

# Displaying InduSoft Web Studio Applications on Two Monitors

Application Note

©2014 InduSoft, Inc.



AN INVENSYS COMPANY.

## Displaying InduSoft Web Studio Applications on Two Monitors<sup>i</sup>

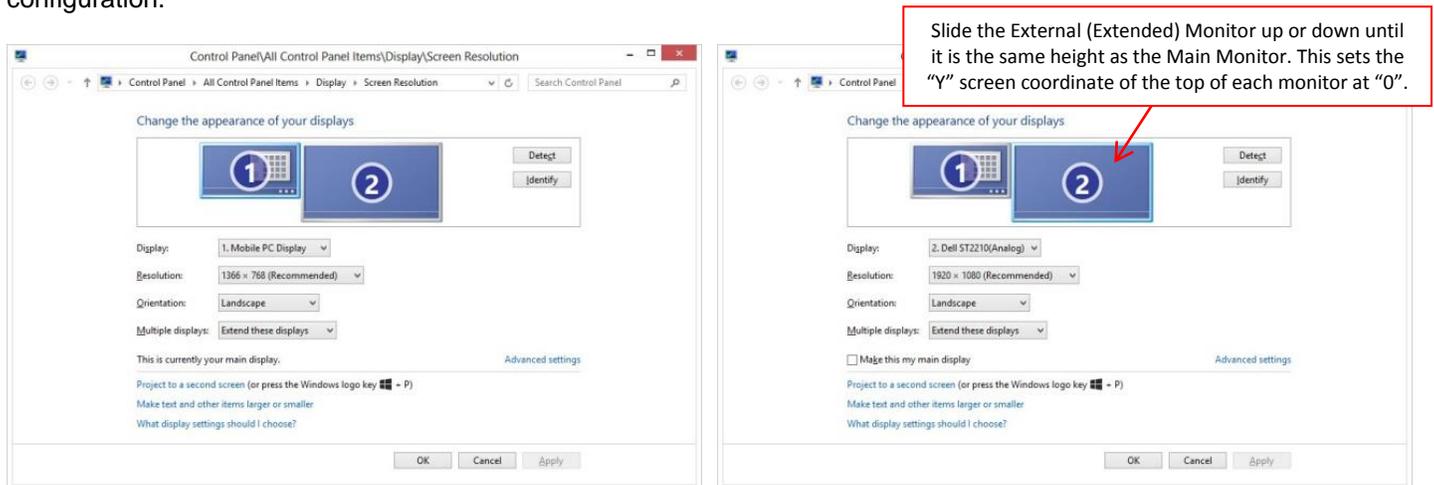
### Implementation Specifications or Requirements

Category	Item
Software	IWS Version: 7.1 and later
	Service Pack: SP2 and later
	Windows Version: Win 7, 8, 2008/12 Server, W7SE, and later
	Web Thin Client: Yes
Equipment	Panel Manufacturer: N/A
	Panel Model: N/A
	Other Hardware: External Monitor
	Comm. Driver: All
	Controller (e.g.: PLC) All
Software Demo Application	Discussed
Date Released	March 7, 2014
Author	Richard Clark

## Initial Configuration

The simplest and most common multi-monitor configuration is a notebook or desktop computer with an attached monitor as part of the extended desktop. The first step in configuring this environment is to establish where the external monitor is in relation to the main display. In Windows Control Panel, this means that each monitor's top edge must be at the same height in relation to each other (**Figure 1**).

Assume for the moment that this configuration has a notebook computer with a Native Resolution of 1366 x 768. This display has an aspect ratio of 16:9, or 1.78. Also, assume that you have a monitor connected to this notebook having a 1920 x 1080 pixel Native Resolution, which also has an aspect ratio of 16:9, or 1.78. Although there are more horizontal and vertical pixels in the attached monitor than on the notebook screen, both of the screens are in a 16:9 aspect ratio configuration.



**Figure 1: Multiple monitor configurations in Windows 8 OS on the same machine.**

# Displaying InduSoft Web Studio Applications on Two Monitors

Application Note

©2014 InduSoft, Inc.



AN INVENSY COMPANY.

If the application has the “*Auto Screen Scaling*” and “*Start Maximized*” attributes set in the Viewer Configurator (Figure 2), it will now start (assuming that the Viewer Task is set to “*Automatic*”) in either monitor, and fill the entire monitor (assuming that the monitor is set as the “*main*” display in Windows).

