# How to Get Screen Resolution in Win10 UWP App

## Introduction

The sample demonstrates how to get screen resolution in Win10 UWP app.

## Building the Sample

This sample should be run in Microsoft Visual Studio 2015 version and Windows 10. Before you start, make sure you have installed Visual Studio 2015 in Windows 10.

Start Microsoft Visual Studio and select **File > Open > Project/Solution.**

* Go to the directory to which the sample was unzipped. Then go to the subdirectory named for the sample and double-click the Visual Studio 2015 Solution (.sln) file.
* Select **Build > Build Solution** to build the sample.

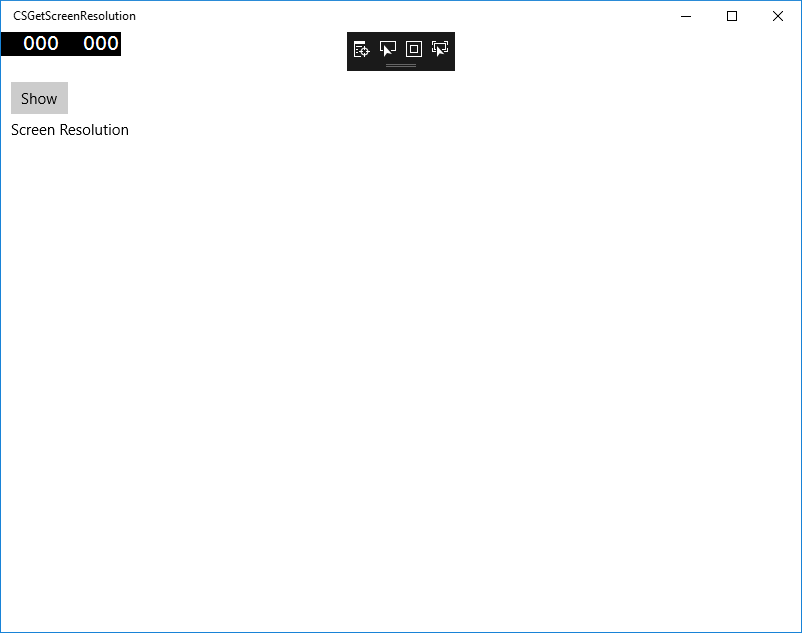
## Running the Sample

**Desktop**

In the target device menu on the Standard toolbar, make sure that **Local Machine** is selected. Do one of the following:

* Click the **Start Debugging** button on the toolbar.
* Click **Start Debugging** in the **Debug** menu.
* Press F5.

The app window will be opened.



**Mobile**

In the target device menu on the Standard toolbar, select one of the Windows 10 mobile emulators and do one of the following:

* Click the **Start Debugging** button on the toolbar.
* Click **Start Debugging** in the **Debug** menu.
* Press F5.

The app will be opened in the emulator.



## Using the Code

MainPage.xaml:

<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

<Button x:Name="btnShow" Content="Show" HorizontalAlignment="Left" Margin="10,50,0,0" VerticalAlignment="Top" Click="btnShow\_Click"/>

<TextBlock x:Name="txtInfo" HorizontalAlignment="Left" Margin="10,87,0,0" TextWrapping="Wrap" Text="Screen Resolution" VerticalAlignment="Top" Height="221" Width="340"/>

</Grid>

MainPage.xaml.cs

/// <summary>

/// btnShow click event

/// Get the screen resolution and show the information in the textblock.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

// Clean the TextBlock

this.txtInfo.Text = string.Empty;

//Add the screen resolution information to the textblock.

if (App.ScreenResolutionSize != Size.Empty)

{

var fullSize = App.ScreenResolutionSize;

this.txtInfo.Text = string.Format("The screen resolution is: {0}x{1}", fullSize.Width, fullSize.Height);

}

var windowSize = ScreenResolutionHelper.GetScreenResolutionInfo();

//Add the application window's resolution information to the textblock.

if (windowSize != null)

{

this.txtInfo.Text += (string.IsNullOrEmpty(this.txtInfo.Text) ? string.Empty : Environment.NewLine)

+ string.Format("The application window's resolution is: {0}x{1}", windowSize.Width, windowSize.Height);

}

}

}

public class ScreenResolutionHelper

{

/// <summary>

/// Get screen resolution.

/// If you want to get the resolution on every page in your solution, you need to call this method from app.xaml.cs and save the data as a global variable.

/// If you have more than one computer monitor, you can only get the main monitor's screen resolution.

/// </summary>

/// <returns></returns>

public static Size GetScreenResolutionInfo()

{

var applicationView = ApplicationView.GetForCurrentView();

var displayInformation = DisplayInformation.GetForCurrentView();

var bounds = applicationView.VisibleBounds;

var scale = displayInformation.RawPixelsPerViewPixel;

var size = new Size(bounds.Width \* scale, bounds.Height \* scale);

return size;

}

}

App.xaml.cs

/// <summary>

/// strore the screen resolution size.

/// </summary>

public static Size ScreenResolutionSize;

protected override void OnLaunched(LaunchActivatedEventArgs e)

{

//……

Window.Current.Activate();

ScreenResolutionSize = ScreenResolutionHelper.GetScreenResolutionInfo();

}

## More Information

* ApplicationView class reference: <https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.viewmanagement.applicationview.aspx>
* DisplayInformation class reference: <https://msdn.microsoft.com/en-us/library/windows/apps/windows.graphics.display.displayinformation.aspx>