Tutorialr.com

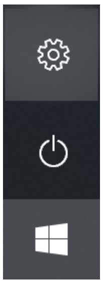
Universal Windows Platform

tutorialr.com

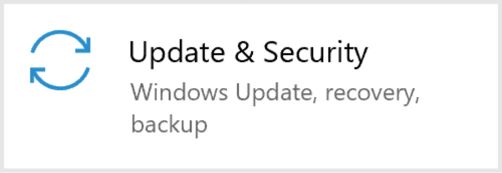
Universal Windows Platform - Setup and Start

Setup

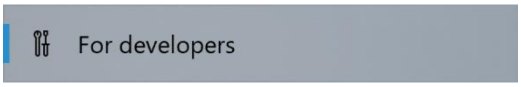
You will need to enable **Developer Mode** in **Windows 10** if this has not been done already by completing the following:



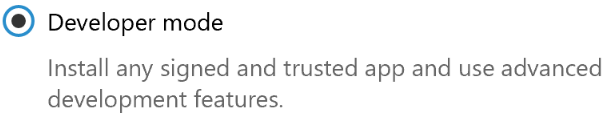
Choose **Start** then **Settings** from the **Start Menu**



Next from **Windows Settings** choose **Update & Security**



Then from **Update & Security** choose **For developers**



● Developer mode

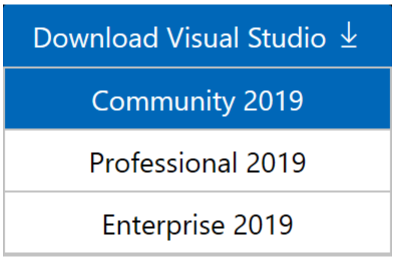
Install any signed and trusted app and use advanced development features.

Finally make sure **Developer mode** is selected

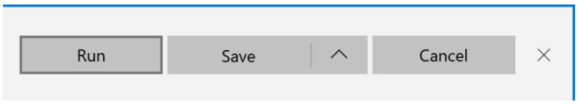
Universal Windows Platform - Setup and Start

Install

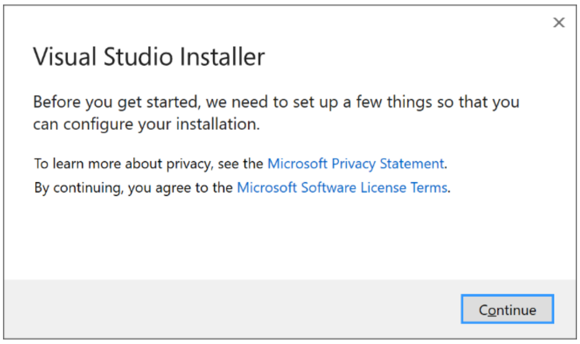
You will need to install **Visual Studio 2019 Community**, if this has not been done already you just need to do the following:



Visit **VisualStudio.com** and then from the **Visual Studio IDE** section choose **Download Visual Studio** then **Community 2019**



Next on the **Thank you for downloading Visual Studio** page when the download prompt appears, select **Run**

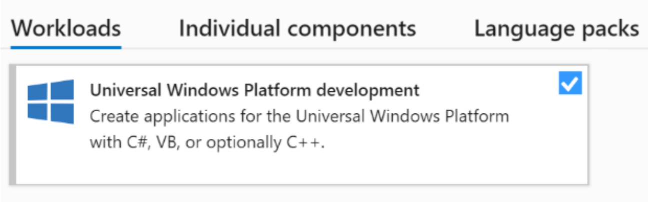


Visual Studio Installer

Before you get started, we need to set up a few things so that you can configure your installation.

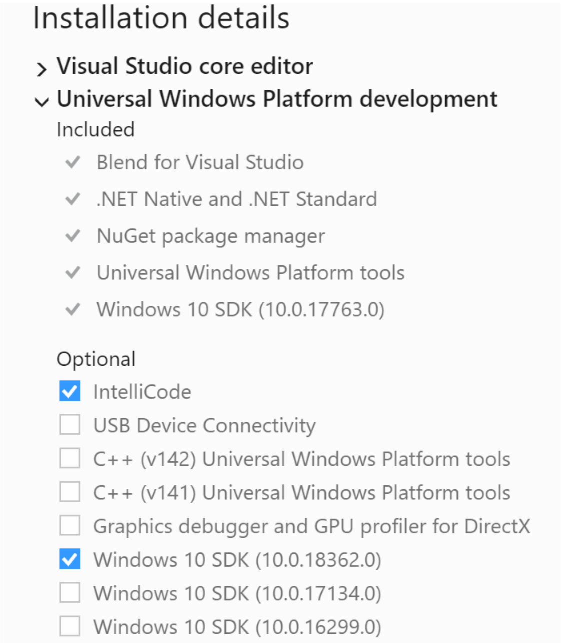
To learn more about privacy, see the Microsoft Privacy Statement.
By continuing, you agree to the Microsoft Software License Terms.

Once downloaded, this should start the **Visual Studio Installer** and select **Continue** to begin the installation



☑ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Next once ready select **Universal Windows Platform development** from the **Workloads** section



Installation details

> Visual Studio core editor

∨ Universal Windows Platform development

Included

- ✓ Blend for Visual Studio
- ✓ .NET Native and .NET Standard
- ✓ NuGet package manager
- ✓ Universal Windows Platform tools
- ✓ Windows 10 SDK (10.0.17763.0)

Optional

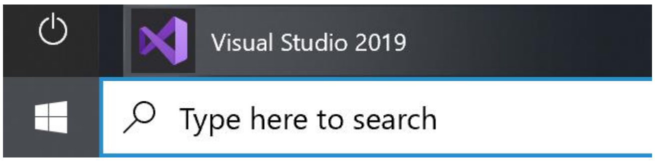
- ☑ IntelliCode
- □ USB Device Connectivity
- □ C++ (v142) Universal Windows Platform tools
- □ C++ (v141) Universal Windows Platform tools
- □ Graphics debugger and GPU profiler for DirectX
- ☑ Windows 10 SDK (10.0.18362.0)
- □ Windows 10 SDK (10.0.17134.0)
- □ Windows 10 SDK (10.0.16299.0)

Then make sure the latest **Windows 10 SDK** has been selected from **Installation details** if not already included then select **Install** and follow any instructions to complete the installation

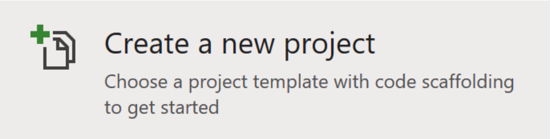
Universal Windows Platform - Setup and Start

Start

You will need to open **Visual Studio 2019 Community** and **Create a new project**, if this has not been done already you just need to do the following:

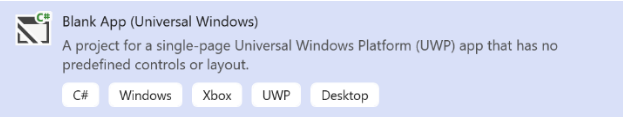


In **Windows 10** choose **Start**, and then from the **Start Menu** find and select **Visual Studio 2019**


Create a new project

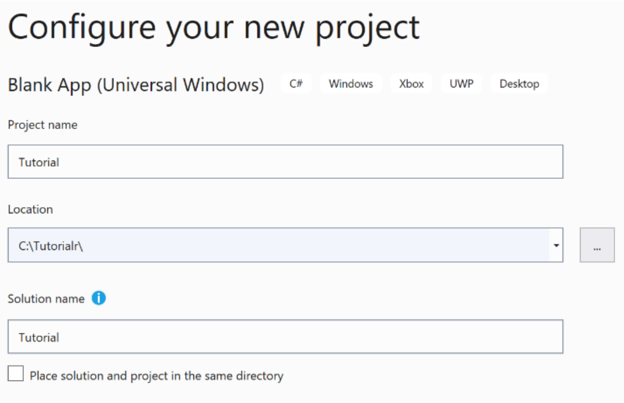
Choose a project template with code scaffolding to get started

Once done, from the **Get started** screen for **Visual Studio 2019** select **Create a new project**


Blank App (Universal Windows)

A project for a single-page Universal Windows Platform (UWP) app that has no predefined controls or layout.

