# How to zoom in/out the content in WebView in Universal apps

## Introduction

This code sample will show you how to zoom in/out the content in WebView in Universal apps.

To resize the content of the WebView, we can manipulate the content by using JS/CSS that are fully supported by WebView.

## Building the Sample

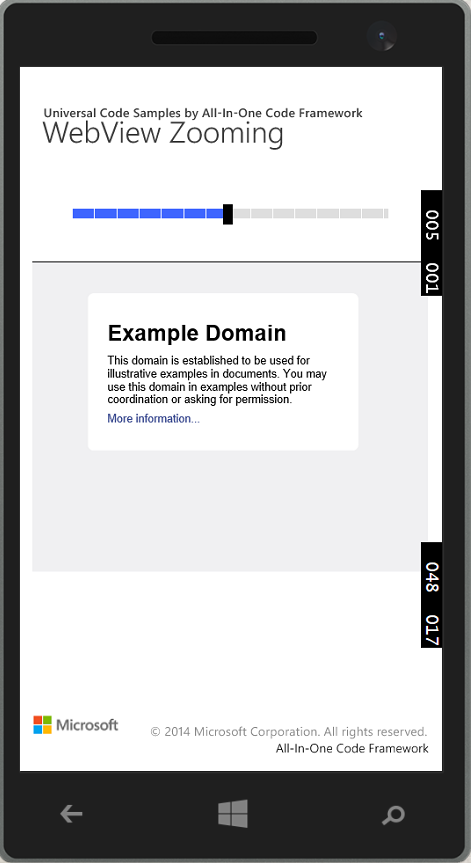
Just build the solution in Visual Studio 2013.

Note: to build and run Universal apps, you have to apply Visual Studio 2013 Update 2.

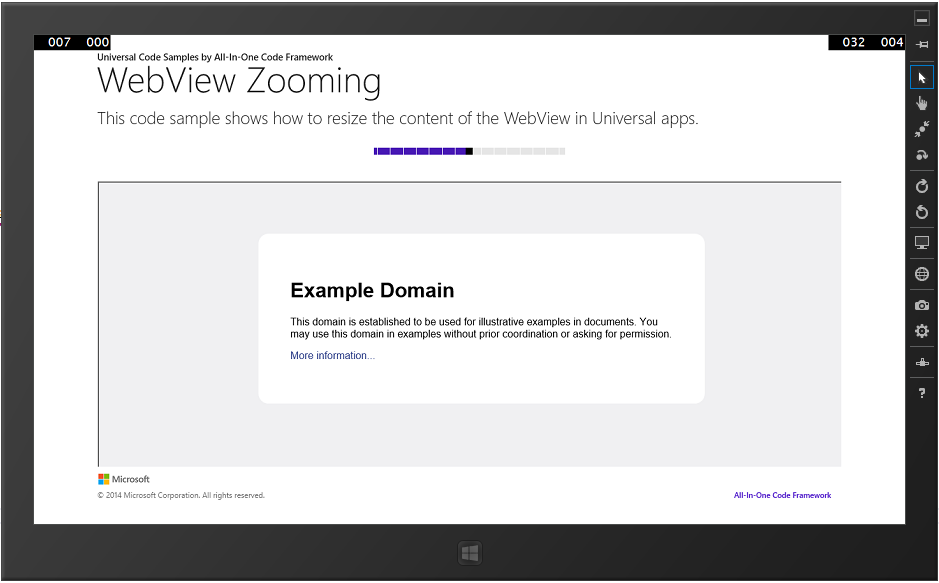
## Running the Sample

Make sure you've built the solution successfully. Then you can run the app.

If you launch the Windows Phone app, it will look like this:



If you launch the Windows Store app, it will look like this:



## Using the Code

Step1. Create a Universal app in Visual Studio.

