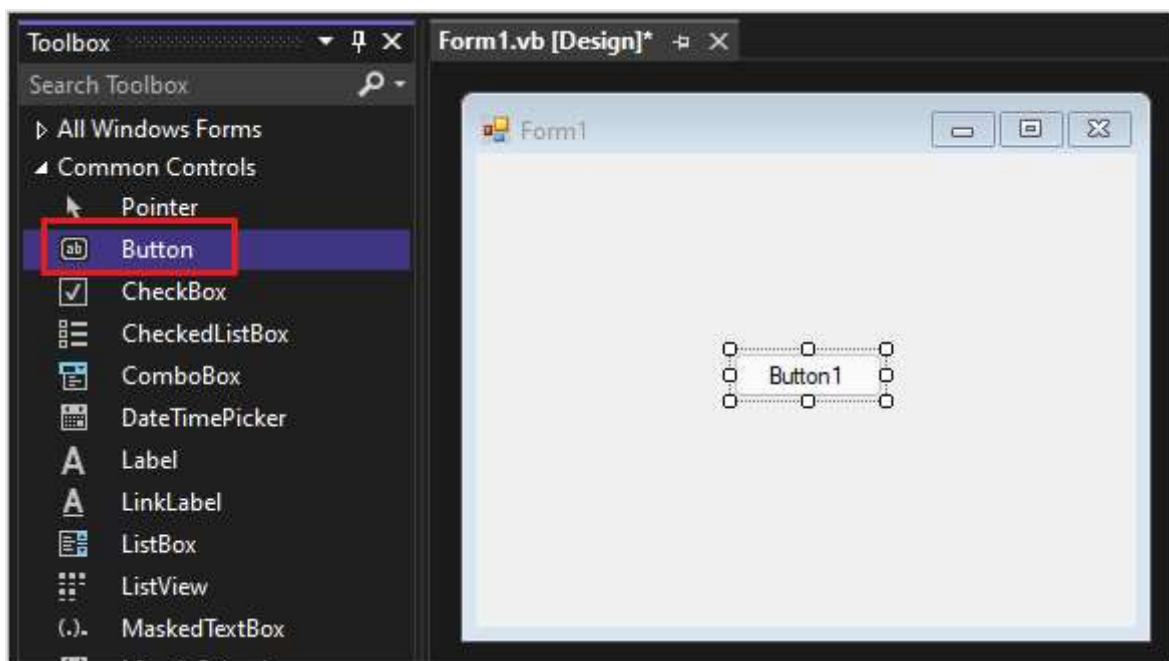# Add a button to the form

After you select your Visual Basic project template and name your file, Visual Studio opens a form for you. A form is a Windows user interface. You create a *Hello World* application by adding controls to the form.

1. On the left side of the Visual Studio IDE, select the **Toolbox** tab. If you don't see it, select **View** > **Toolbox** from the menu bar or **Ctrl+Alt+X**.
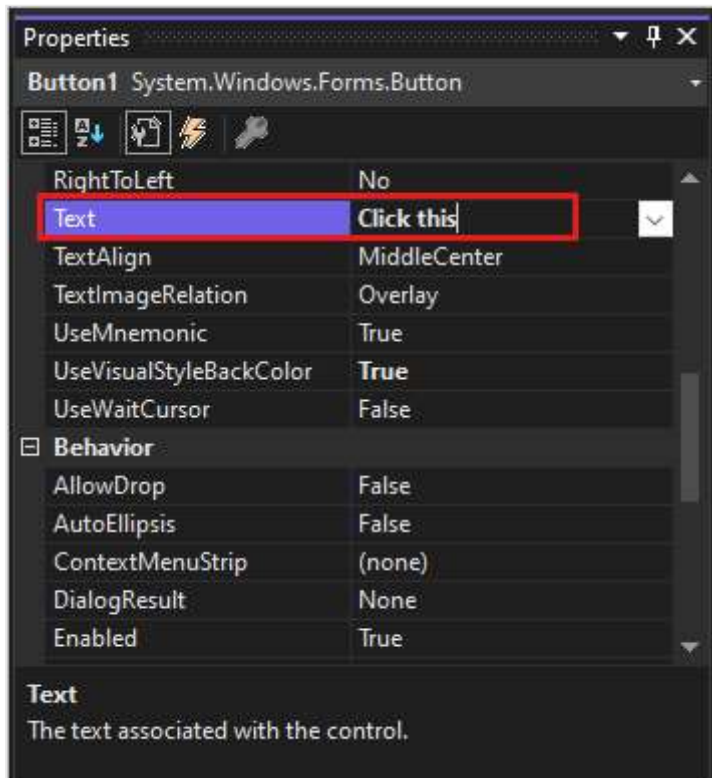
If you want, select the **Pin** icon to dock the **Toolbox** window.

