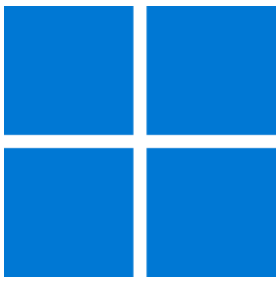
tutorial

# Windows App SDK



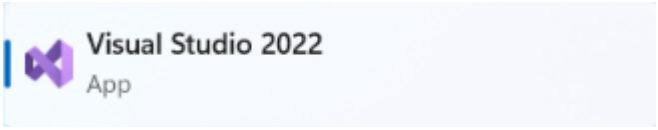
# Shade Effect

**Shade Effect** shows how you can use **DropShadow** with an Element to create a **Shade Effect** in an Application using the **Windows App SDK**.

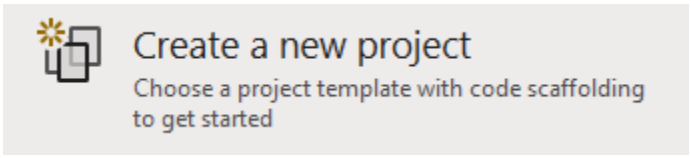
## Step 1

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

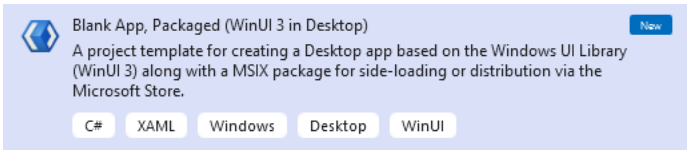
In **Windows 11** choose **Start** and then find or search for **Visual Studio 2022** and then select it.



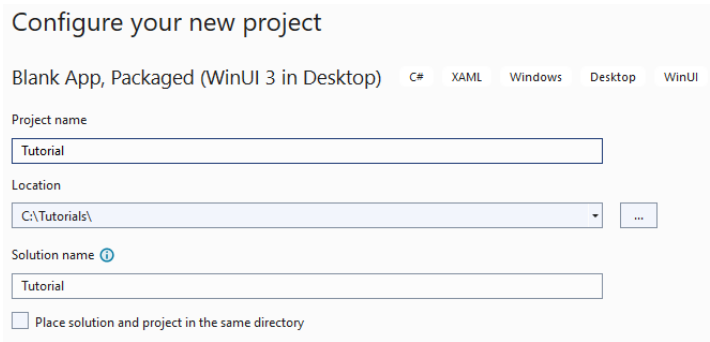

Once **Visual Studio 2022** has started select **Create a new project**.




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

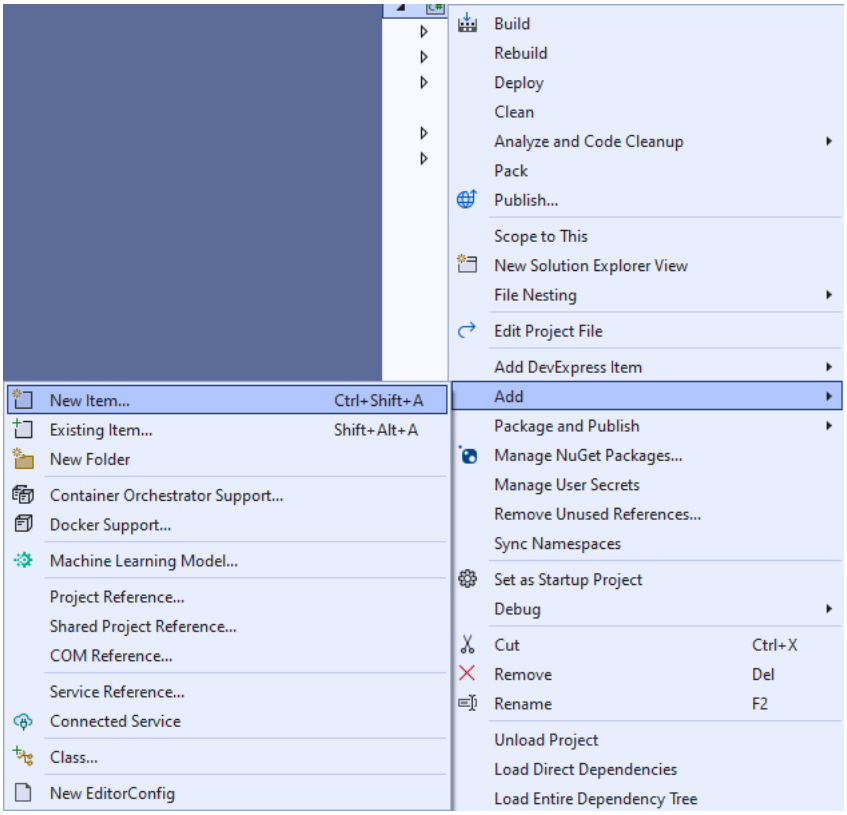



After that in **Configure your new project** type in the **Project name** as *ShadeEffect*, then select a Location and then select **Create** to start a new **Solution**.