Step2. Design the UI for both Windows and Windows Phone apps. The following code snippet only shows the page body code. For detailed information, please look into the MainPage.xaml file.

|  |
| --- |
| -Code block start-  --XAML code snippet start--  <!-- page body -->  <Grid Grid.Row="0">  <Grid.RowDefinitions>  <RowDefinition Height="auto"></RowDefinition>  <RowDefinition Height="\*"></RowDefinition>  </Grid.RowDefinitions>  <StackPanel>  <Slider Margin="10" Minimum="1" Maximum="200" Value="100" Width ="300" ValueChanged="Slider\_ValueChanged" SnapsTo="Ticks" TickFrequency="1" />  </StackPanel>  <WebView Grid.Row="1" x:Name="MyWebView" Source="ms-appx-web:///iframe.html" Margin="0,0,0,10" />  </Grid>  --XAML code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

Step3. Create an html file named "iframe.html", in which we embed an **iframe** element which hosts a web site source. The JavaScript function "**ZoomFunction**" is used to zoom in/out the web page.

|  |
| --- |
| -Code block start-  --HTML code snippet start--  <!DOCTYPE html>  <html lang="en" xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta charset="utf-8" />  <title></title>  <script type="text/javascript">  var OriginalWidth = 0;  var OriginalHeight = 0;  function ZoomFunction(Percentage) {  var mybody = document.getElementById("mybody");  var myframe = document.getElementById("myiframe");    if (OriginalWidth == 0 && OriginalHeight == 0) {  OriginalWidth = myframe.style.width.replace("px", "");  OriginalHeight = myframe.style.height.replace("px", "");  }    if (Percentage < 100) {  mybody.style.overflow = "hidden";  myframe.style.overflowY = "auto";  NewWidth = (100 \* OriginalWidth) / Percentage;  NewHeight = (100 \* OriginalHeight) / Percentage;  }  else if (Percentage == 100) {  mybody.style.overflow = "hidden";  myframe.style.overflowY = "auto";  NewWidth = OriginalWidth;  NewHeight = OriginalHeight;  }  else {  mybody.style.overflow = "auto";  myframe.style.overflowY = "hidden";  NewWidth = OriginalWidth \* (Percentage / 100);  NewHeight = OriginalHeight \* (Percentage / 10);  }  myframe.style.zoom = Percentage + "%";  myframe.style.width = NewWidth + "px";  myframe.style.height = NewHeight + "px";  }  </script>  </head>  <body id="mybody" style="margin:0px;padding:0px; overflow:hidden;">  <iframe id="myiframe" src="http://www.example.com/" style="width:1200px; height:800px; overflow-X:auto; overflow-y:auto;"></iframe>  </body>  </html>  --HTML code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

Step4. In the code-behind file of MainPage.xaml, add the following code snippet to make sure the WebView control will call the **ZoomFunction**.

|  |
| --- |
| -Code block start-  --C# code snippet start--  private async void Slider\_ValueChanged(object sender, RangeBaseValueChangedEventArgs e)  {  if (MyWebView != null)  await MyWebView.InvokeScriptAsync("eval", new string[] { "ZoomFunction(" + e.NewValue.ToString() + ");" });  }  --C# code snippet end--  --C++ code snippet start--  void CppUniveralAppWebViewZoom::MainPage::Slider\_ValueChanged(Platform::Object^ sender, Windows::UI::Xaml::Controls::Primitives::RangeBaseValueChangedEventArgs^ e)  {  if (MyWebView != nullptr)  {  Platform::Collections::Vector<String^>^ arguments = ref new Platform::Collections::Vector<String^>(1);  String^ s1 = "ZoomFunction(";  String^ s2 = String::Concat(s1, e->NewValue.ToString()) + ");";  arguments->SetAt(0, s2);  // Invoke the javascript function called 'ZoomFunction' that is loaded into the WebView.  create\_task(MyWebView->InvokeScriptAsync(L"eval", arguments)).then([this](String ^result)  {  })  .then([](task<void> t)  {  try  {  t.get();  }  catch (Platform::Exception^ ex)  {  // An exception can be thrown if a webpage has not been loaded into the WebView or no javascript function named "ZoomFunction" is found in the webpage.  Platform::Details::Console::WriteLine(ex->Message);  }  });  }  }  --C++ code snippet end--  -Code block end- |

## More Information

Build universal Windows apps that target Windows and Windows Phone

<http://msdn.microsoft.com/en-us/library/windows/apps/dn609832.aspx>

Using Visual Studio to build Universal XAML Apps

<http://blogs.msdn.com/b/visualstudio/archive/2014/04/14/using-visual-studio-to-build-universal-xaml-apps.aspx>

WebView class

<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.webview.aspx>

WebView Magic Tricks: Zoom Levels

<http://blogs.msdn.com/b/wsdevsol/archive/2013/05/28/webview-magic-tricks-zoom-levels.aspx>