2. Select the **Button** control and then drag it onto the form.
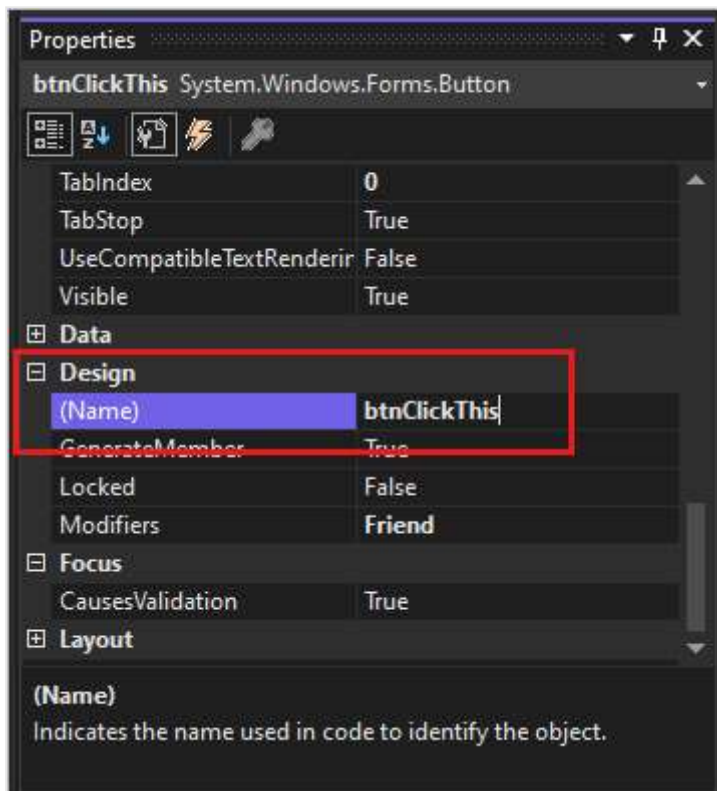

3. In the **Appearance** section of the **Properties** window, for **Text**, type *Click this*, and then select **Enter**.

If you don't see the **Properties** window, you can open it from the menu bar. Select **View** > **Properties Window** or **F4**.

4. In the **Design** section of the **Properties** window, change the name from **Button1** to *btnClickThis*, and then select **Enter**.

> ⓘ **Note**
>
> If you've alphabetized the list in the **Properties** window, **Button1** appears in the **(DataBindings)** section, instead.
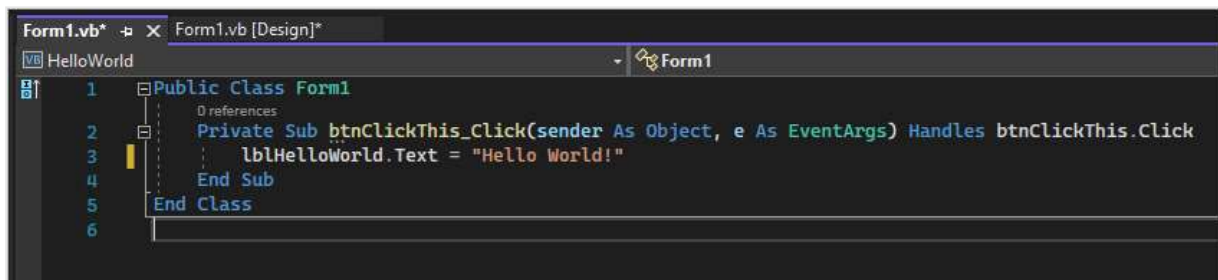
Add a label and code

After you add a button control to create an action, add a label control to send text to.

