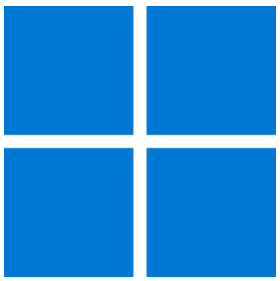
tutorial

Windows App SDK



Custom TextBox

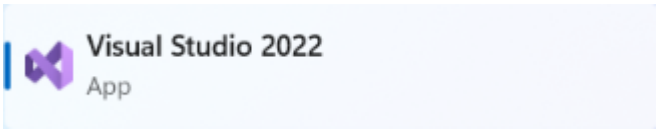
Custom TextBox

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

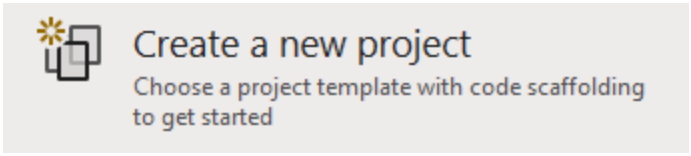
Step 1

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

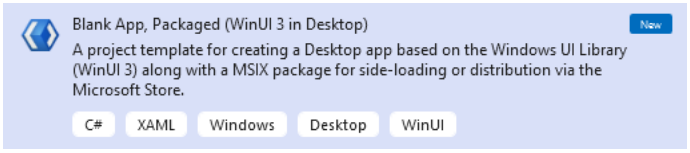
In **Windows 11** choose **Start** and then find or search for **Visual Studio 2022** and then select it.



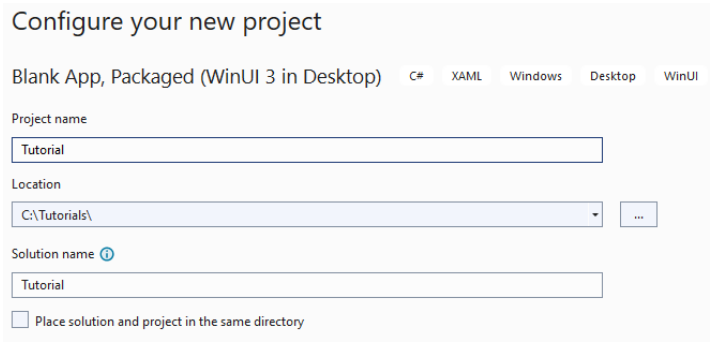
Once **Visual Studio 2022** has started select **Create a new project**.



Then choose the **Blank App, Packages (WinUI in Desktop)** and then select **Next**.

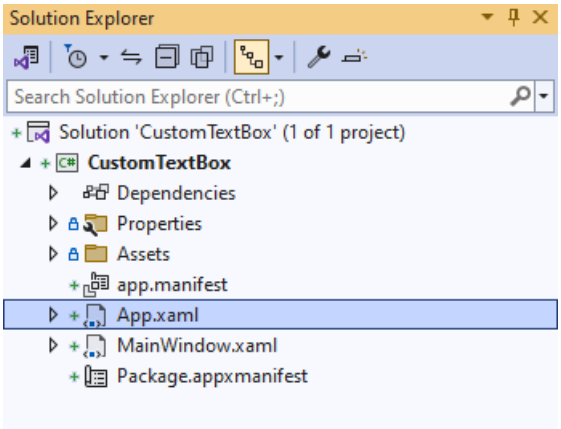


After that in **Configure your new project** type in the **Project name** as *CustomTextBox*, then select a Location and then select **Create** to start a new **Solution**.



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** for the **`Style`** of **`CustomTextBox`** that will be used to target a **`TextBox`**:

```
<Style x:Key="CustomTextBox" TargetType="TextBox">
    <Setter Property="MinWidth"
    Value="{ThemeResource TextControlThemeMinWidth}"/>
    <Setter Property="MinHeight"
    Value="{ThemeResource TextControlThemeMinHeight}"/>
    <Setter Property="Foreground" Value="Gold"/>
    <Setter Property="Background"
    Value="{ThemeResource TextControlBackground}"/>
    <Setter Property="SelectionHighlightColor"
    Value="{ThemeResource TextControlSelectionHighlightColor}"/>
    <Setter Property="BorderThickness" Value="2"/>
    <Setter Property="FontFamily"
    Value="{ThemeResource ContentControlThemeFontFamily}"/>
    <Setter Property="FontSize"
    Value="{ThemeResource ControlContentThemeFontSize}"/>
    <Setter Property="ScrollViewer.HorizontalScrollMode" Value="Auto"/>
    <Setter Property="ScrollViewer.VerticalScrollMode" Value="Auto"/>
    <Setter Property="ScrollViewer.HorizontalScrollBarVisibility" Value="Hidden"/>
    <Setter Property="ScrollViewer.VerticalScrollBarVisibility" Value="Hidden"/>
    <Setter Property="ScrollViewer.IsDeferredScrollingEnabled" Value="False"/>
    <Setter Property="Padding" Value="{ThemeResource TextControlThemePadding}"/>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="TextBox">
                <Grid>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="*"/>
                        <ColumnDefinition Width="Auto"/>
                    </Grid.ColumnDefinitions>
                    <Grid.RowDefinitions>
                        <RowDefinition Height="Auto"/>
                        <RowDefinition Height="*"/>
                    </Grid.RowDefinitions>
                    <VisualStateManager.VisualStateGroups>
                        <VisualStateGroup x:Name="CommonStates">
                            <!-- Visual State Disabled -->

                            <!-- Visual State Normal & Pointer Over -->

                            <!-- Visual State Focused -->

                        </VisualStateGroup>
                    </VisualStateManager.VisualStateGroups>
                    <!-- Content -->

                </Grid>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

Step 4

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