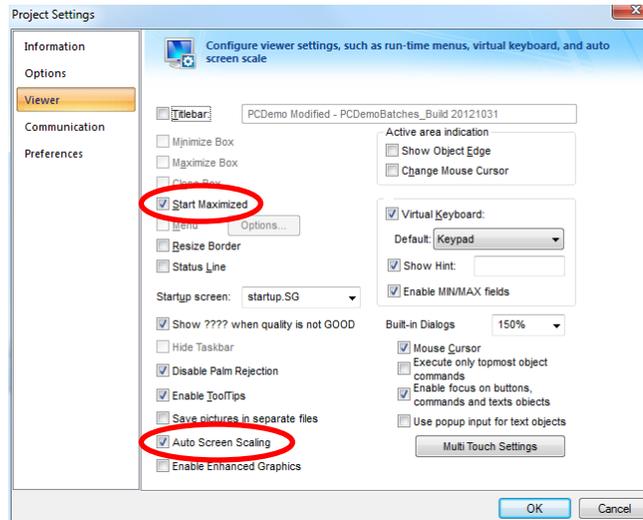


Figure 2: Viewer configurator showing the default settings “Start Maximized” and “Auto Screen Scaling” selected.

Since the aspect ratios of both monitors are the same, and by using these settings, there will not be any horizontal or vertical gray space on either monitor (Figure 3).<sup>ii</sup>



Figure 3: The InduSoft “PC Demo” application is natively built in 1280x800 resolution (5:3 or 1.6 aspect ratio). It is shown displayed on a 1366x768 monitor (16:9 or 1.67 aspect ratio) on the left, and a 1024x768 monitor (4:3 or 1.3 aspect ratio) on the right. The application is Auto-scaled to fit maximized in the monitor at either the Y-Max (left picture) leaving extra X space on the monitor, or X-Max (right picture) leaving extra Y space on the monitor.

## Building a Simple Application and a Discussion of Viewer Module Settings

The simplest example application has only one screen and one viewer.exe (e.g., Graphics Module) that can be displayed on either the “*main*” or the “*external*” monitor.

# Displaying InduSoft Web Studio Applications on Two Monitors

A new project is opened and the size defined as 1366 x 768, which is the size of the (smaller) laptop display. This will make the Graphics Module (or the viewer.exe) the same pixel size as the computer monitor. We will create one screen called Main, the same size as the project, put a button with a **ShutDown()** function in it, and draw some objects in order to give the screen some perspective. This screen is set as the **Startup Screen**, the Viewer Task Startup is set to “**Automatic**”, and the “**Auto Scaling**” and “**Start Maximized**” options are checked in the Viewer Configuration.

Starting the **Runtime** will open the Graphics Module and the screen to the full resolution of whichever monitor is selected as the **Main Monitor** in Windows. Even though the project size is defined as 1366 x 768, the Graphics Module will open and scale up to 1920 x 1080, resizing everything including the fonts, when the larger external monitor is set to be the **Main Monitor**. If the project is shown on a smaller monitor such as a 1280 x 1024 then the Graphics Module will open at the smaller 1280 x 1024 Resolution, and the screen will scale to the width of the monitor, with a grey bar at the bottom of the monitor.

By deselecting the “**Start Maximized**” checkbox, the Graphics Module will open to a somewhat smaller size than the monitor. Depending on if “**Auto Screen Scaling**” is selected or not, the screen will be resized to fit the graphics module X-maximum or Y maximum (with a grey bar to make up the difference between the screen and the Graphics Module), or open at the defined size of the screen respectively. If **Resize Border** is selected, the graphics module can be dynamically resized during runtime.

Notice that the **defined font point size** chosen for any particular object on any screen is scaled and displayed *according to the size of the Graphics Module (viewer.exe)* that the screen is opened on (along with all other objects on the screen), when “**Auto Scaling**” is selected. The ramification of this statement is that if screens are resized by using the **Open()** function, the fonts on that screen **will not be** resized accordingly. They will remain scaled to the size of the Graphics Module (**Figure 4**).

The final point about resizing screens during runtime is that when using the **Open()** function to open or resize screens, the objects on the screen will not resize (even with the “resize” option being selected in the **Open()** function) if that screen is currently open, *though the screen itself will be resized to the new parameters*. The screen must first be closed just prior to the **Open()** call with the new parameters. The objects on the screen will be scaled to the new size.



**Figure 4: Three images of a resized screen. The Viewer Module is not resized (grey area), but the screen is resized using the **Open()** function. (The screen needs to be closed before the new **Open()** function resizing the same screen can be used). Text size font size does not change.**

Using these techniques, it is now possible to create a new project defining the project size as the width of the extended desktop (the width of both monitors added together) and the height as the vertical size of the bigger monitor, or in the case of monitors using vertical orientation, the height of both monitors added together and the width of the larger one.

Screens can be designed and opened which will fit within each of the monitors or a portion of each monitor by using **Open()** to define the location of the screen within the opened viewer.

# Displaying InduSoft Web Studio Applications on Two Monitors

Application Note

©2014 InduSoft, Inc.



AN INVENSYS COMPANY.