Choose **Blank App (Universal Windows)** and select **Next**


Configure your new project

Blank App (Universal Windows) C# Windows Xbox UWP Desktop

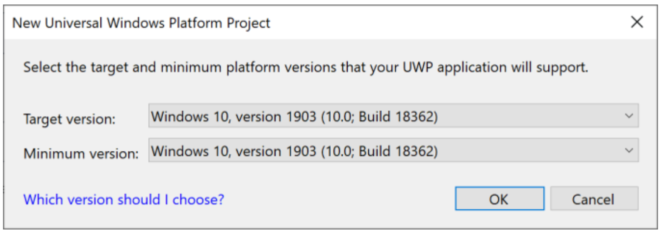
Project name: Tutorial

Location: C:\Tutorialr\

Solution name: Tutorial

□ Place solution and project in the same directory

Then in **Configure your new project** enter a **Project name** and **Location** and select **Create**


New Universal Windows Platform Project

Select the target and minimum platform versions that your UWP application will support.

Target version: Windows 10, version 1903 (10.0; Build 18362)

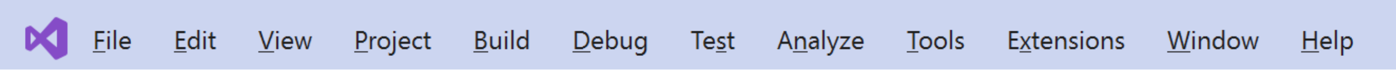
Minimum version: Windows 10, version 1903 (10.0; Build 18362)

Finally, in **New Universal Windows Platform Project** pick the latest **Target version** and **Minimum version** and select **OK**

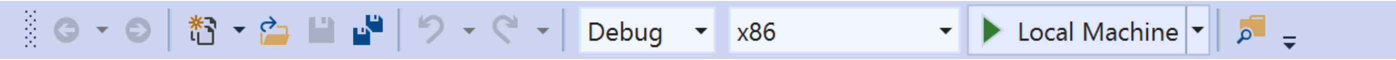
Universal Windows Platform - Setup and Start

Guide

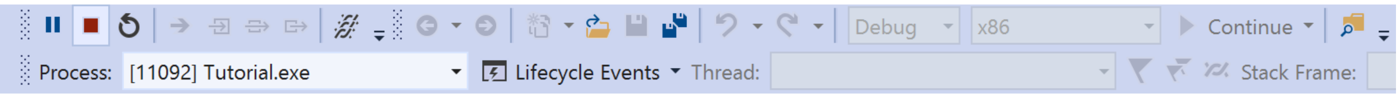
In **Visual Studio 2019** there is a **Menu** at the top



Below this is the **Toolbar**, options there include **Local Machine** to start debugging

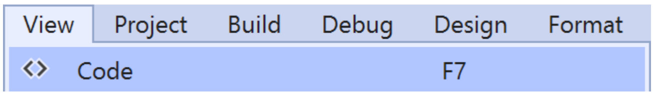


When an Application is running, the **Toolbar** will change to include other options including **Stop** to finish Debugging.

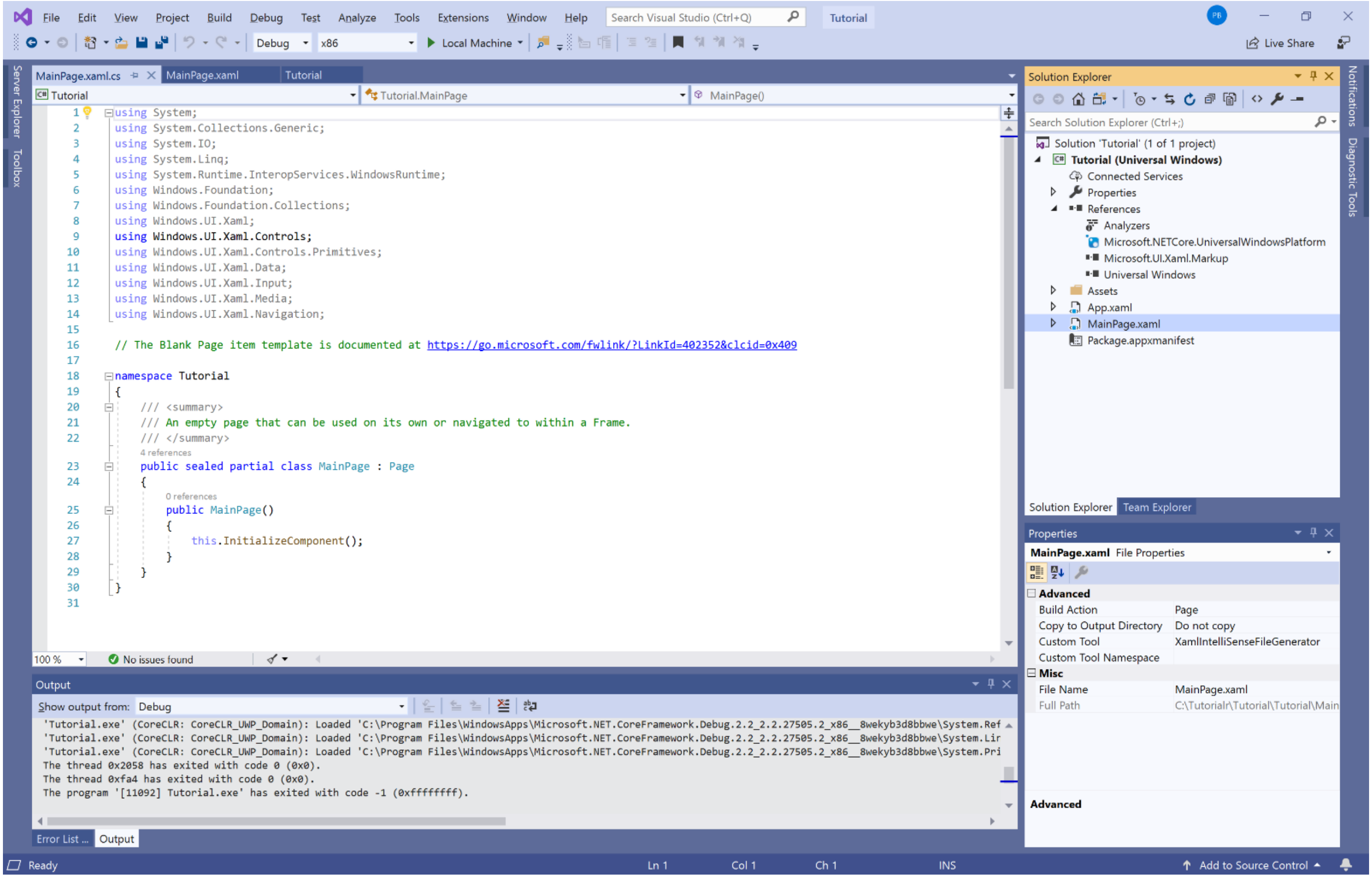


Process: [11092] Tutorial.exe

After a **Solution** or **Project** has been opened or created, choose **View** then **Code** from the **Menu**



This will display the **Code** View for the code in a **Class** or for a **XAML** page such as **MainPage.xaml.cs**