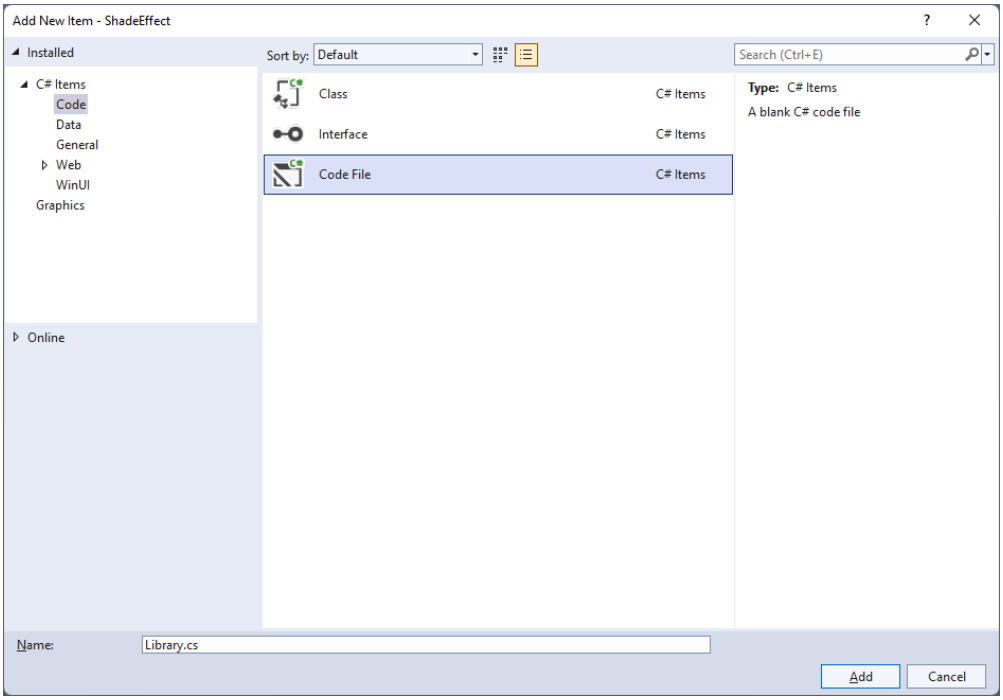
## Step 2

Then in **Visual Studio** within **Solution Explorer** for the **Solution**, right click on the **Project** shown below the **Solution** and then select **Add** then **New Item...**




## Step 3

Then in **Add New Item** from the **C# Items** list, select **Code** and then select **Code File** from the list next to this, then type in the name of *Library.cs* and then **Click** on **Add**.



## Step 4

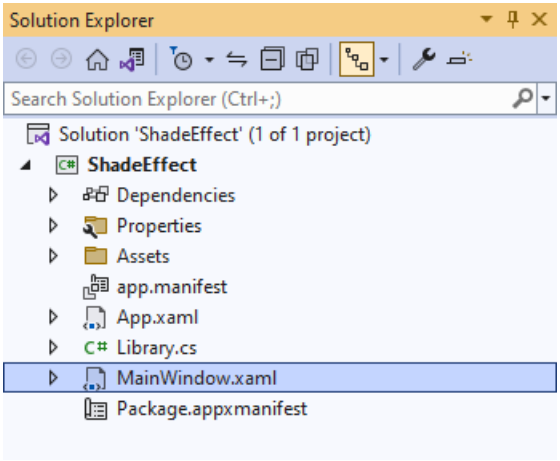
You will now be in the **View** for the **Code** of *Library.cs*, within this type the following **Code**:

```
using Microsoft.UI;
using Microsoft.UI.Composition;
using Microsoft.UI.Xaml;
using Microsoft.UI.Xaml.Hosting;
using Microsoft.UI.Xaml.Shapes;

internal class Library
{
    private SpriteVisual _shade;

    public void SetShade(Shape shape, FrameworkElement element)
    {
        var compositor = ElementCompositionPreview
            .GetElementVisual(shape).Compositor;
        _shade = compositor.CreateSpriteVisual();
        _shade.Size = new System.Numerics.Vector2(
            (float)shape.ActualWidth,
            (float)shape.ActualHeight);
        DropShadow shadow = compositor.CreateDropShadow();
        shadow.Color = Colors.Black;
        shadow.Offset = new System.Numerics.Vector3(10, 10, 0);
        shadow.Mask = shape.GetAlphaMask();
        _shade.Shadow = shadow;
        ElementCompositionPreview.SetElementChildVisual(element, _shade);
    }

    public void ClearShade()
    {
        if (_shade != null)
            _shade.Shadow = null;
    }
}
```

The **`Class`** that has been defined in *Library.cs* has a **Member** for a **`SpriteVisual`** then there is a **Method** of **`SetShade`** which will create a **Shadow Effect** for a **`Shape`** in a **`FrameworkElement`** by first creating an **Compositior** with **`ElementCompositionPreview`** you'll also notice the use of **`var`**, which means the type of the value doesn't need to be explicitly specified, instead it will be **Inferred**. Then a **`Compositor`** is used with **`CreateSpriteVisual`** is configured where various values are set for a **`DropShadow`** to display it as needed as a **`Shadow`** with the **`SpriteVisual`**. The other method of **`ClearShade`** is used to remove the **Shade Effect** of the **`DropShadow`** from the **`SpriteVisual`**.

## Step 5

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




## Step 6

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 7

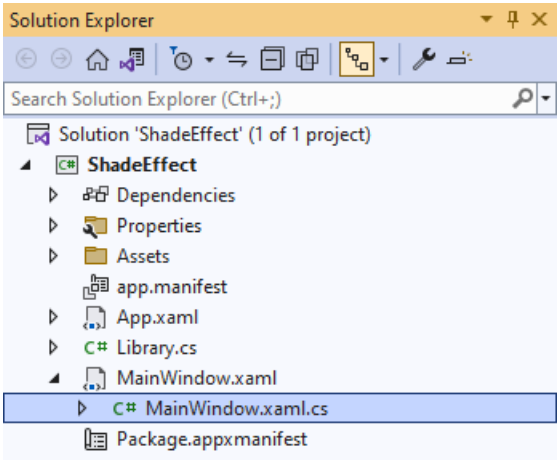
While still in the **XAML** for **MainWindow.xaml** above **`</Window>`**, type in the following **XAML**:

```
<Grid>
    <Viewbox Margin="25">
        <Grid Margin="50" Height="400" Width="400">
            <Border x:Name="ShadowElement"/>
            <Rectangle Name="Display" Stretch="Uniform" Fill="#FF5c2d91"/>
        </Grid>
    </Viewbox>
    <CommandBar Name="Options" VerticalAlignment="Bottom"
        HorizontalAlignment="Stretch">
        <AppBarButton Icon="Accept" Label="Accept" Click="Accept_Click"/>
        <AppBarButton Icon="Cancel" Label="Clear" Click="Clear_Click"/>
    </CommandBar>
</Grid>
```

This **XAML** features a **`Grid`** with a **`ViewBox`** which is used to **Scale** elements, then within this is a **`Border`** that will form the **Shade Effect** for the **`Rectangle`** which is a **`FrameworkElement`**. Then there is a **`CommandBar`** with an **`AppBarButton`** to apply the **Shade Effect** of *Accept* and another to remove it of *Clear*.

## Step 8

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 9

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 10

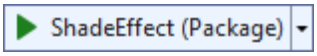
Once **`myButton_Click(...)`** has been removed, type in the following **Code** below the end of the **Constructor** of **`public MainWindow() { ... }`**:

```
private readonly Library _library = new();

private void Accept_Click(object sender, RoutedEventArgs e)
{
    _library.SetShade(Display, ShadowElement);
}

private void Clear_Click(object sender, RoutedEventArgs e)
{
    _library.ClearShade();
}
```

The **Method** of **`Accept_Click`** will call the **Method** within *Library.cs* of **`SetShade`** from an **Instance** of **`Library`** called **`_library`** created with **`new()`** and **`Clear_Click`** will call the **Method** of **`ClearShade`**.

## Step 11

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




## Step 12

Once running you should see a **`Rectangle`** and **`CommandBar`** with the *Accept* and *Clear* options.

## Step 13

You can select *Apply* to set the **Shade Effect** and *Clear* to remove the **Shade Effect**


## Step 14

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!