## Using Two Graphics Modules (viewer.exe)

The next project expansion step is to open a Graphics Module (**viewer.exe**) on each monitor. Since the monitors are different sizes, it is necessary to specify where and what size the viewers will be. Assuming that the notebook monitor is the main display, this is how it's done (Figure 5):

1. Set the Viewer Task to start manually.
2. Create a Global Procedure that starts the viewers:

```
Sub StartViewer()  
  If $ViewerNumber < 2  
    $WinExec( $GetProductPath() & "Bin\viewer.exe") 'Starts viewer.exe threads  
  End If  
End Sub
```

3. Create a Script Task to control the call that starts the viewer(s) by setting a Boolean Tag, **\$EnableScreenStart** in the "Execute" field:

```
Call StartViewer()  
$EnableScreenStart = 0 'This halts the script execution by resetting the Execution Tag
```

4. In the Graphics Script, **Graphics\_OnStart()** section, create some code to position each viewer and open a Screen to display on each viewer:

```
Sub Graphics_OnStart()  
  If $ViewerNumber < 1 Then  
    $SetViewerPos( 1366, 0, 1920, 1080) 'Opens the first viewer on the external monitor  
    $Open("RedScreen")  
    $ViewerNumber = $ViewerNumber + 1  
  Else  
    If $ViewerNumber = 1 Then  
      $SetViewerPos(0, 0, 1366, 768) 'Opens the second viewer on the notebook monitor  
      $Open("BlueScreen")  
      $ViewerNumber = $ViewerNumber + 1  
    End If  
  End If  
  $EnableScreenStart = 1 'Enables the Script Task in order to open the next viewer  
End Sub  
  
Sub Graphics_OnEnd()  
  $ViewerNumber = 0 'Resets the Viewer Counter  
End Sub
```

5. Add a **Call StartViewer()** to the **Application Startup Script Task** to open the first Graphics Module.
6. Finally, add a way to close the application—Place "**Call CloseViewer()**" in every Exit Button:

```
Sub CloseViewer()  
  $WinExec( "Taskkill /IM viewer.exe") 'Kills all viewer.exe threads  
  $ShutDown() 'Shuts the application down normally  
End Sub
```

# Displaying InduSoft Web Studio Applications on Two Monitors

Application Note

©2014 InduSoft, Inc.



AN INVENSYS COMPANY.

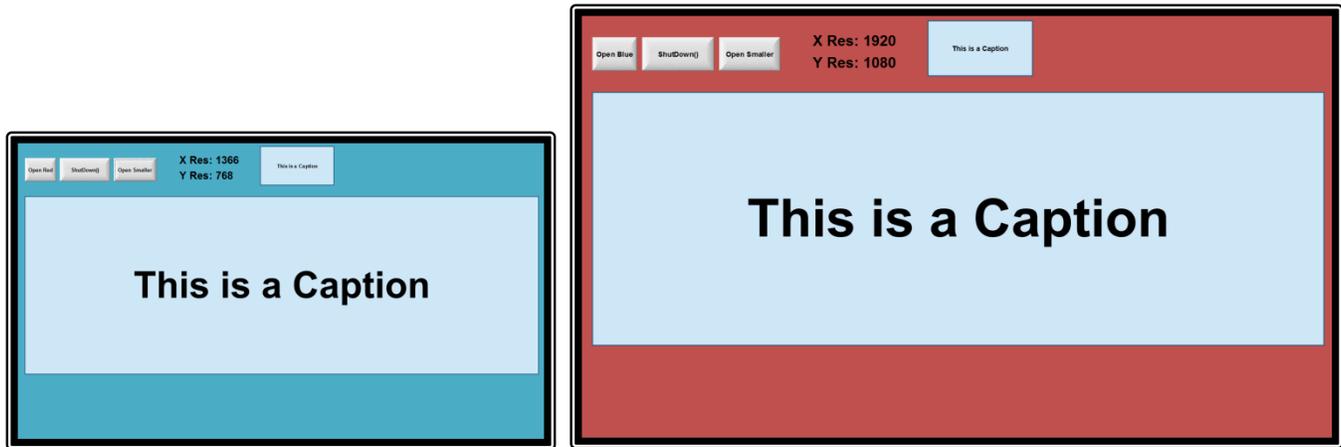


Figure 5: Identically sized project screens (1366 x 768) using identically sized objects and text point sizes, opened full-screen on two monitors that are different sizes within the same project and Runtime.

## End Notes

<sup>i</sup> This Application Note is an excerpt from the Tech Note, “*Using Multiple Monitors Configurations with InduSoft Web Studio*”. More information about complex multi-monitor implementations can be found in the document which can be downloaded from the InduSoft Website.

<sup>ii</sup> More information on using aspect ratios in InduSoft Web Studio projects can be found in this blog posting: <http://www.indusoft.com/blog/2013/10/01/aspect-ratios-and-your-indusoft-web-studio-project/>