

[](https://www.tutorialr.com/tutorials/)

Windows App SDK

Custom Button





# Custom Button

**Custom Button** shows how to create a customised **Style** for a **Button** using **Windows App SDK**

## Step 1

Follow **Setup and Start** on how to get **Setup** and **Install** what you need for **Visual Studio 2022** and **Windows App SDK**.

|  |  |
| --- | --- |
| In **Windows 11** choose **Start** and then find or search for **Visual Studio 2022** and then select it. | Text  Description automatically generated |
| Once **Visual Studio 2022** has started select **Create a new project**. | **Graphical user interface, text  Description automatically generated** |
| Then choose the **Blank App, Packages (WinUI in Desktop)** and then select **Next**. | **Graphical user interface, text  Description automatically generated** |
| After that in **Configure your new project** type in the **Project name** as *CustomButton*, then select a Location and then select **Create** to start a new **Solution**. | **Graphical user interface, text, application, email  Description automatically generated** |

## Step 2

|  |  |
| --- | --- |
| Then in **Visual Studio** within **Solution** **Explorer** for the **Solution** double-click on **App.xaml** to see the **XAML** for the **Project**. |  |

## Step 3

In the **XAML** for **App.xaml** below the **Comment** of **<!-- Other app resources here -->** type in the following **XAML**:

<Style x:Key="CustomButton" TargetType="Button">

<Setter Property="Background">

<Setter.Value>

<LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">

<GradientStop Offset="0" Color="LightSalmon"/>

<GradientStop Offset="1" Color="DarkSalmon"/>

</LinearGradientBrush>

</Setter.Value>

</Setter>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Grid>

<!-- Visual State Groups -->

<!-- Content -->

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

This **XAML** is part of a **Style** of **CustomButton** that will be used to target a **Button** which will contain some **Visual State Groups** and **Content** for the **Custom Button**.

## Step 4

While still in the **XAML** for **App.xaml** below the **Comment** of **<!-- Visual State Groups -->** type the following **XAML**:

<VisualStateManager.VisualStateGroups>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal"/>

<VisualState x:Name="PointerOver">

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="Inner"

Storyboard.TargetProperty="(ScaleTransform.ScaleY)">

<DiscreteObjectKeyFrame KeyTime="0" Value="-1"/>

</ObjectAnimationUsingKeyFrames>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="Outer"

Storyboard.TargetProperty="(ScaleTransform.ScaleY)">

<DiscreteObjectKeyFrame KeyTime="0" Value="1"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</VisualState>

<VisualState x:Name="Pressed">

<Storyboard>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="Inner"

Storyboard.TargetProperty="(ScaleTransform.ScaleY)">

<DiscreteObjectKeyFrame KeyTime="0" Value="1"/>

</ObjectAnimationUsingKeyFrames>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="Outer"

Storyboard.TargetProperty="(ScaleTransform.ScaleY)">

<DiscreteObjectKeyFrame KeyTime="0" Value="-1"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</VisualState>

</VisualStateGroup>

</VisualStateManager.VisualStateGroups>

This **XAML** is for the **Visual State Groups** that will represent the **States** for the **Button** including how it will behave when **Normal** and when it is **Pressed** for the **Custom Button**.

## Step 5

Then in the **XAML** for **App.xaml** below the **Comment** of **<!-- Content -->** type the following **XAML:**

<Ellipse Margin="4" Fill="{TemplateBinding Background}"

RenderTransformOrigin="0.5,0.5">

<Ellipse.RenderTransform>

<ScaleTransform ScaleY="1" x:Name="Outer"/>

</Ellipse.RenderTransform>

</Ellipse>

<Ellipse Margin="20" Fill="{TemplateBinding Background}"

RenderTransformOrigin="0.5,0.5">

<Ellipse.RenderTransform>

<ScaleTransform ScaleY="-1" x:Name="Inner"/>

</Ellipse.RenderTransform>

</Ellipse>

<ContentPresenter x:Name="Content"

HorizontalAlignment="Center"

VerticalAlignment="Center"/>

This **XAML** is the **Content** for the layout of a **Button** when the **Style** is applied for the **Custom Button**.

## Step 6

|  |  |
| --- | --- |
| Within **Solution** **Explorer** for the **Solution** double-click on **MainWindow.xaml** to see the **XAML** for the **Main Window**. |  |

## Step 7

In the **XAML** for **MainWindow.xaml** there be some **XAML** for a **StackPanel**, this should be **Removed** by removing the following:

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center" VerticalAlignment="Center">

<Button x:Name="myButton" Click="myButton\_Click">Click Me</Button>

</StackPanel>

## Step 8

While still in the **XAML** for **MainWindow.xaml** above **</Window>**, type in the following **XAML**:

<Button HorizontalAlignment="Center" Content="Button"

Height="200" Width="200" Style="{StaticResource CustomButton}"/>

This **XAML** contains a **Button** with **Style** set to the **StaticResource** of **CustomButton** from **App.xaml**.

