

# Windows Phone 7 Development

31 Days of Windows Phone 7 Development



[WWW.F5DEBUG.NET](http://WWW.F5DEBUG.NET)

February 1, 2012

Karthikeyan Anbarasan, Microsoft MVP

# Learn Windows Phone 7 Development in 31 Days – Day 22

## – Working with Device Information in WP7

### Introduction:

In this article we are going to see how to get the Device information for managing the device. We will see how to make use of the Device detection API's available with the Windows Phone 7 SDK to detect the Device information, Memory detection, Keyboard installed or not etc with the API.

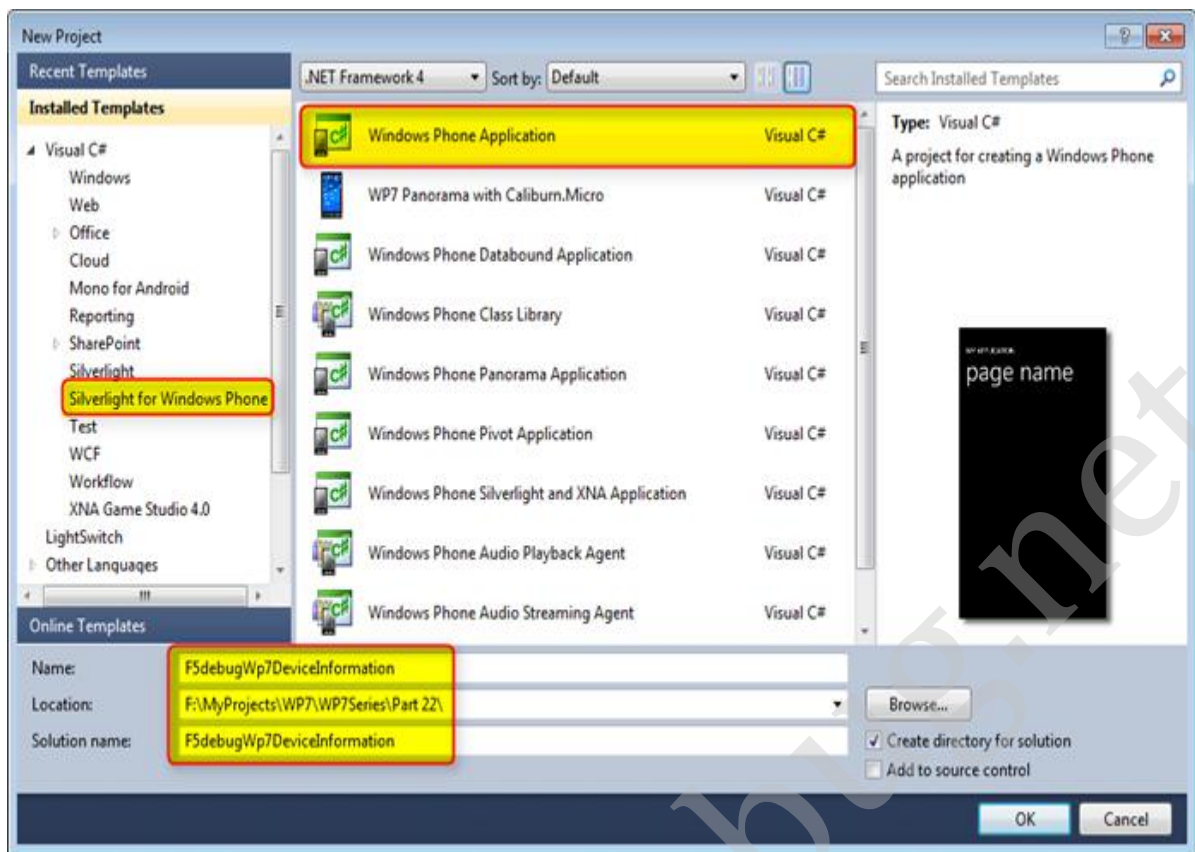
To detect the device information, we are going to use the **DeviceExtendedProperties** class. We have different options to get the information of the device, network etc.. with the API. We will see only retrieving the device information here with this article. We are going to retrieve the below list of device information