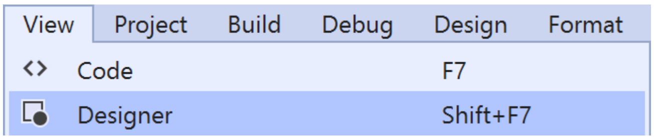
```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

// The Blank Page item template is documented at https://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x409

namespace Tutorial
{
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a Frame.
    /// </summary>
    4 references
    public sealed partial class MainPage : Page
    {
        0 references
        public MainPage()
        {
            this.InitializeComponent();
        }
    }
}
```

```
'Tutorial.exe' (CoreCLR: CoreCLR_UWP_Domain): Loaded 'C:\Program Files\WindowsApps\Microsoft.NET.CoreFramework.Debug.2.2_2.2.27505.2_x86__8wekyb3d8bbwe\System.Ref
'Tutorial.exe' (CoreCLR: CoreCLR_UWP_Domain): Loaded 'C:\Program Files\WindowsApps\Microsoft.NET.CoreFramework.Debug.2.2_2.2.27505.2_x86__8wekyb3d8bbwe\System.Lir
'Tutorial.exe' (CoreCLR: CoreCLR_UWP_Domain): Loaded 'C:\Program Files\WindowsApps\Microsoft.NET.CoreFramework.Debug.2.2_2.2.27505.2_x86__8wekyb3d8bbwe\System.Pri
The thread 0x2058 has exited with code 0 (0x0).
The thread 0xfa4 has exited with code 0 (0x0).
The program '[11092] Tutorial.exe' has exited with code -1 (0xffffffff).
```

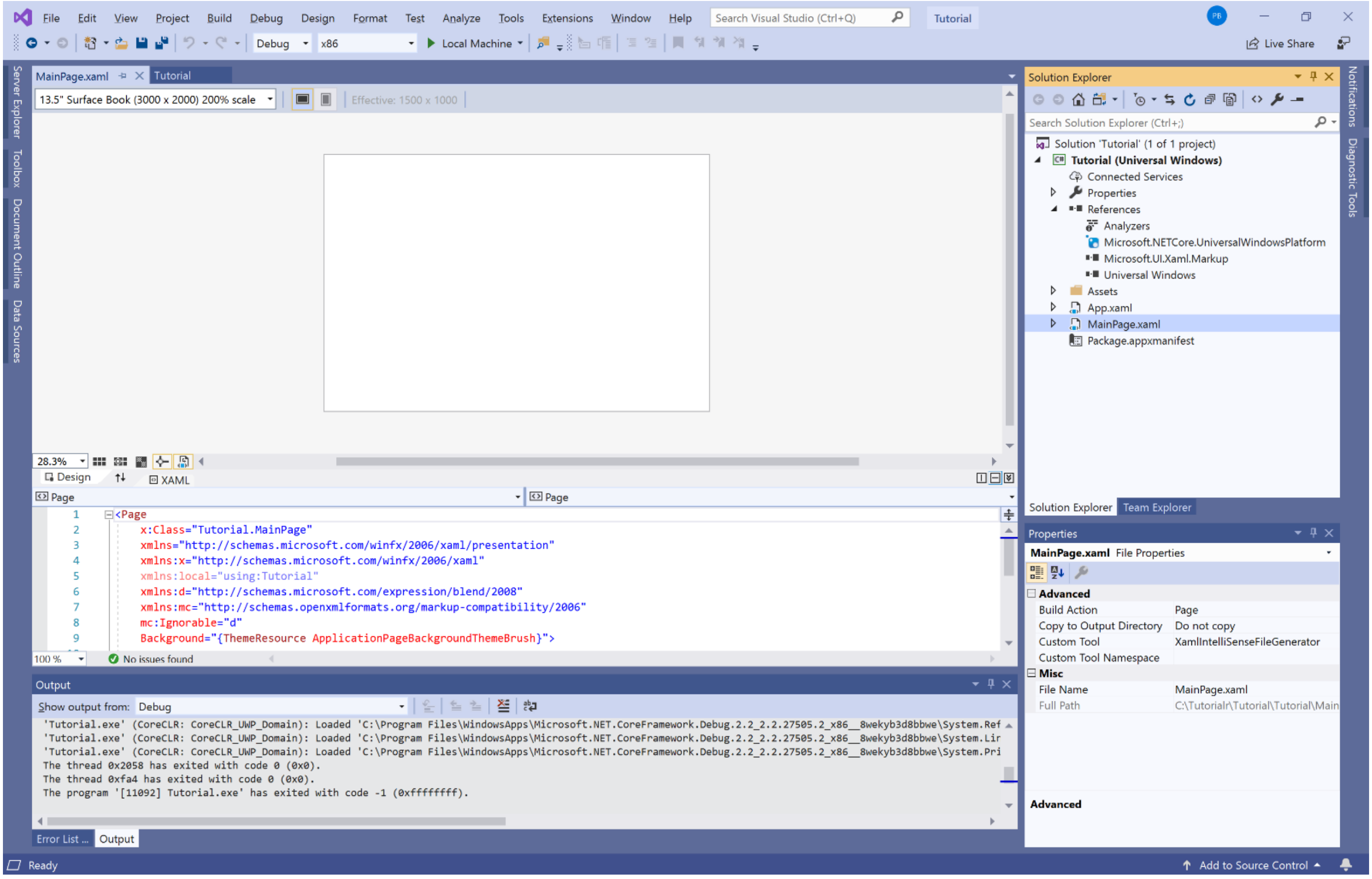
Build Action	Page
Copy to Output Directory	Do not copy
Custom Tool	XamlIntelliSenseFileGenerator
Custom Tool Namespace	
File Name	MainPage.xaml
Full Path	C:\Tutorial\Tutorial\Tutorial\Main

Universal Windows Platform - Setup and Start

Also, after a **Solution** or **Project** has been opened or created, choose **View** then **Designer** from the **Menu**



This will display the **Design** and **XAML** views for a **XAML** page such as **MainPage.xaml**



```
<Page
    x:Class="Tutorial.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:Tutorial"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
```

Show output from: Debug

```
'Tutorial.exe' (CoreCLR: CoreCLR_UWP_Domain): Loaded 'C:\Program Files\WindowsApps\Microsoft.NET.CoreFramework.Debug.2.2_2.2.27505.2_x86__8wekyb3d8bbwe\System.Ref
'Tutorial.exe' (CoreCLR: CoreCLR_UWP_Domain): Loaded 'C:\Program Files\WindowsApps\Microsoft.NET.CoreFramework.Debug.2.2_2.2.27505.2_x86__8wekyb3d8bbwe\System.Lir
'Tutorial.exe' (CoreCLR: CoreCLR_UWP_Domain): Loaded 'C:\Program Files\WindowsApps\Microsoft.NET.CoreFramework.Debug.2.2_2.2.27505.2_x86__8wekyb3d8bbwe\System.Pri
The thread 0x2058 has exited with code 0 (0x0).
The thread 0xfa4 has exited with code 0 (0x0).
The program '[11092] Tutorial.exe' has exited with code -1 (0xffffffff).
```

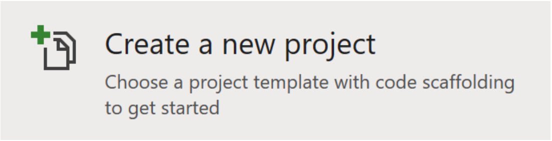
MainPage.xaml File Properties

Advanced	
Build Action	Page
Copy to Output Directory	Do not copy
Custom Tool	XamlIntelliSenseFileGenerator
Custom Tool Namespace	
Misc	
File Name	MainPage.xaml
Full Path	C:\Tutorialr\Tutorial\Tutorial\Main

Universal Windows Platform - Hello World

Hello World, is used to introduce many new programming language examples in this case it is an introduction to the **Universal Windows Platform** where a message will be displayed on screen when a **Button** is clicked