## Step 9

|  |  |
| --- | --- |
| Then, within **Solution** **Explorer** for the **Solution** select the arrow next to **MainWindow.xaml** then double-click on **MainWindow.xaml.cs** to see the **Code** for the **Main Window**. |  |

## Step 10

In the **Code** for **MainWindow.xaml.cs** there be a **Method** of **myButton\_Click(...)** this should be **Removed** by removing the following:

private void myButton\_Click(object sender, RoutedEventArgs e)

{

myButton.Content = "Clicked";

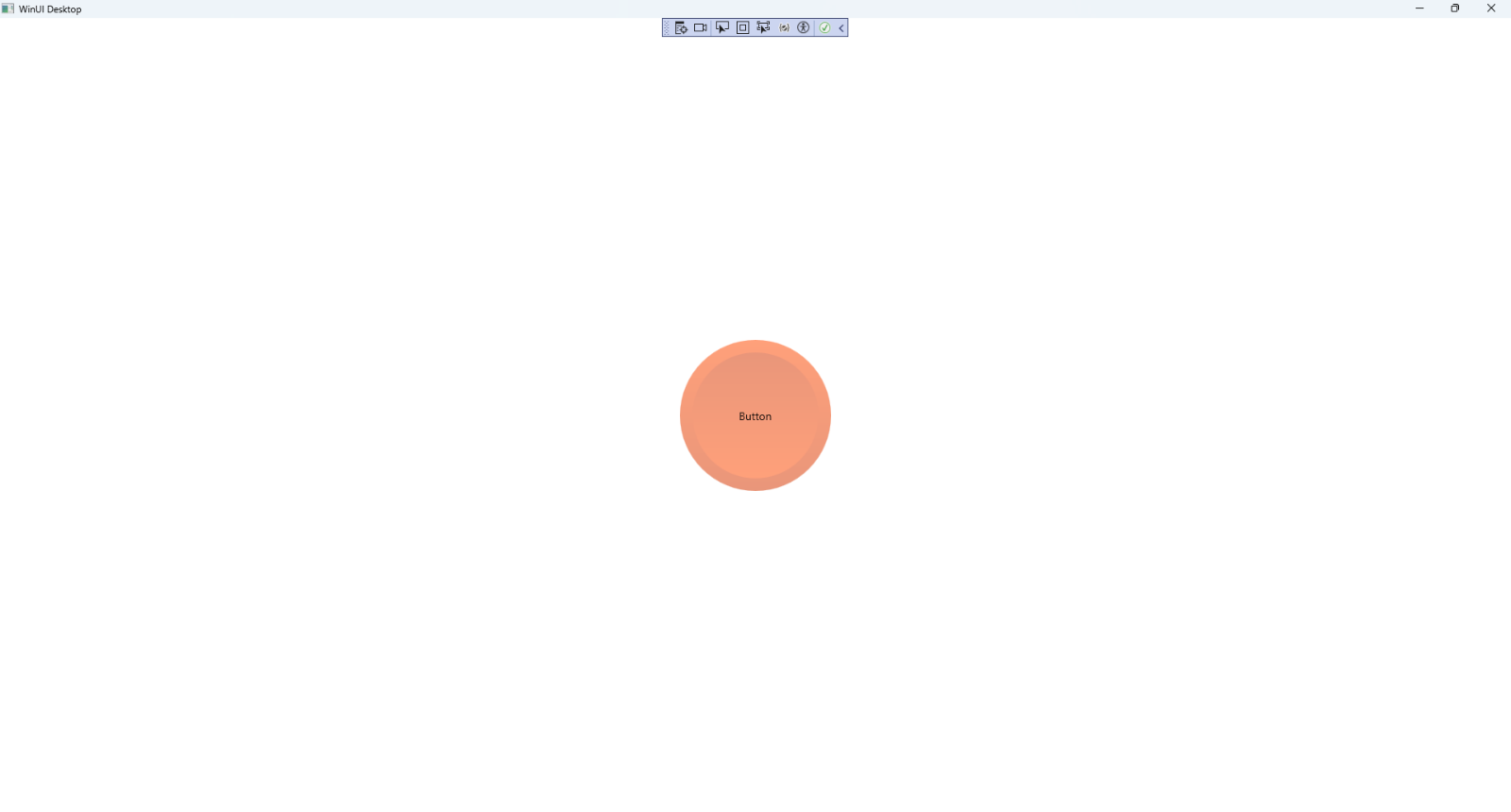
}

## Step 11

|  |  |
| --- | --- |
| That completes the **Windows App SDK** application. In **Visual Studio 2022** from the **Toolbar** select **CustomButton (Package)** to **Start** the application. |  |

## Step 12

Once running you will see the **Custom Button** displayed.

****

## Step 13

|  |  |
| --- | --- |
| To **Exit** the **Windows App SDK** application, select the **Close** button from the top right of the application as that concludes this **Tutorial** for **Windows App SDK** from [tutorialr.com](https://tutorialr.com)! |  |