S No	Information	Description
1	DeviceUniqueId	Gets device Unique ID.
2	DeviceManufacturer	Gets Device Manufacture details.
3	ApplicationCurrentMemoryUsage	Gets Application Memory Usage.
4	ApplicationPeakMemoryUsage	Gets Application Memory Leak Usage.
5	DeviceName	Gets the Device Name.
6	DeviceFirmwareVersion	Gets the Firmware Version.
7	DeviceHardwareVersion	Gets the Hardware Version.
8	DeviceTotalMemory	Gets the physical Memory RAM Usage.
9	Power Source	Indicates if the device works with power or battery.
10	Iskeyboardpresent	Indicates if any physical keyboard is installed with device.
11	IsKeyboardDeployed	Indicates if the user deploy the physical keyboard
12	ApplicationMemoryUsageLimit	Gets the maximum amount of memory.

Let us jump start to see the step by step process on how to implement the device information class to get the information and show it to the end users.

### Steps:

Open Visual Studio 2010 and create a new Silverlight for Windows Phone 7 Application with a valid project name as shown in the screen below.



Now let us start with our design to see the device informations in one page. Once we are done with the design we can see the screen looks like below screen. Just copy the XAML code to get the same look and feel. Else we can directly drag and drop the controls from the tool box and design it as per the requirement.

### XAML Code:

```
<phone:PhoneApplicationPage
x:Class="F5debugWp7DeviceInformation.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
FontFamily="{StaticResource PhoneFontFamilyNormal}"
FontSize="{StaticResource PhoneFontSizeNormal}"
Foreground="{StaticResource PhoneForegroundBrush}"
SupportedOrientations="Portrait" Orientation="Portrait"
shell:SystemTray.IsVisible="True">
```

```
<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="Transparent">
<Grid.RowDefinitions>
<RowDefinition Height="Auto"/>
<RowDefinition Height="*/>
</Grid.RowDefinitions>
```

```
<!--TitlePanel contains the name of the application and page title-->
```

```

<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
<TextBlock x:Name="ApplicationTitle" Text="F5DEBUG WP7 TUTORIALS"
Style="{StaticResource PhoneTextNormalStyle}"/>
<TextBlock x:Name="PageTitle" Text="Device Info" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
</StackPanel>

<!--ContentPanel - place additional content here-->
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,23,0,0"
Name="ttDeviceInfo" Text="F5debug - Device Information List"
VerticalAlignment="Top" Width="438" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,76,0,0"
Name="ttdeviceuniqueid" Text="Device Unique ID:" VerticalAlignment="Top"
Width="176" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,76,0,0"
Name="txtDeviceUniqueID" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,112,0,0"
Name="textBlock4" Text="Device Manufacturer:" VerticalAlignment="Top"
Width="200" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,112,0,0"
Name="txtDeviceManufacturer" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,148,0,0"
Name="textBlock6" Text="Device Name:" VerticalAlignment="Top" Width="200"
/>
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,148,0,0"
Name="txtDeviceName" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,184,0,0"
Name="textBlock8" Text="Firmware Version:" VerticalAlignment="Top"
Width="200" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,184,0,0"
Name="txtFirmwareVersion" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,220,0,0"
Name="textBlock10" Text="Hardware Version:" VerticalAlignment="Top"
Width="200" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,220,0,0"
Name="txtHardwareVersion" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,251,0,0"
Name="textBlock12" Text="Total Memory:" VerticalAlignment="Top" Width="200"
/>
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,251,0,0"
Name="txtTotalMemory" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,287,0,0"
Name="textBlock14" Text="Current Memory:" VerticalAlignment="Top"
Width="200" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,287,0,0"
Name="txtCurrentMemory" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,323,0,0"
Name="textBlock16" Text="Peak Memory:" VerticalAlignment="Top" Width="200"
/>
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,323,0,0"
Name="txtPeakMemory" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,394,0,0"
Name="textBlock18" Text="Power Source:" VerticalAlignment="Top" Width="200"
/>
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,394,0,0"
Name="txtPowerSource" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,430,0,0"
Name="textBlock20" Text="Keyboard Installed:" VerticalAlignment="Top"
Width="200" />

```

```

<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,430,0,0"
Name="txtkeyboardinstalled" Text="" VerticalAlignment="Top" Width="219" />
<Button Content="Get Details" Height="72" HorizontalAlignment="Left"
Margin="46,520,0,0" Name="button1" VerticalAlignment="Top" Width="359"
Click="button1_Click" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,467,0,0"
Name="textBlock3" Text="Keyboard Deployed:" VerticalAlignment="Top"
Width="200" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,467,0,0"
Name="txtKeyboardDeployed" Text="" VerticalAlignment="Top" Width="219" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="12,359,0,0"
Name="textBlock1" Text="Maximum Memory:" VerticalAlignment="Top"
Width="200" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="218,359,0,0"
Name="txtMaximumMemory" Text="" VerticalAlignment="Top" Width="219" />
</Grid>
</Grid>

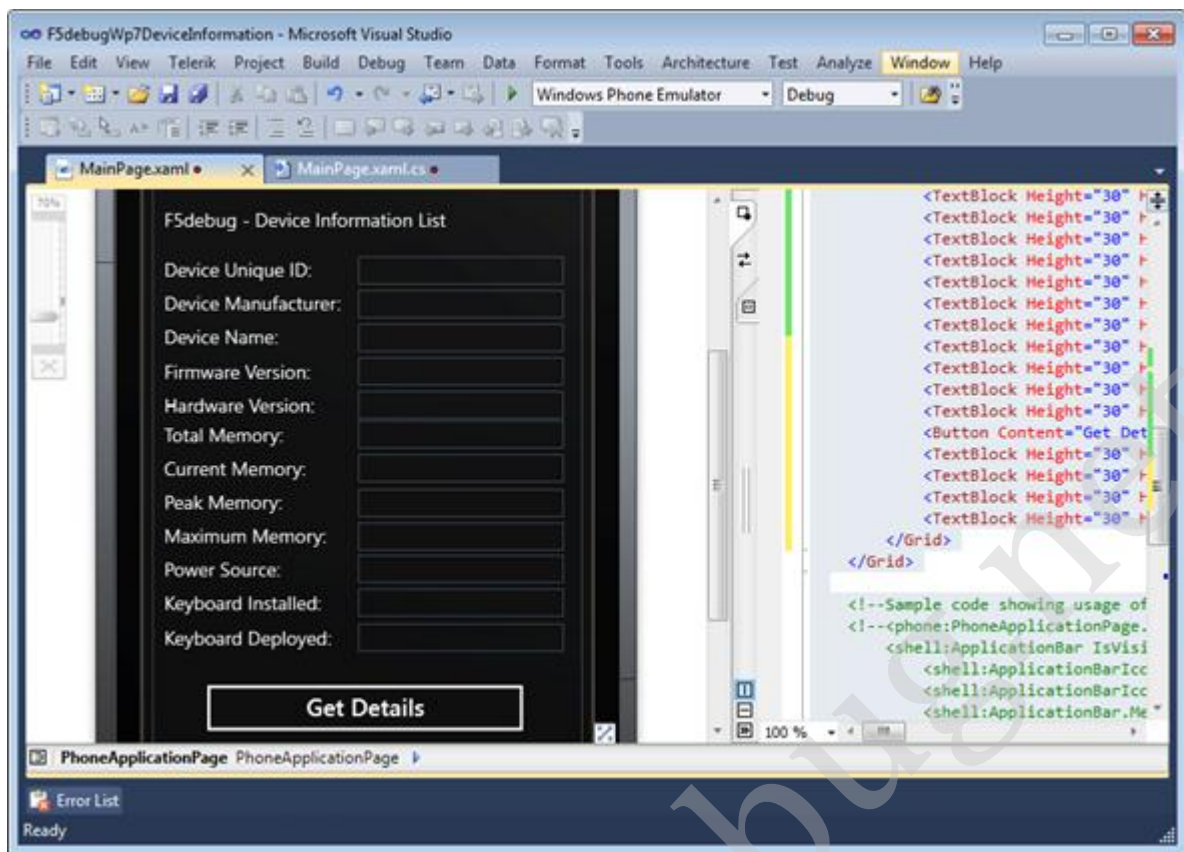
```

```

<!--Sample code showing usage of ApplicationBar-->
<!--<phone:PhoneApplicationPage.ApplicationBar>
<shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
<shell:ApplicationBarIconButton IconUri="/Images/appbar_button1.png"
Text="Button 1"/>
<shell:ApplicationBarIconButton IconUri="/Images/appbar_button2.png"
Text="Button 2"/>
<shell:ApplicationBar.MenuItems>
<shell:ApplicationBarMenuItem Text="MenuItem 1"/>
<shell:ApplicationBarMenuItem Text="MenuItem 2"/>
</shell:ApplicationBar.MenuItems>
</shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar-->

</phone:PhoneApplicationPage>

```



Now let us start with our code to get the device information and load it to the respective text blocks one by one as shown in the code below. We can use the Device Status class to get the details of the device properties except the device unique ID. We will write the code later to get the unique id of the device.

### **C# Code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Info;
using System.Windows.Threading;

namespace F5debugWp7DeviceInformation
{
    public partial class MainPage : PhoneApplicationPage
    {
        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }
    }
}
```

```

private void button1_Click(object sender, RoutedEventArgs e)
{
    txtCurrentMemory.Text =
DeviceStatus.ApplicationCurrentMemoryUsage.ToString();
    txtMaximuMemory.Text = DeviceStatus.ApplicationMemoryUsageLimit.ToString();
    txtPeakMemory.Text = DeviceStatus.ApplicationPeakMemoryUsage.ToString();
    txtTotalMemory.Text = DeviceStatus.DeviceTotalMemory.ToString();

    txtDeviceManufacturer.Text = DeviceStatus.DeviceManufacturer.ToString();
    txtDeviceName.Text = DeviceStatus.DeviceName.ToString();

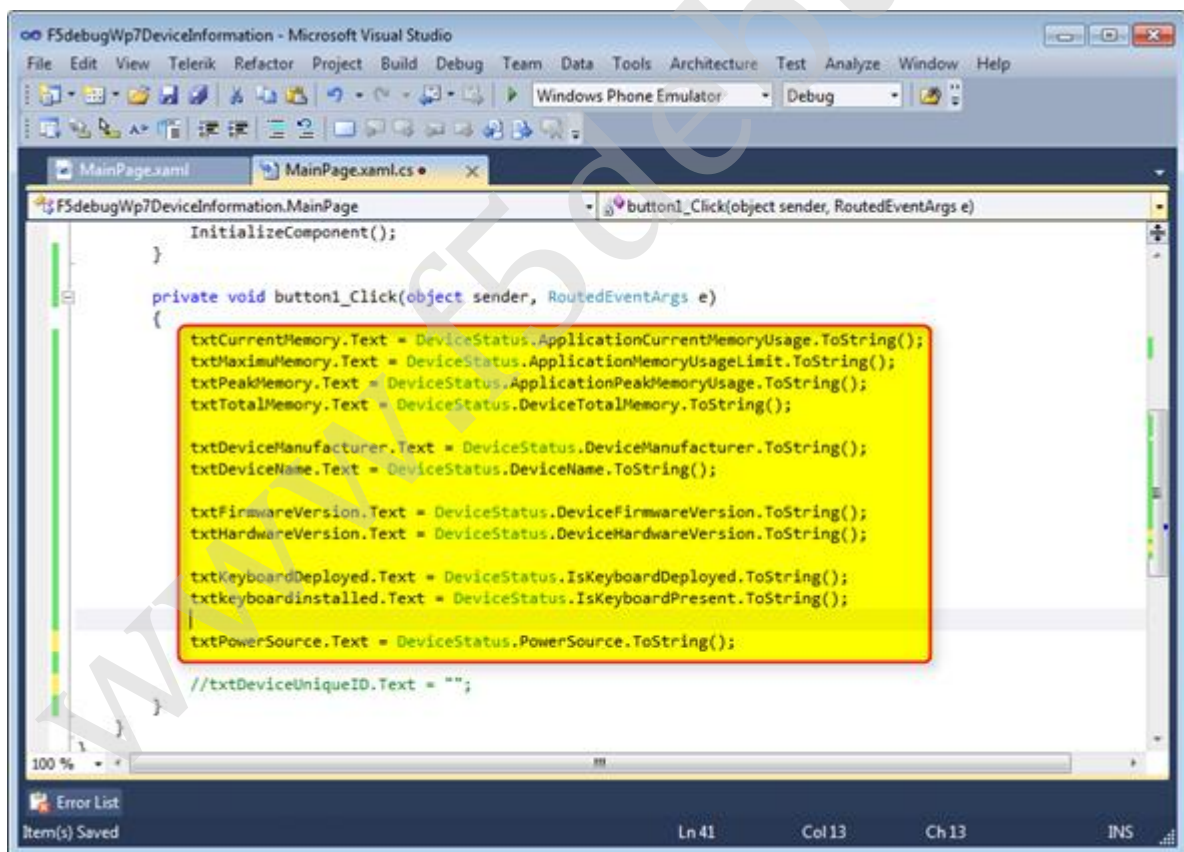
    txtFirmwareVersion.Text = DeviceStatus.DeviceFirmwareVersion.ToString();
    txtHardwareVersion.Text = DeviceStatus.DeviceHardwareVersion.ToString();

    txtKeyboardDeployed.Text = DeviceStatus.IsKeyboardDeployed.ToString();
    txtkeyboardinstalled.Text = DeviceStatus.IsKeyboardPresent.ToString();

    txtPowerSource.Text = DeviceStatus.PowerSource.ToString();

