Create your first WPF application in Visual Studio 2019

This tutorial shows you how to develop your first Windows Presentation Foundation (WPF) desktop application that includes the elements that are common to most WPF applications: Extensible Application Markup Language (XAML) markup, code-behind, application definitions, controls, layout. To develop the application, you'll use Visual Studio.

You learn how to:

- Create a WPF project.
- Use XAML to design the appearance of the application's user interface (UI).
- Write code to build the application's behavior.
- Add controls and create the layout to compose the application UI.

Prerequisites

- Visual Studio 2019 with the **.NET desktop development** workload installed.

  For more information about installing the latest version of Visual Studio, see Install Visual Studio.

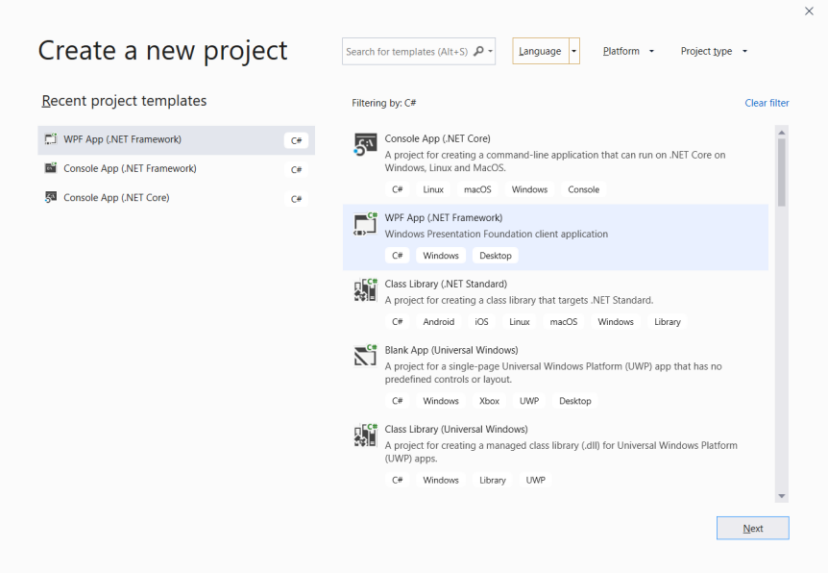
Create the application project

The first step is to create the application infrastructure, which includes an application definition, two pages, and an image.

1. Create a new WPF Application project in Visual C# named **`FirstWPF`**:
   a. Open Visual Studio and select **Create a new project** under the **Get started** menu.

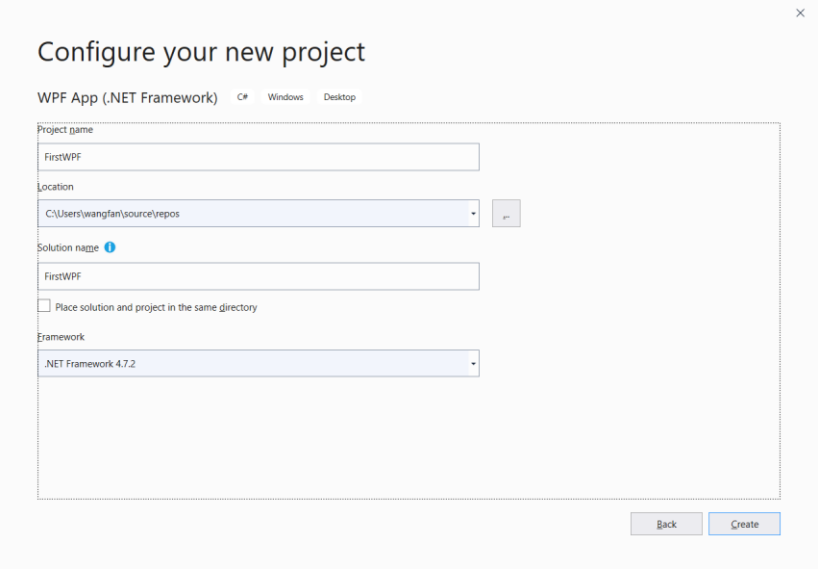
      The **Create a new project** dialog opens.

   b. In the **Language** dropdown, select **C#** .
   c. Select the **WPF App (.NET Framework)** template and then select **Next**.

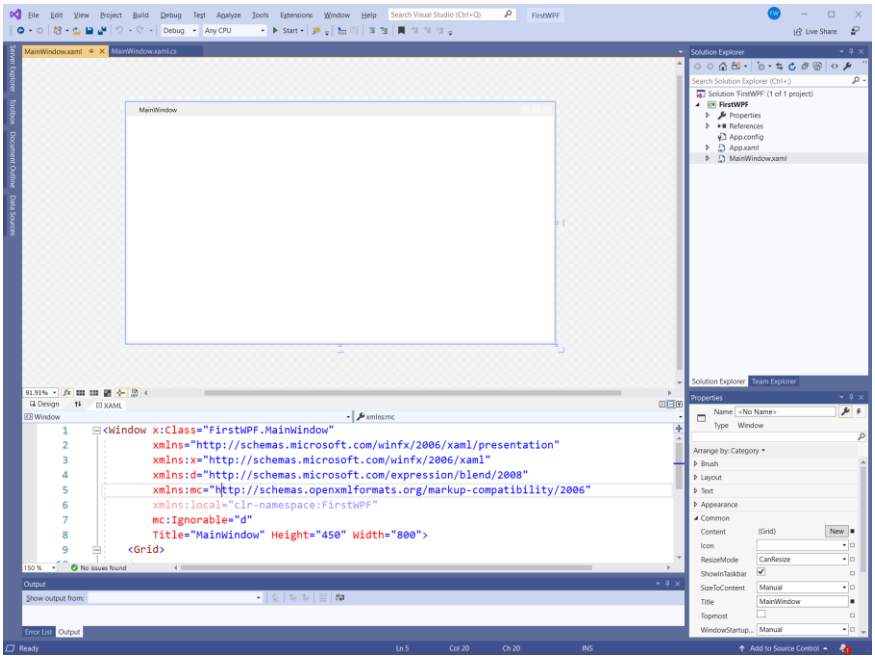


The **Configure your new project** dialog opens.

d. Enter the project name **`FirstWPF`** and then select **Create**.



