

<b>Name of Experiment:</b> Device Info	<b>Exp No:</b> MC2
<b>Background:</b> Student should have a basic knowledge of C#.	
<b>Summary:</b> User sometimes required to know, certain system defined properties that are abstracted from the end user. These properties must be known to user during updating of OS or updating the firmware.	
<b>Learning Objective:</b> Here, Student should learn the procedure to get all necessary details about the device. After, doing this experiment user himself know the running firmware version, hardware version, Device name, Total Memory etc.	
<b>Target Platforms:</b> This experiment is tested on Windows Phone Emulator and Nokia Lumia (800).	
<b>Procedure:</b>  Step1. Repeat the steps [1-4] as in experiment no MC1(HelloWorld).[Refer Exp No MC1]  Step2. Add Reference for Microsoft.Phone.Info.  Step3. Override the methods given in class Microsoft.Phone.Info.DeviceStatus. e.g IskeyboardDeployed, IskeyboardPresent, PowerSource, ApplicationCurrentMemoryUsage, ApplicationPeakMemoryUsage, DeviceTotalMemory, DeviceName, DeviceFirmwareVersion, DeviceHardwareVersion, and DeviceManufacturer.  Step4. Override the method in the class Microsoft.Phone.Info.MediaCapabilites i.e. IsMultiResolutionVideoSupported.  Step5. Now bind each method with the Text block in order to get the value displayed on the screen.  Step6. Save all the changes using Ctrl+ S.  Step7. Press Ctrl+F5 to run this experiment on the emulator.  Step8. By this way, DeviceInfo experiment is deployed on the windows Emulator.  Step9. To deploy this experiment on Windows Phone[Refer Deployment of the WP7 application on the Target Device].	

Source Code	Comments
<p><b>Main.xaml</b></p> <pre> &lt;StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28"&gt; &lt;TextBlock x:Name="ExperimentTitle" Text="Exp.No.MC2" TextAlignment="Right" Style="{StaticResource PhoneTextNormalStyle}"/&gt; &lt;TextBlock x:Name="ApplicationTitle" Text="Mobile Computing" Style="{StaticResource PhoneTextNormalStyle}"/&gt; &lt;TextBlock x:Name="PageTitle" Text="Device info" Margin="9,-7,0,0" Style="{StaticResource PhoneTextTitle1Style}"/&gt; &lt;/StackPanel&gt;  &lt;!--ContentPanel - place additional content here--&gt; &lt;Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0"&gt; &lt;StackPanel Margin="12,6,6,70" Orientation="Vertical" &gt; &lt;TextBlock x:Name="tbkeyDeployed" Text="" Height="43" /&gt; &lt;TextBlock x:Name="tbkeyPresent" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbPowerS" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbApplicationCurrentMem" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbApplicationPeakMem" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbDeviceMem" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbDeviceName" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbDeviceManu" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbFWareVer" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbHwVer" Text="" Height="43"/&gt; &lt;TextBlock x:Name="tbMultiVideo" Text=" " Height="43"/&gt;  &lt;/StackPanel&gt; &lt;/Grid&gt; &lt;/Grid&gt; </pre> <p><b>MainPage.xaml.cs</b></p> <pre> using System; using System.Windows; using Microsoft.Phone.Controls; using Microsoft.Phone.Info;  namespace DeviceInfo {     public partial class MainPage : PhoneApplicationPage     {         string s1, s2, s3, s4, s5,s6;         bool b1, b2,b3;         long l1, l2, l3;         // Constructor         public MainPage()         {             InitializeComponent();             Loaded+=new RoutedEventHandler(MainPage_Loaded);         }         //Initialising the UI         public void MainPage_Loaded(object sender,RoutedEventArgs e) {             b1=IsKeyboardDeployed;             tbkeyDeployed.Text=String.Format("IskeyboardDeployed: {0}\r\n",b1); </pre>	<p>← Exp No.MC2</p> <p>← Mobile Computing ← Device Info</p> <p>← Add Reference for Microsoft.Phone.Info</p>

```

        b2=IsKeyboardPresent;
        tbkeyPresent.Text=String.Format("IskeyboardPresent:{0}\r\n",b2);
        s1=PowerSource;
        tbPowerS.Text=String.Format("PowerSource: {0}\r\n",s1);
        l1=ApplicationCurrentMemoryUsage;
        tbApplicationCurrentMem.Text=
String.Format("ApplicationCurrentMemoryUsage:{0}\r\n",l1);
        l2=ApplicationPeakMemoryUsage;
        tbApplicationPeakMem.Text=
String.Format("ApplicationPeakMemoryUsage:{0}\r\n",l2);
        l3=DeviceTotalMemory;
        tbDeviceMem.Text=
String.Format("DeviceTotalMemory:{0}\r\n",l3);
        s2=DeviceName;

        tbDeviceName.Text=String.Format("DeviceName:{0}\r\n",s2);
        s3=DeviceManufacturer;

        tbDeviceManu.Text=String.Format("DeviceManufacturer:{0}\r\n",s3);
        s4=DeviceFirmwareVersion;

        tbFWareVer.Text=String.Format("DeviceFirmwareVersion:{0}\r\n",s4);
        s5=DeviceHardwareVersion;

        tbHwVer.Text=String.Format("DeviceHardwareVersion:{0}\r\n",s5);
        b3 = IsMultiResolutionVideoSupported;
        tbMultiVideo.Text =
String.Format("MultiResolutionVideoSupported:{0}\r\n",b3);
    }
    // Overriding the Methods, defined in DeviceStatus and
    //MediaCapabilities class
    public bool IsKeyboardDeployed { get { return
DeviceStatus.IsKeyboardDeployed; } }
    public bool IsKeyboardPresent { get { return
DeviceStatus.IsKeyboardPresent; } }
    public string PowerSource { get { return
DeviceStatus.PowerSource.ToString(); } }
    public long ApplicationCurrentMemoryUsage { get { return
DeviceStatus.ApplicationCurrentMemoryUsage; } }
    public long ApplicationPeakMemoryUsage { get { return
DeviceStatus.ApplicationPeakMemoryUsage; } }
    public long DeviceTotalMemory { get { return
DeviceStatus.DeviceTotalMemory; } }
    public string DeviceName { get { return
DeviceStatus.DeviceName; } }
    public string DeviceFirmwareVersion { get { return
DeviceStatus.DeviceFirmwareVersion; } }
    public string DeviceHardwareVersion { get { return
DeviceStatus.DeviceHardwareVersion; } }
    public string DeviceManufacturer { get { return
DeviceStatus.DeviceManufacturer; } }
    public bool IsMultiResolutionVideoSupported { get { return
MediaCapabilities.IsMultiResolutionVideoSupported; } }
}
}

```

## Screenshots



Fig. No.1 Device info

### **Observations:**

It is observed that after running this DeviceInfo experiment, end user now get to know some information related to device which are abstracted from him.