    //txtDeviceUniqueID.Text = "";
}
}
}

```



Now let us write the code for getting the Device Unique ID as shown in the code below. We use the `DeviceExtendedProperties` property `TryGetValue` to get the Unique ID, since we get it as a byte array we do a conversion to the string and assign it to the text block as shown below.

## **C# Code:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Info;
using System.Windows.Threading;

namespace F5debugWp7DeviceInformation
{
    public partial class MainPage : PhoneApplicationPage
    {
        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, RoutedEventArgs e)
        {
            txtCurrentMemory.Text =
            DeviceStatus.ApplicationCurrentMemoryUsage.ToString();
            txtMaximumMemory.Text = DeviceStatus.ApplicationMemoryUsageLimit.ToString();
            txtPeakMemory.Text = DeviceStatus.ApplicationPeakMemoryUsage.ToString();
            txtTotalMemory.Text = DeviceStatus.DeviceTotalMemory.ToString();

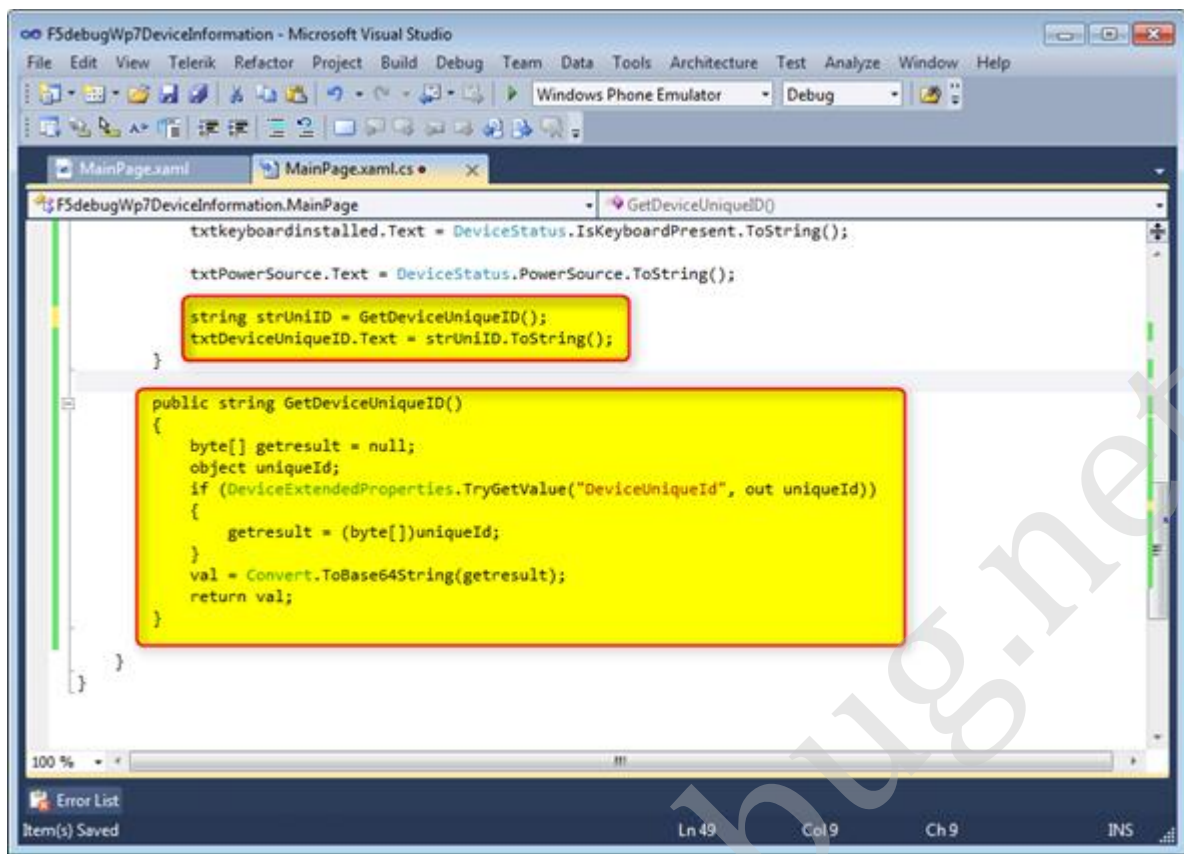
            txtDeviceManufacturer.Text = DeviceStatus.DeviceManufacturer.ToString();
            txtDeviceName.Text = DeviceStatus.DeviceName.ToString();

            txtFirmwareVersion.Text = DeviceStatus.DeviceFirmwareVersion.ToString();
            txtHardwareVersion.Text = DeviceStatus.DeviceHardwareVersion.ToString();

            txtKeyboardDeployed.Text = DeviceStatus.IsKeyboardDeployed.ToString();
            txtkeyboardinstalled.Text = DeviceStatus.IsKeyboardPresent.ToString();

            txtPowerSource.Text = DeviceStatus.PowerSource.ToString();

            //txtDeviceUniqueID.Text = "";
        }
    }
}
```



Now we are done with our code, let us build and execute the application by pressing F5 directly from the keyboard and we can see the application loaded to the Windows Phone 7 Emulator as shown in the screen below with the expected output.



## Conclusion:

So in this article we have seen how to get the device information details and the device unique id using the Device Status class in detail.

Thanks for reading my article. If you like my blog and if you are interested in getting the latest updates on new articles, kindly follow me through one of these options.