Visual Studio creates the project and opens the designer for the default application window named **MainWindow.xaml**.



2. Open *App.xaml* .

This XAML file defines a WPF application and any application resources. You also use this file to specify the UI, in this case *MainWindow.xaml*, that automatically shows when the application starts.

```
<Application x:Class="FirstWPF.App"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:local="clr-namespace:FirstWPF"
             StartupUri="MainWindow.xaml">
    <Application.Resources>

    </Application.Resources>
</Application>
```

3. Open *MainWindow.xaml*.

This XAML file is the main window of your application and displays content created in pages. The Window class defines the properties of a window, such as its title, size, or icon, and handles events, such as closing or hiding.

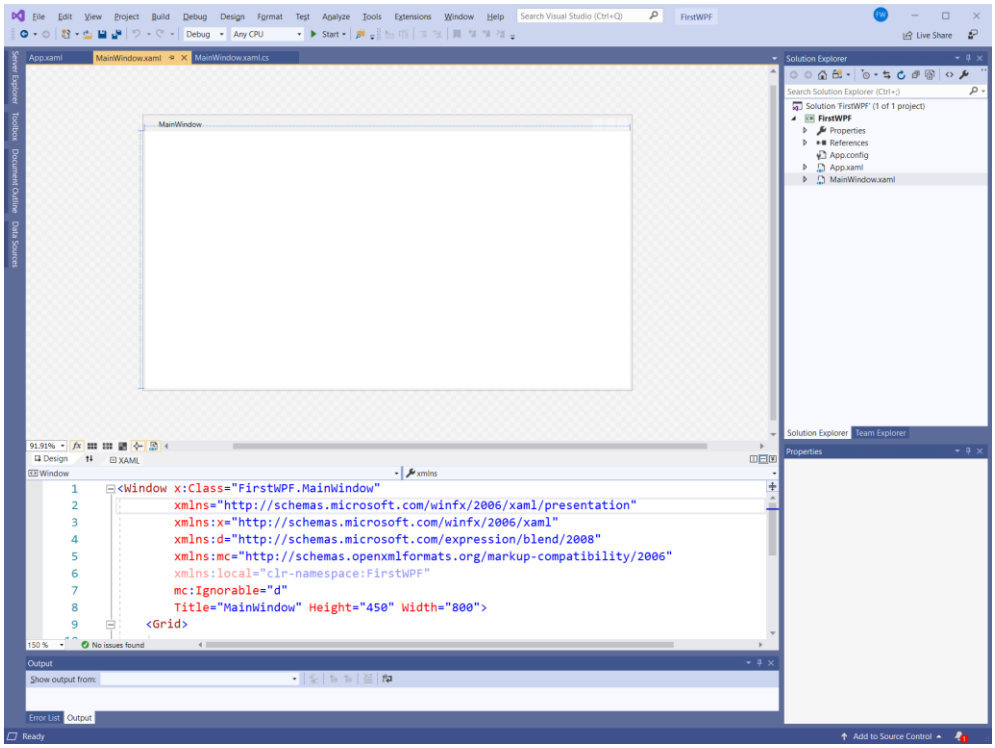
```
<Window x:Class="FirstWPF.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:FirstWPF"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">

    <Grid>
```

4. Add controls

Controls are UI objects that allow users to interact with your application. For more information, see Controls.

The WPF Designer shows a design view and a XAML view of *MainWindow.xaml* in a split view.



a. Use the **Toolbox** to drag and drop UI elements into the design view. Add *TextBox*, *TextBlock*, *RadioButton* and *Button* into the design view.

   Add Names to the two *RadioButton* and *Button*.

b. Subscribing to Events

You can subscribe to a control's event by using either XAML or code, but you can only handle an event in code. The following example shows how to subscribe to the **Click** event of a Button.

XAML

```
<Button Grid.Row="2" Grid.ColumnSpan="2" Name="submitName"
Click="submit_Click"
  Background="Green">View message</Button>
```

The following example handles the Click event of a Button.

C#

```
void submit_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Hello, " + firstName.Text + " " + lastName.Text);
}
```

The following illustration shows when control names (*RadioButton, Button*) and events (*Button*) are added.

5. Add code behind
   Now we can work on the code behind. In **Solution Explorer**, open up *MainWindow.xaml.cs*, which is nested under *MainWindow.xaml*. Type in the following code.

```
namespace FirstWPF
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            if (HelloRB.IsChecked == true)
            {
                MessageBox.Show("Hello");
            }
            else
            {
                ByeRB.IsChecked = true;
                MessageBox.Show("Good Bye!");
            }
        }
    }
}
```

6. To build and run the application, press Ctrl + **F5** or select **Start Without Debugging** from the **Debug** menu. Test with the Button and RadionButtons. The following illustration shows a sample test result.