

**Name of Experiment:** System Specification related to Software and Hardware that includes OS version, CLR version, Tick Count and HasShutDownStarted.

**Exp No:** MC12

**Background:** Student should have basic knowledge of C#.

**Summary:** System information related to software and hardware are abstracted from the end-user. These information is not require to solve generic issues, while sometimes user want to know the OS version, CLR version in order to either update the current version or to reinstall the specific application software.

**Learning Objective:** Using this experiment, student should able to analyse the system information on running the same experiment on various mobile devices in order to get the complete understanding of the system software.

**Target Platforms:** This experiment is tested on Windows Phone Emulator and Nokia Lumia 800.

**Procedure:**

Step1. Repeat the steps [1-4] as in experiment no MC1. [i.e. Hello World]

Step2. Design the UI layout. [refer source code section]

Step3. Define the String Builder.

Step4. Use the class System.Environment in order to display the various system specific values such as OS version including Major, Minor, Revision and Build. Similarly for CLR version, HasShutDownStarted, and Tick Count for getting the timespan in milisec since the experiment is running. [refer source code section]

Step5. Save all the changes by pressing Ctrl+ S.

Step6. Press F5, in order to debug the experiment in Windows Phone Emulator.

Step7. Hence, by this way we are able to get the system specific information.

**Source Code**

**Comments**

**MainPage.xaml**

```
<!--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="ExperimentTitle" Text="Exp
No.MC12" TextAlignment="Right" Style="{StaticResource
PhoneTextNormalStyle}"/>
    <TextBlock x:Name="ApplicationTitle"
Text="Mobile Computing" Style="{StaticResource
PhoneTextNormalStyle}"/>
    <TextBlock x:Name="PageTitle" Text="Software
Specification" 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">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="227"/>
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="150"/>
        <RowDefinition Height="150"/>
        <RowDefinition Height="100"/>
        <RowDefinition Height="100"/>
        <RowDefinition Height="100"/>
    </Grid.RowDefinitions>

    <TextBlock Text="OS Version" