How to set WebBrowser control viewport dimensions on Windows Phone

This article describes how to set the `WebBrowser` control's viewport to match the device resolution using the GDR3 update.

16 Feb 2014

Introduction

For some applications it is useful to create a full screen `WebBrowser` control and set the viewport's width and height equal to the phone's actual resolution.

In GDR2 this can be done by specifying the `device-width` and `device-height` keywords in the `@-ms-viewport` tag. Unfortunately in GDR3, Microsoft has changed how IE10 interprets meta tags and CSS so that these keywords now return values in inches. The result is that it is no longer possible to specify the screen dimensions using this method, and content is stretched beyond the viewport or displayed as if the screen itself had a very low pixel density.

This article present a workaround for GDR3, getting the dimensions from C# code and passing them to the html file inside the `WebBrowser` control.

Viewport setting methods

This can be a very confusing topic because the method and effect of setting the viewport has changed between WP7, WP8, and even between updates. I have even read reports that it differed between phone models!

Windows Phone 7

Originally, using WP7, one of the **three** ways to control the viewport was to set the meta viewport tag in your html file - see The IE Mobile Viewport on Windows Phone 7 (MSDN Blogs).

```
<meta name="Viewport" content="width=device-width; user-scaleable=no; initial-scale=1.0"
/>
```

The `device-width` keyword (and also `device-height`) is the root cause of this article. For the WP7 device, `device-width` means 320 pixels, *not* the width of your device (480 pixels). Microsoft's reasons for this choice are provided in the previous link.

RESULT: Cannot determine screen resolution inside IE9

Windows Phone 8 through GDR2

Windows Phone 8 uses IE10, instead of IE9 in WP7. Different browsers mean different standards and interpretations of those standards. The `device-width` and `device-height` keywords now refer to the actual resolution of the screen in physical pixels.

In addition to this change, Microsoft added the @-ms-viewport rule for CSS to `viewport` control. This can be used in place of the meta `viewport` tag in the html file (the meta `viewport` tag still works).

```
@-ms-viewport { height:device-height; width:device-width}
```

RESULT: Can determine screen resolution inside IE10

Windows Phone 8 GDR3

In IE10 on GDR3 (as found in the Lumia 1520 and on developer phones), Microsoft has once again changed the rules for the keywords `device-height` and `device-width`. The ViewportSettings keywords now refer to the height and width of your device in **inches**!

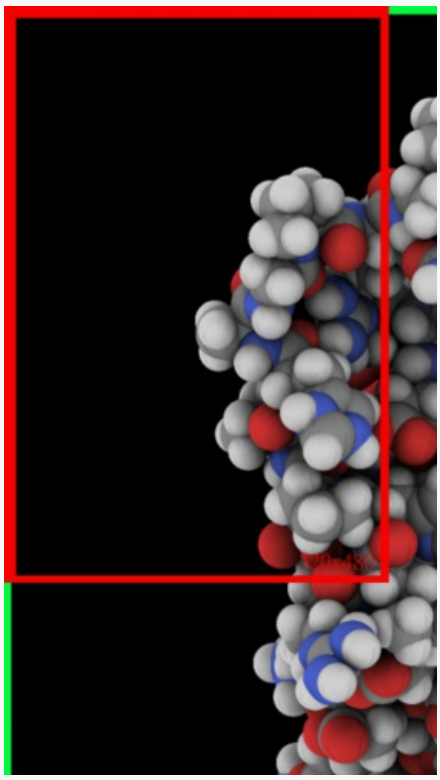
A Nokia Lumia 920 returns a value of 371, which translates to 2.32 inches. IE10 in WP8 does not provide a pixel density value, so you cannot use the width in inches to calculate a device pixel width.

RESULT: Cannot determine screen resolution inside IE10

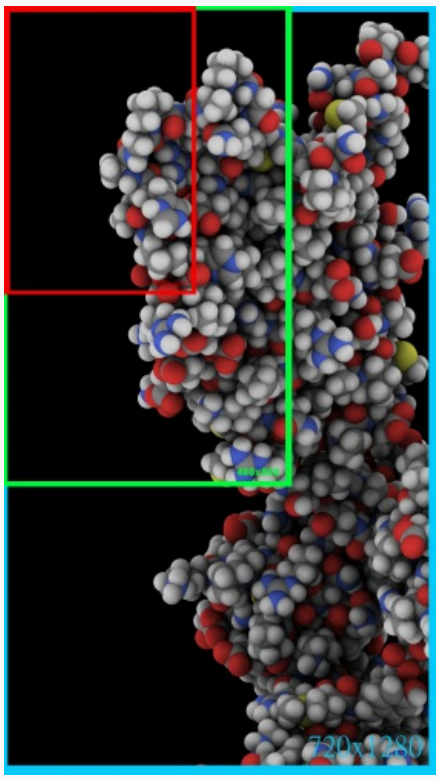
(C) Copyright Nokia 2014. All rights reserved.

Visual aid

The images below show the GDR3 720p emulator with the viewport set using the `@-ms-viewport` rule and using the workaround described in this article. In both cases the picture displayed is 1920x1080 resolution PNG image with typical phone resolutions marked with rectangle outlines.

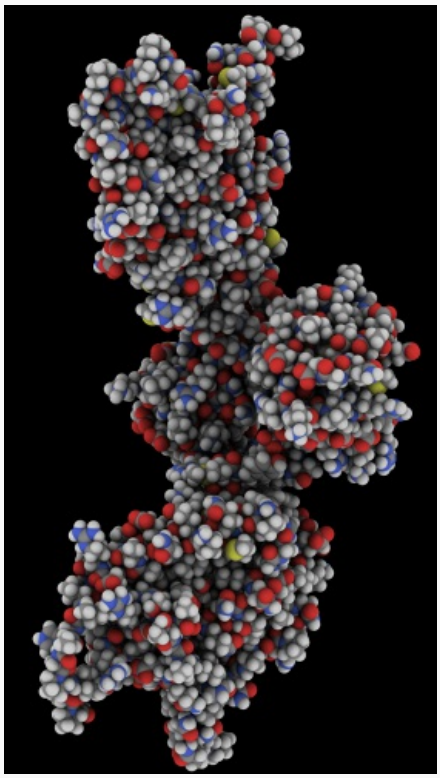


1080x1920 image displayed on a 720p GDR3 device in a WebBrowser control using a viewport set with the keywords: device-width and device-height.



1080x1920 image displayed on a 720p GDR3 device using a viewport set with the workaround.

 (C) Copyright Nokia 2014. All rights reserved.



Original 1080x1920 image without resolution overlays for comparison.

Workaround

The easiest way to determine the screen resolution is to get it from C# code before navigating to your html file inside the WebBrowser control. The previous method of determining actual screen size in C# using just the RenderSize and ScaleFactor are now incorrect since Microsoft introduced 1080p resolution in GDR3.

The following code works for both GDR2 and GDR3. Place this in your `WebBrowser_Loaded` event handler (or `OnNavigatedTo`).

```
string width = "";
string height = "";
try
   {
      var _resolution =
(Size)DeviceExtendedProperties.GetValue("PhysicalScreenResolution");
      width = _resolution.Width.ToString();
      height = _resolution.Height.ToString();
   }
catch
   {
      width = (App.Current.RootVisual.RenderSize.Width *
App.Current.Host.Content.ScaleFactor / 100).ToString();
      height = (App.Current.RootVisual.RenderSize.Height *
App.Current.Host.Content.ScaleFactor / 100).ToString();
   }
this.startUri = new Uri(String.Format("/html/index.html#width={0}&height={1}", width,
height), UriKind.Relative);
```

Notice the URL string contains a '#' character instead of a '?' for the parameter list delineation. Using '?' does not work when navigating to a webpage from C#. We will be using JavaScript, so you **must enable scripting** for the `WebBrowser` control. Somewhere, set `wb.IsScriptEnabled = true`. Finally, in the `Loaded` event handler, navigate to the `startUri` URI that you generated.

 (C) Copyright Nokia 2014. All rights reserved.

Next, edit your html file and place the following JavaScript in the head. Also, ensure that you have `<!DOCTYPE html>` at the top.

```
<script>
        var pwidth, pheight;
        var args = window.location.hash.split('&');
        for (var i = 0; i < args.length; i++) {
            var pair = args[i].split('=');
            if (decodeURIComponent(pair[0]) == 'height') {
                pheight = Math.round(+decodeURIComponent(pair[1]));
            }
            else if (decodeURIComponent(pair[0]) == '#width') {
                pwidth = Math.round(+decodeURIComponent(pair[1]));
            }
        }
        document.write('<meta name="viewport" content="width=' + pwidth + ', height=' +
pheight + ', user-scalable=true">');
</script>
```

The `document.write` command creates a dynamic meta viewport tag with fixed pixel values determined earlier in our C# code. You can set user-scalable to `false` if you do not want to be able to zoom the viewport. The `Math.round` function is present because the 720p emulator did not report an integer value for its device height, but instead reported a fraction.

JavaScript's `screen.width` and `screen.height` properties will now return the actual device resolution of your phone.

Sample Solution

For those of you who want to see the difference between the old method and workaround for setting the viewport's dimensions to the device resolution, please download the sample solution. Try them on both types of emulators: GDR2 and GDR3.

File:WebBrowserViewportSettingsRevised1.zip

 (C) Copyright Nokia 2014. All rights reserved.