

📄

📝

ChatGPT

Code Copilot

Explore GPTs

Today

WPF Basic Controls Example

WPF Controls and Styles

WPF Trigger Types Explained

Yesterday

WPF Login Form Layout

WPF vs WinForms Comparison

WPF Overview and Features

Upgrade plan

Get GPT-4, DALL-E, and more

ChatGPT

Write the Code For WPF Basic Controls code

Below is a simple WPF (Windows Presentation Foundation) application that demonstrates basic controls like buttons, textboxes, labels, and a simple event handler for button clicks.

XAML Code (MainWindow.xaml)

This defines the layout and appearance of the UI elements.

```
<?xml version="1.0" encoding="utf-8" />
<Window x:Class="BasicControlsApp.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Basic Controls Demo" Height="300" Width="400">
    <Grid>
        <!-- Label -->
        <Label Content="Enter your name:" HorizontalAlignment="Left" Margin="30,40,0,0" VerticalAlignment="Top" />
        <!-- TextBox -->
        <TextBox Name="NameTextBox" HorizontalAlignment="Left" Height="23" Margin="140,40,0,0" VerticalAlignment="Top" />
        <!-- Button -->
        <Button Content="Submit" HorizontalAlignment="Left" Margin="140,80,0,0" VerticalAlignment="Top" />
        <!-- Label to display result -->
        <Label Name="ResultLabel" Content="" HorizontalAlignment="Left" Margin="30,120,0,0" VerticalAlignment="Top" />
    </Grid>
</Window>
```

C# Code-behind (MainWindow.xaml.cs)

This handles the button click event and updates the UI.

```
using System.Windows;

namespace BasicControlsApp
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void SubmitButton_Click(object sender, RoutedEventArgs e)
        {
            // Get the input from the TextBox and display it in the ResultLabel
            string userName = NameTextBox.Text;
            if (!string.IsNullOrEmpty(userName))
            {
                ResultLabel.Content = $"Hello, {userName}!";
            }
            else
            {
                ResultLabel.Content = "Please enter your name.";
            }
        }
    }
}
```

Explanation:

1. **Label:** Displays static text or can dynamically display content.
2. **TextBox:** Allows users to enter input.
3. **Button:** Triggers the click event. When clicked, it runs the `SubmitButton_Click` event handler.
4. **Event Handler (SubmitButton_Click):** This gets the user input from the `TextBox` and displays a message in the `ResultLabel`.

Steps to run:

1. Create a new WPF project in Visual Studio.
2. Replace the default XAML and C# code with the code above.
3. Build and run the application.

You will have a window that takes user input and displays a message when the "Submit" button is clicked.

🔍 📄 🗑️ 🔄 📌

Message ChatGPT

ChatGPT can make mistakes. Check important info.