```
<VisualState x:Name="Disabled">
    <Storyboard>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Foreground"
            Storyboard.TargetName="HeaderContentPresenter">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlHeaderForegroundDisabled}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Background"
            Storyboard.TargetName="BorderElement">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlBackgroundDisabled}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="BorderBrush"
            Storyboard.TargetName="BorderElement">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlBorderBrushDisabled}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Foreground"
            Storyboard.TargetName="ContentElement">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlForegroundDisabled}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Foreground"
            Storyboard.TargetName="PlaceholderTextContentPresenter">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlPlaceholderForegroundDisabled}"/>
        </ObjectAnimationUsingKeyFrames>
    </Storyboard>
</VisualState>
```

This **XAML** is for the **Visual State** that will represent the **State** of **`Disabled`** for the **`TextBox`** used in the **Custom TextBox**.

Step 5

While still in the **XAML** for **App.xaml** below the **Comment** of **`<!-- Visual State Normal & Pointer Over -->`** type the following **XAML**:

```
<VisualState x:Name="Normal"/>
<VisualState x:Name="PointerOver">
    <Storyboard>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="BorderBrush"
            Storyboard.TargetName="BorderElement">
            <DiscreteObjectKeyFrame KeyTime="0" Value="GoldenRod"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Background"
            Storyboard.TargetName="BorderElement">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlBackgroundPointerOver}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Foreground"
            Storyboard.TargetName="PlaceholderTextContentPresenter">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlPlaceholderForegroundPointerOver}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Foreground"
            Storyboard.TargetName="ContentElement">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlForegroundPointerOver}"/>
        </ObjectAnimationUsingKeyFrames>
    </Storyboard>
</VisualState>
```

This **XAML** is for the **Visual State** that will represent the **States** of **`Normal`** and **`PointerOver`** for the **`TextBox`** used in the **Custom TextBox**.

Step 6

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

```
<VisualState x:Name="Focused">
    <Storyboard>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Foreground"
            Storyboard.TargetName="PlaceholderTextContentPresenter">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlPlaceholderForegroundFocused}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Background"
            Storyboard.TargetName="BorderElement">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlBackgroundFocused}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="BorderBrush"
            Storyboard.TargetName="BorderElement">
            <DiscreteObjectKeyFrame KeyTime="0" Value="Gold"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="Foreground"
            Storyboard.TargetName="ContentElement">
            <DiscreteObjectKeyFrame KeyTime="0"
            Value="{ThemeResource TextControlForegroundFocused}"/>
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames
            Storyboard.TargetProperty="RequestedTheme"
            Storyboard.TargetName="ContentElement">
            <DiscreteObjectKeyFrame KeyTime="0" Value="Light"/>
        </ObjectAnimationUsingKeyFrames>
    </Storyboard>
</VisualState>
```

This **XAML** is for the **Visual State** that will represent the **State** of **`Focused`** for the **`TextBox`** used in the **Custom TextBox**.

Step 7

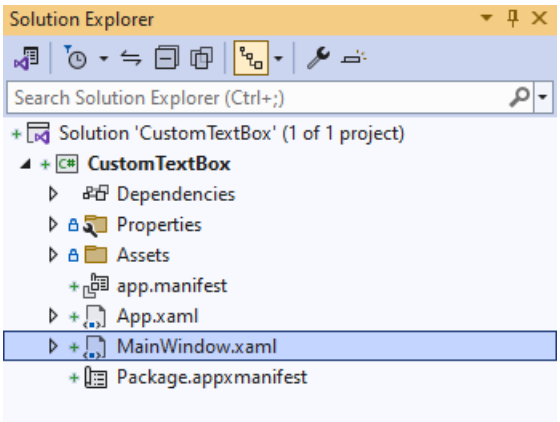
Then in the **XAML** for **App.xaml** below the **Comment** of **`<!-- Content -->`** type the following **XAML:**

```
<Border x:Name="BorderElement" Grid.Row="1"
Grid.RowSpan="1" Grid.ColumnSpan="2" CornerRadius="15"
BorderBrush="Salmon" Background="LightSalmon"
BorderThickness="{TemplateBinding BorderThickness}" />
<ScrollViewer x:Name="ContentElement" Grid.Row="1"
AutomationProperties.AccessibilityView="Raw"
IsTabStop="False" ZoomMode="Disabled"
HorizontalScrollMode="{TemplateBinding
ScrollViewer.HorizontalScrollMode}"
HorizontalScrollBarVisibility="{TemplateBinding
ScrollViewer.HorizontalScrollBarVisibility}"
IsHorizontalRailEnabled="{TemplateBinding
ScrollViewer.IsHorizontalRailEnabled}"
IsVerticalRailEnabled="{TemplateBinding
ScrollViewer.IsVerticalRailEnabled}"
IsDeferredScrollingEnabled="{TemplateBinding
ScrollViewer.IsDeferredScrollingEnabled}"
Margin="{TemplateBinding BorderThickness}"
Padding="{TemplateBinding Padding}"
VerticalScrollBarVisibility="{TemplateBinding
ScrollViewer.VerticalScrollBarVisibility}"
VerticalScrollMode="{TemplateBinding
ScrollViewer.VerticalScrollMode}" />
<ContentPresenter x:Name="PlaceholderTextContentPresenter"
Grid.Row="1" Grid.ColumnSpan="2"
Content="{TemplateBinding PlaceholderText}"
Foreground="Gold" IsHitTestVisible="False"
Margin="{TemplateBinding BorderThickness}"
Padding="{TemplateBinding Padding}"
TextWrapping="{TemplateBinding TextWrapping}"/>
```

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

Step 8

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



Step 9

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 10

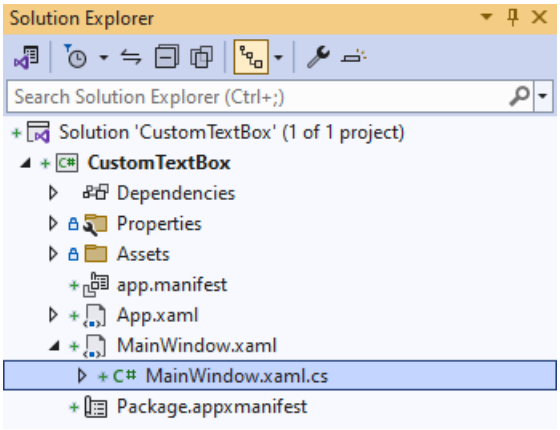
While still in the **XAML** for **MainWindow.xaml** above **`</Window>`**, type in the following **XAML**:

```
<TextBox Margin="50" Text="TextBox" VerticalAlignment="Center"
Style="{StaticResource CustomTextBox}"/>
```

This **XAML** contains a **`TextBox`** with **`Style`** set to the **`StaticResource`** of **`CustomTextBox`** from **App.xaml**.

Step 11

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**.



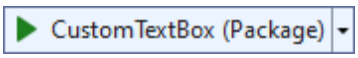
Step 12

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";
}
```

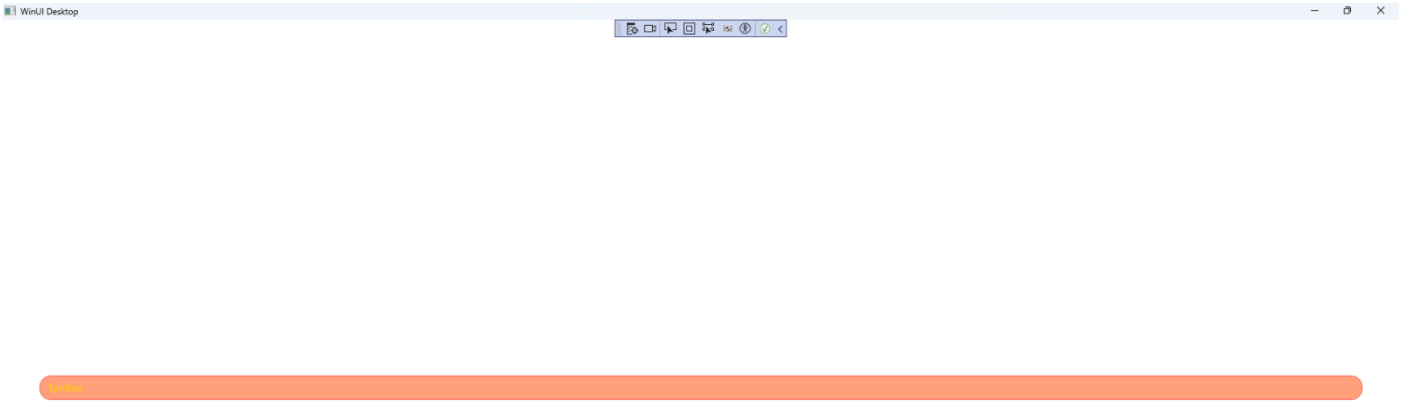
Step 13

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



Step 14

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



Step 15

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!