1. Select the **Label** control in the **Toolbox** window, and then drag it onto the form. Place it beneath the **Click this** button.

2. In either the **Design** section or the **(DataBindings)** section of the **Properties** window, change the name **Label1** to *lblHelloWorld*, and then press **Enter**.

3. In the **Form1.vb [Design]** window, double-click the **Click this** button to open the **Form1.vb** window.

   Another option is to expand **Form1.vb** in **Solution Explorer**. Then select **Form1**.
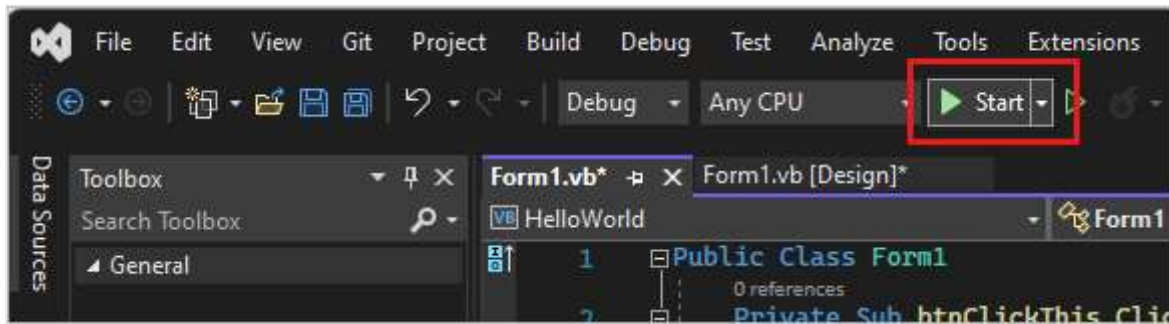
4. In the **Form1.vb** window, between the `Private Sub` and `End Sub` lines, enter *lblHelloWorld.Text = "Hello World!"* as shown in the following screenshot:


```vb
Public Class Form1

    Private Sub btnClickThis_Click(sender As Object, e As EventArgs) Handles btnClickThis.Click
        lblHelloWorld.Text = "Hello World!"
    End Sub
End Class
```
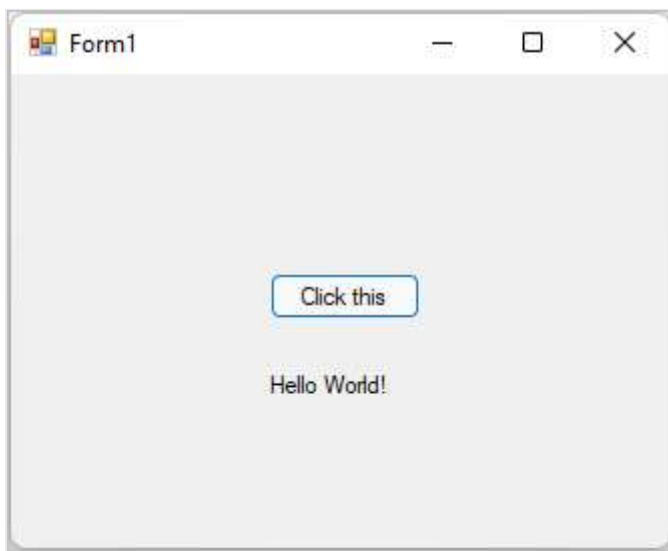
Run the application

Your application is ready to build and run.

1. Select **Start** to run the application.

Several things happen. In the Visual Studio IDE, the **Diagnostics Tools** window opens, and an **Output** window opens. Outside of the IDE, a **Form1** dialog box appears. It includes your **Click this** button and text that says **Label1**.

2. Select the **Click this** button in the **Form1** dialog box.

The **Label1** text changes to **Hello World!**.

3. Close the **Form1** dialog box to stop running the app.

Related content

To learn more about Windows Forms, continue with the following tutorial:

Tutorial: Create a picture viewer

Or try these other tutorials:

- Visual Basic tutorials
- C# tutorials
- C++ tutorials

Feedback

Was this page helpful?   👍 Yes   👎 No