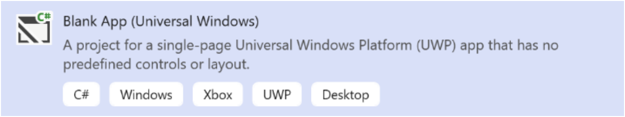
Step 1



Create a new project

Choose a project template with code scaffolding to get started

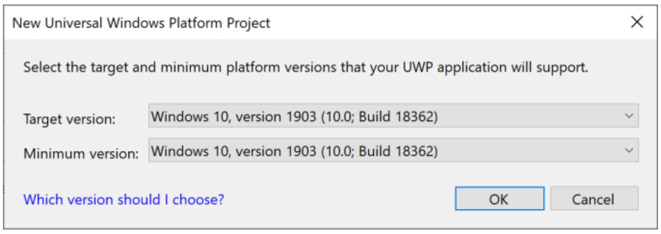
Follow **Setup and Start** on how to Install and/or Get Started with **Visual Studio 2019** if not already or in **Windows 10** choose **Start**, find and select **Visual Studio 2019** then from the **Get started** screen select **Create a new project**



Blank App (Universal Windows)

A project for a single-page Universal Windows Platform (UWP) app that has no predefined controls or layout.

Then choose **Blank App (Universal Windows)** and select **Next** and then in **Configure your new project** enter the **Project name** as **HelloWorld** and select **Create**



New Universal Windows Platform Project

Select the target and minimum platform versions that your UWP application will support.

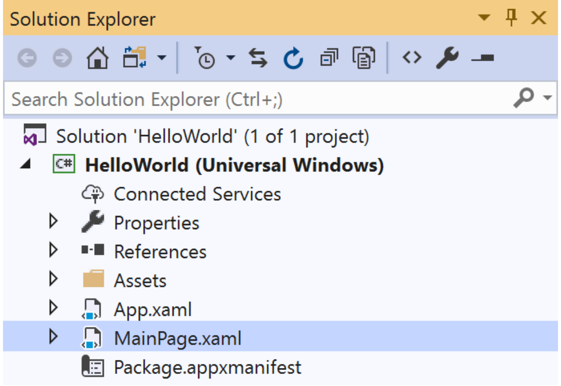
Target version: Windows 10, version 1903 (10.0; Build 18362)

Minimum version: Windows 10, version 1903 (10.0; Build 18362)

Finally, in **New Universal Windows Platform Project** pick the **Target version** and **Minimum version** to be at least **Windows 10, version 1903 (10.0; Build 18362)** and then select **OK**

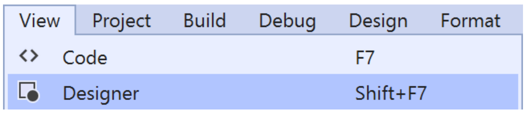
Target Version will control the most recent features of Windows 10 your application can use. To make sure you always have the most recent version, check for any Notifications or Updates in Visual Studio 2019

Step 2



In the **Solution Explorer** of **Visual Studio 2019** select **MainPage.xaml**

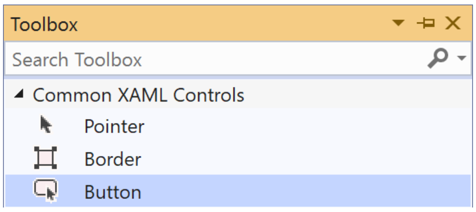
Step 3



Choose **View** then **Designer** from the **Menu** in **Visual Studio 2019**

Universal Windows Platform - Hello World

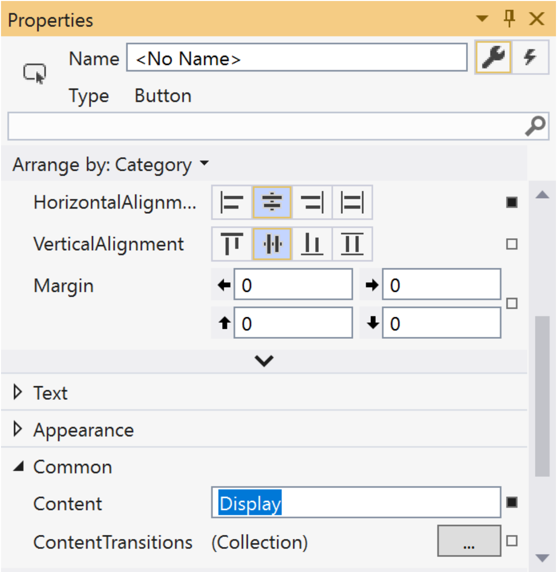
Step 4



In the **Toolbox** of **Visual Studio 2019** from **Common XAML Controls**, double-click **Button** to add it to the **Design** View

MainPage.xaml makes up the look of the application by placing Controls on the Design View

Step 5

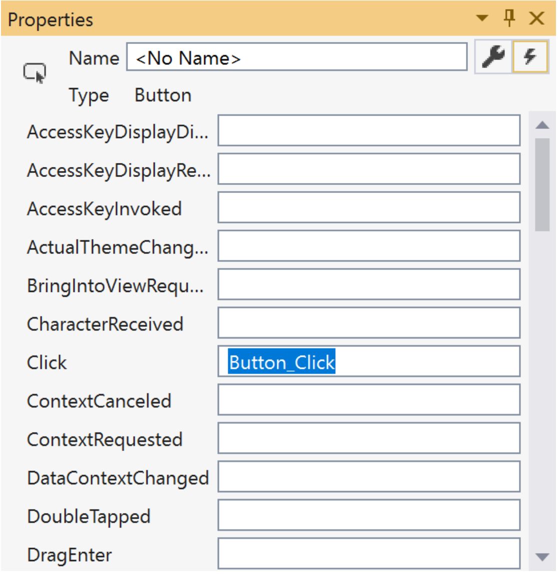


Name <No Name>
Type Button
HorizontalAlignm...
VerticalAlignment
Margin 0 0 0 0
Content Display
ContentTransitions (Collection)

When the **Button** has been added to the **Design** View go to **Properties** set **HorizontalAlignment** to **Center**, **VerticalAlignment** to **Center** and **Content** to **Display**

The Button will appear in the middle of the Design View with the Content of Display once the Properties have been set correctly

Step 6



Name <No Name>
Type Button
AccessKeyDisplayDi...
AccessKeyDisplayRe...
AccessKeyInvoked
ActualThemeChang...
BringIntoViewRequ...
CharacterReceived
Click Button_Click
ContextCanceled
ContextRequested
DataContextChanged
DoubleTapped
DragEnter

While still in the **Properties** select **Events** and then set **Click** to **Button_Click** then either double-click on the text or press Enter once that has been typed in

Universal Windows Platform - Hello World

Step 7

Finally, once done the **Code** View will be displayed and inside the **`Button_Click(...)`** method the following should be entered:

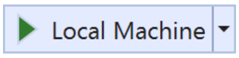
```
_ = new Windows.UI.Popups.MessageDialog("Hello World").ShowAsync();
```

The **`Button_Click(...)`** method should then appear as follows:

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    _ = new Windows.UI.Popups.MessageDialog("Hello World").ShowAsync();
}
```

Clicking on the Button the Event of `Button_Click(...)` will be triggered and this display a MessageDialog with the Text Hello World

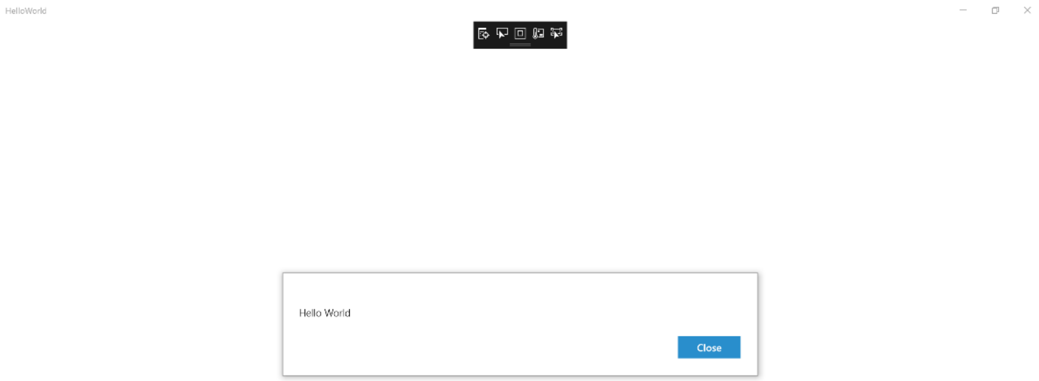
Step 8



That completes the **Universal Windows Platform** Application, in **Visual Studio 2019** select **Local Machine** to run the Application

Step 9

Once the running, you can click **Display** to show the **MessageDialog** and dismiss it with **Close**



Hello World

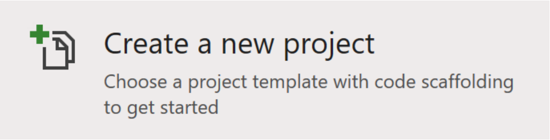
Step 10

To Exit the Application, select the **Close** button in the top right of the Application

Universal Windows Platform – Command Bar

Command Bar is where **AppBarButton** Controls can be added, these allow a standard-looking interface for applications to perform actions or access options

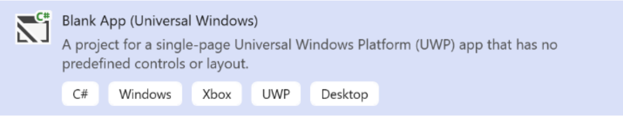
Step 1



Create a new project

Choose a project template with code scaffolding to get started

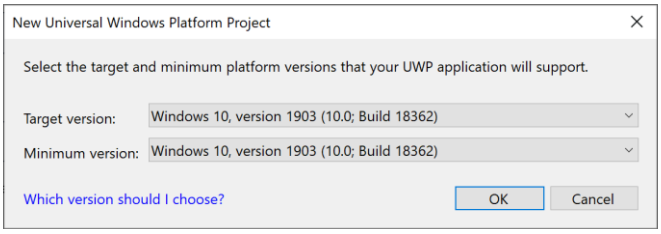
Follow **Setup and Start** on how to Install and/or Get Started with **Visual Studio 2019** if not already or in **Windows 10** choose **Start**, find and select **Visual Studio 2019** then from the **Get started** screen select **Create a new project**



Blank App (Universal Windows)

A project for a single-page Universal Windows Platform (UWP) app that has no predefined controls or layout.

Then choose **Blank App (Universal Windows)** and select **Next** and then in **Configure your new project** enter the **Project name** as **CommandBar** and select **Create**



New Universal Windows Platform Project

Select the target and minimum platform versions that your UWP application will support.

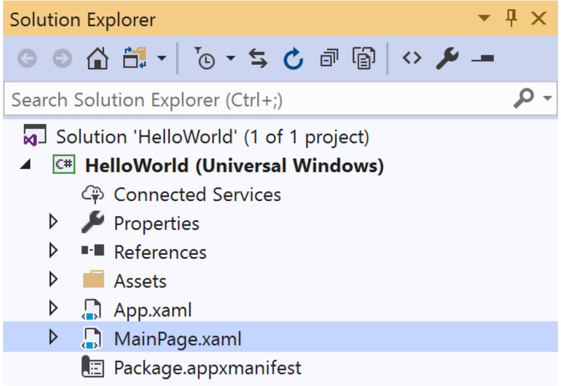
Target version: Windows 10, version 1903 (10.0; Build 18362)

Minimum version: Windows 10, version 1903 (10.0; Build 18362)

Finally, in **New Universal Windows Platform Project** pick the **Target version** and **Minimum version** to be at least **Windows 10, version 1903 (10.0; Build 18362)** and then select **OK**

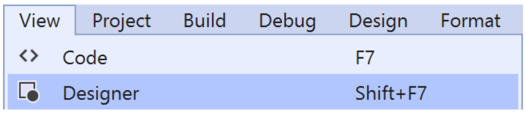
Target Version will control the most recent features of Windows 10 your application can use. To make sure you always have the most recent version, check for any Notifications or Updates in Visual Studio 2019

Step 2



In the **Solution Explorer** of **Visual Studio 2019** select **MainPage.xaml**

Step 3



Choose **View** then **Designer** from the **Menu** in **Visual Studio 2019**

Universal Windows Platform – Command Bar

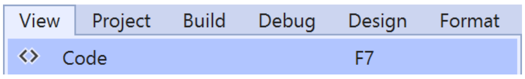
Step 4

In the **Design** View and **XAML** View of **Visual Studio 2019** will be displayed, and in this between the **`Grid`** and **`/Grid`** elements enter the following **XAML**:

```
<CommandBar IsOpen="True" IsSticky="True" VerticalAlignment="Bottom">
    <CommandBar.SecondaryCommands>
        <AppBarButton Name="Hide" Icon="Cancel" Label="Hide Other"
        Visibility="Collapsed" Click="Show_Click"/>
    </CommandBar.SecondaryCommands>
    <AppBarButton Name="Show" Icon="Accept" Label="Show Other"
    Click="Show_Click"/>
</CommandBar>
```

CommandBar is a Control that can contain AppBarButton that will be displayed o show the main toolbar of a Universal Windows Platform Application in Windows 10

Step 5



Choose **View** then **Code** from the **Menu** in **Visual Studio 2019**

Step 6

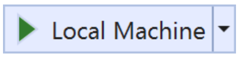
Once in the **Code** View, below the end of `public MainPage() { ... }` the following Code should be entered:

```
private void Show_Click(object sender, RoutedEventArgs e)
{
    if (Hide.Visibility == Visibility.Collapsed)
    {
        Hide.Visibility = Visibility.Visible;
    }
    else
    {
        Hide.Visibility = Visibility.Collapsed;
    }
}
```

`Show_Click` is an Event handler that will be triggered when Hide Other or Show Other is Clicked. This will `if` the `Hide.Visibility` is `Visibility.Collapsed` will set `Hide.Visibility` to `Visibility.Visible` or `else` it will set `Hide.Visibility` it to `Visibility.Collapsed`

Universal Windows Platform – Command Bar

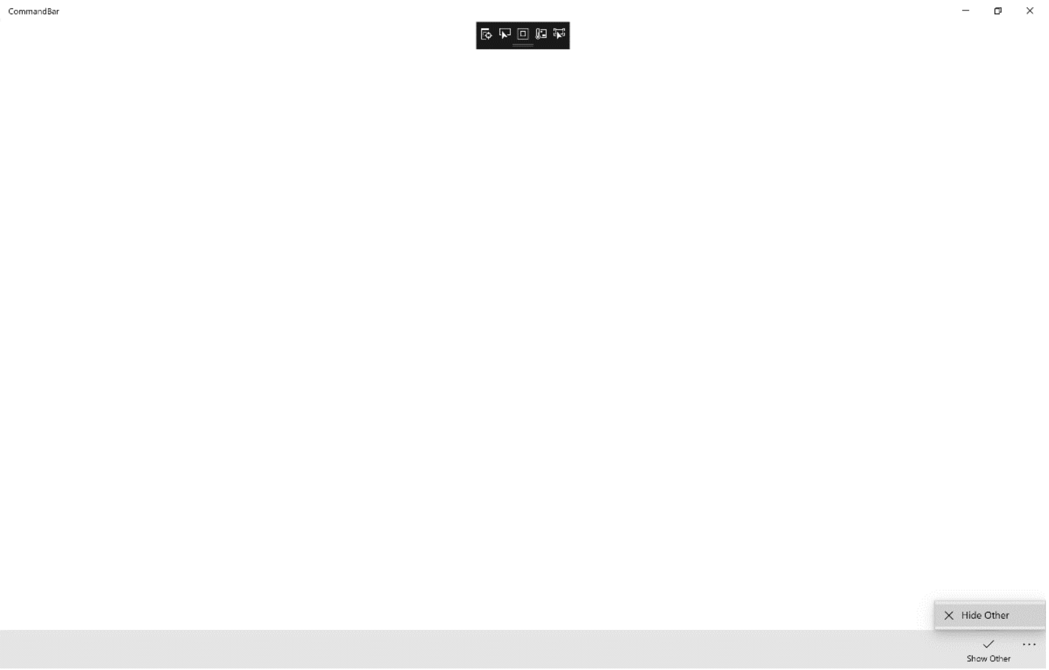
Step 7



That completes the **Universal Windows Platform** Application, in **Visual Studio 2019** select **Local Machine** to run the Application

Step 8

Once the Application is running click **Show Other** to show an option on the bottom when **...** is tapped and tap **Hide Other** to hide this option again



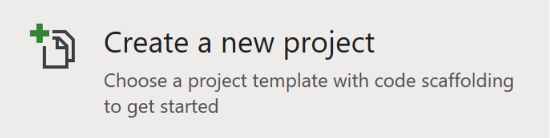
Step 9

To Exit the Application, select the **Close** button in the top right of the Application

Universal Windows Platform – Data Input

Data Input shows how to use **InputScope** for on-screen Keyboards where supported and and loading **ApplicationData**

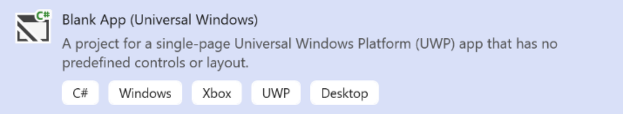
Step 1



Create a new project

Choose a project template with code scaffolding to get started

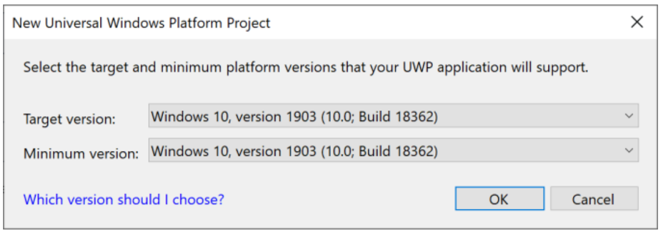
Follow **Setup and Start** on how to Install and/or Get Started with **Visual Studio 2019** if not already or in **Windows 10** choose **Start**, find and select **Visual Studio 2019** then from the **Get started** screen select **Create a new project**



Blank App (Universal Windows)

A project for a single-page Universal Windows Platform (UWP) app that has no predefined controls or layout.

Then choose **Blank App (Universal Windows)** and select **Next** and then in **Configure your new project** enter the **Project name** as **DataInput** and select **Create**



New Universal Windows Platform Project

Select the target and minimum platform versions that your UWP application will support.

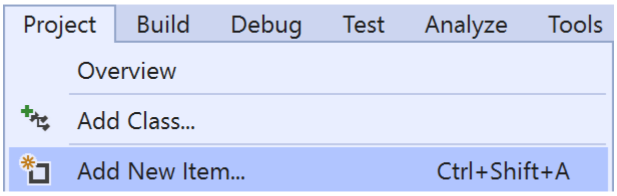
Target version: Windows 10, version 1903 (10.0; Build 18362)

Minimum version: Windows 10, version 1903 (10.0; Build 18362)

Finally, in **New Universal Windows Platform Project** pick the **Target version** and **Minimum version** to be at least **Windows 10, version 1903 (10.0; Build 18362)** and then select **OK**

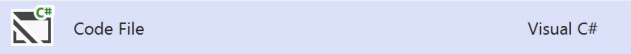
Target Version will control the most recent features of Windows 10 your application can use. To make sure you always have the most recent version, check for any Notifications or Updates in Visual Studio 2019

Step 2



Choose **Project** then **Add New Item…** from the **Menu** in **Visual Studio 2019**

Step 3



Then choose **Code File** from **Add New Item** in **Visual Studio 2019**, enter the **Name** as **Library.cs** and select **Add**

Universal Windows Platform – Data Input

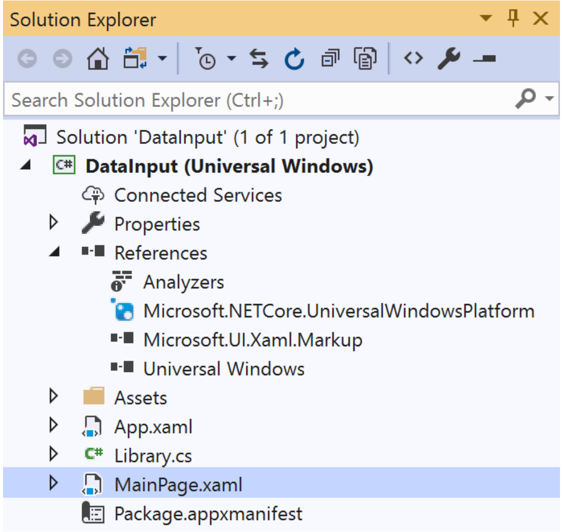
Step 4

In the **Code** View of **Library.cs** will be displayed and in this the following should be entered:

```
using Windows.Storage;

public class Library
{
    public string LoadSetting(string key)
    {
        return (string)(ApplicationData.Current.LocalSettings.Values[key]
        ?? string.Empty);
    }

    public void SaveSetting(string key, string value)
    {
        ApplicationData.Current.LocalSettings.Values[key] = value;
    }
}
```

There is a `using` statement to include functionality from `Windows.Storage`. `LoadSetting(...)` method takes a `string` parameter to `return` the `LocalSettings` with the `key` if present and using the null coalesce or `??` operator will be `string.Empty` if it is not. `SaveSetting(...)` method takes two `string` parameters to set the `LocalSettings` to be returned later with the `key` and `value` passed in

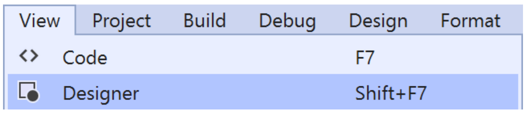
Step 5



In the **Solution Explorer** of **Visual Studio 2019** select **MainPage.xaml**

Universal Windows Platform – Data Input

Step 6



Choose **View** then **Designer** from the **Menu** in **Visual Studio 2019**

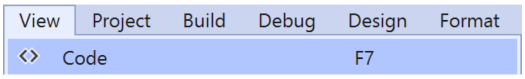
Step 7

In the **Design** View and **XAML** View of **Visual Studio 2019** will be displayed, and in this between the **`Grid`** and **`/Grid`** elements enter the following **XAML**:

```
<StackPanel>
    <TextBox Name="Email" PlaceholderText="Email"
    InputScope="EmailSmtpAddress" Margin="20"/>
    <TextBox Name="Website" PlaceholderText="Website"
    InputScope="Url" Margin="20"/>
    <TextBox Name="Telephone" PlaceholderText="Telephone"
    InputScope="TelephoneNumber" Margin="20"/>
</StackPanel>
<CommandBar VerticalAlignment="Bottom">
    <AppBarButton Icon="Page2" Label="New" Click="New_Click"/>
    <AppBarButton Icon="OpenLocal" Label="Open" Click="Open_Click"/>
    <AppBarButton Icon="Save" Label="Save" Click="Save_Click"/>
</CommandBar>
```

The first block of XAML comprises of TextBox Controls which will show the relevant on-screen Keyboard InputScope if supported. The second block of XAML is the CommandBar containing the operations

Step 8



Choose **View** then **Code** from the **Menu** in **Visual Studio 2019**

Universal Windows Platform – Data Input

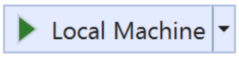
Step 9

Once in the **Code** View, below the end of **`public MainPage() { ... }`** the following Code should be entered:

```
Library library = new Library();

private void New_Click(object sender, RoutedEventArgs e)
{
    Email.Text = string.Empty;
    Website.Text = string.Empty;
    Telephone.Text = string.Empty;
}

private void Open_Click(object sender, RoutedEventArgs e)
{
    Email.Text = library.LoadSetting("Email");
    Website.Text = library.LoadSetting("Website");
    Telephone.Text = library.LoadSetting("Telephone");
}

private void Save_Click(object sender, RoutedEventArgs e)
{
    library.SaveSetting("Email", Email.Text);
    library.SaveSetting("Website", Website.Text);
    library.SaveSetting("Telephone", Telephone.Text);

}
```

Below the MainPage(...) method an instance of the `Library` Class is created. In the `New_Click(...)` Event handler the TextBox Controls have their Text property set to an Empty String. The `Open_Click(...)` Event handler uses the `LoadSetting` method to load a value that has been previously Saved and the `Save_Click(...)` Event handler will use `SaveSetting` to store a value to be loaded later

Step 10

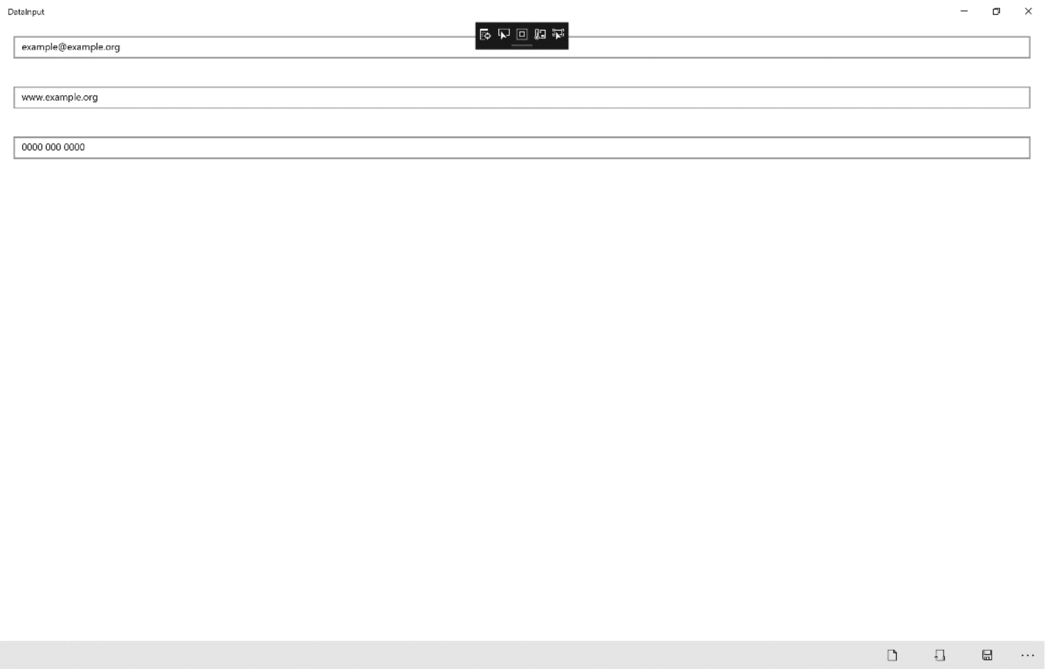


That completes the **Universal Windows Platform** Application, in **Visual Studio 2019** select **Local Machine** to run the Application

Universal Windows Platform – Data Input

Step 11

Once the Application is running you can then input some data such as an **Email Address**, **Website** and **Telephone Number** then store using the **Save** button and recall the data with the **Open** button or reset with the **New** button



example@example.org

www.example.org

0000 000 0000

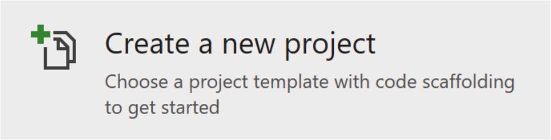
Step 12

To Exit the Application, select the **Close** button in the top right of the Application

Universal Windows Platform – Image Rotate

Image Rotate shows how to use a **Storyboard** to create a simple **DoubleAnimation** to rotate an image in the **X**, **Y** and **X** axis

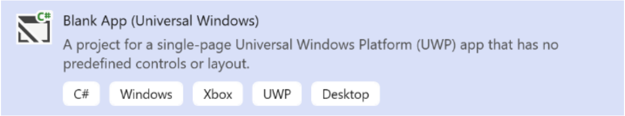
Step 1



Create a new project

Choose a project template with code scaffolding to get started

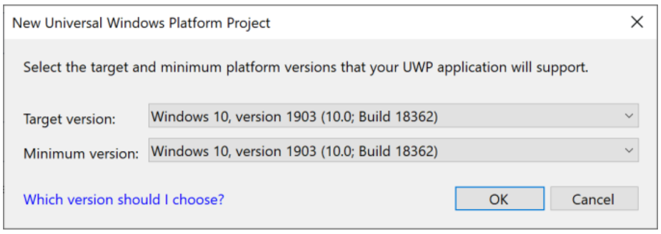
Follow **Setup and Start** on how to Install and/or Get Started with **Visual Studio 2019** if not already or in **Windows 10** choose **Start**, find and select **Visual Studio 2019** then from the **Get started** screen select **Create a new project**



Blank App (Universal Windows)

A project for a single-page Universal Windows Platform (UWP) app that has no predefined controls or layout.

Then choose **Blank App (Universal Windows)** and select **Next** and then in **Configure your new project** enter the **Project name** as **ImageRotate** and select **Create**



New Universal Windows Platform Project

Select the target and minimum platform versions that your UWP application will support.

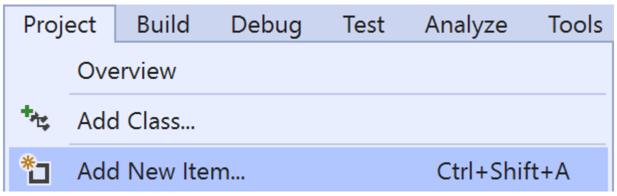
Target version: Windows 10, version 1903 (10.0; Build 18362)

Minimum version: Windows 10, version 1903 (10.0; Build 18362)

Finally, in **New Universal Windows Platform Project** pick the **Target version** and **Minimum version** to be at least **Windows 10, version 1903 (10.0; Build 18362)** and then select **OK**

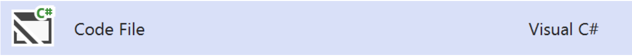
Target Version will control the most recent features of Windows 10 your application can use. To make sure you always have the most recent version, check for any Notifications or Updates in Visual Studio 2019

Step 2



Choose **Project** then **Add New Item…** from the **Menu** in **Visual Studio 2019**

Step 3



Then choose **Code File** from **Add New Item** in **Visual Studio 2019**, enter the **Name** as **Library.cs** and select **Add**

Universal Windows Platform – Image Rotate

Step 4

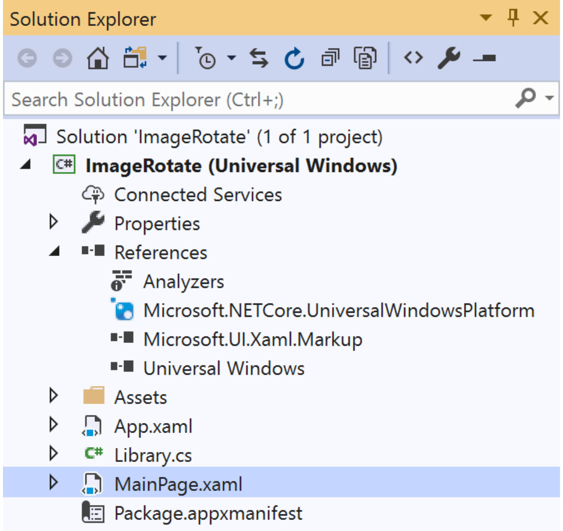
In the **Code** View of **Library.cs** will be displayed and in this the following should be entered:

```
using System;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Media.Animation;

public class Library
{
    private bool _rotating = false;
    private Storyboard _rotation = new Storyboard();

    public void Rotate(string axis, ref Image target)
    {
        if (_rotating)
        {
            _rotation.Stop();
            _rotating = false;
        }
        else
        {
            DoubleAnimation animation = new DoubleAnimation
            {
                From = 0.0,
                To = 360.0,
                BeginTime = TimeSpan.FromSeconds(1),
                RepeatBehavior = RepeatBehavior.Forever
            };
            Storyboard.SetTarget(animation, target);
            Storyboard.SetTargetProperty(animation,
            $"(UIElement.Projection).(PlaneProjection.Rotation{axis})");
            _rotation.Children.Clear();
            _rotation.Children.Add(animation);
            _rotation.Begin();
            _rotating = true;
        }
    }
}
```

There is a `using` statement to include functionality needed for the application. `Storyboard` is used as part of the later `DoubleAnimation` which will animate between 0 and 360 and this will repeat Forever after 1 second. The `PlaneProjection.Rotation` value for each Axis is set on the `UIElement` which in this case is the `Image` named `target` and uses String Interpolation Syntax or `$`

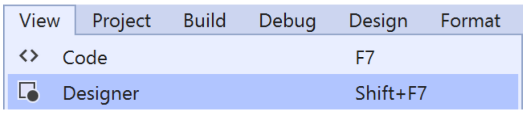
Universal Windows Platform – Image Rotate

Step 5



In the **Solution Explorer** of **Visual Studio 2019** select **MainPage.xaml**

Step 6



Choose **View** then **Designer** from the **Menu** in **Visual Studio 2019**

Step 7

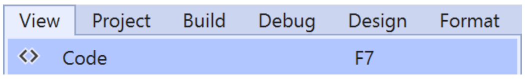
In the **Design** View and **XAML** View of **Visual Studio 2019** will be displayed, and in this between the **`Grid`** and **`/Grid`** elements enter the following **XAML**:

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*"/>
    </Grid.RowDefinitions>
    <TextBox Grid.Row="0" Name="Value" Margin="20"
    InputScope="Url" KeyDown="Go_KeyDown"/>
    <Image Grid.Row="1" Margin="100" Stretch="Uniform" Name="Display">
        <Image.Projection>
            <PlaneProjection/>
        </Image.Projection>
    </Image>
</Grid>
<CommandBar VerticalAlignment="Bottom">
    <AppBarButton Icon="RepeatAll" Label="Pitch" Click="Pitch_Click"/>
    <AppBarButton Icon="Rotate" Label="Roll" Click="Roll_Click"/>
    <AppBarButton Icon="Refresh" Label="Yaw" Click="Yaw_Click"/>
</CommandBar>
```

The first block of XAML the main user interface features a TextBox. The second block of XAML is is the CommandBar which contains Pitch – to rotate the X Axis, Roll – to rotate the Y Axis and Yaw – to rotate the Z Axis

Universal Windows Platform – Image Rotate

Step 8



Choose **View** then **Code** from the **Menu** in **Visual Studio 2019**

Step 9

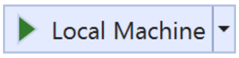
Once in the **Code** View, below the end of **`public MainPage() { ... }`** the following Code should be entered:

```
Library library = new Library();

private void Go_KeyDown(object sender, KeyRoutedEventArgs e)
{
    if (e.Key == Windows.System.VirtualKey.Enter)
    {
        Display.Source = new Windows.UI.Xaml.Media.Imaging
        .BitmapImage(new Uri(Value.Text));
    }
}

private void Pitch_Click(object sender, RoutedEventArgs e)
{
    library.Rotate("X", ref Display);
}

private void Roll_Click(object sender, RoutedEventArgs e)
{
    library.Rotate("Y", ref Display);
}

private void Yaw_Click(object sender, RoutedEventArgs e)
{
    library.Rotate("Z", ref Display);
}
```

Below the MainPage(...) method an instance of the `Library` Class is created. In the `Go_KeyDown(...)` Event handler the Image has the `Source` property set to the contents any URL entered in the TextBox, the `Pitch_Click(...)`, `Roll_Click(...)` and `Yaw_Click(...)` event handler will use the `Rotate` method to set which Axis the Image should be rotated by

Universal Windows Platform – Image Rotate

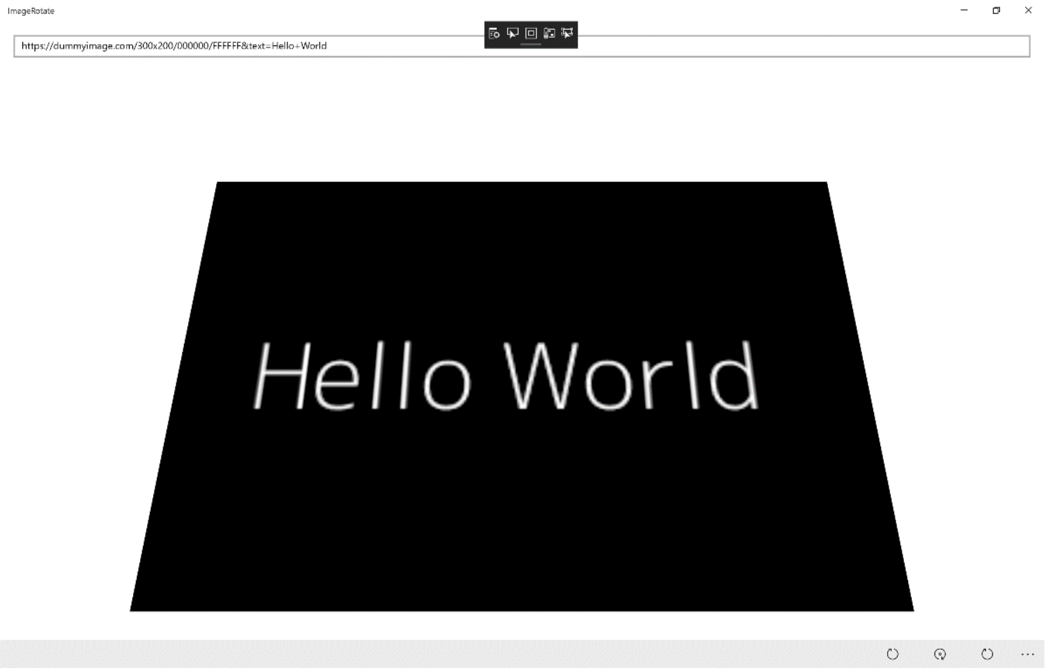
Step 10



That completes the **Universal Windows Platform** Application, in **Visual Studio 2019** select **Local Machine** to run the Application

Step 11

Once the Application is running you can then type in the URL of any image e.g. https://dummyimage.com/300x200/000000/FFFFFF&text=Hello+World then press or tap Enter to load it, then use the **Pitch**, **Roll** or **Yaw** buttons to rotate the **Image**



https://dummyimage.com/300x200/000000/FFFFFF&text=Hello+World

Step 12

To Exit the Application, select the **Close** button in the top right of the Application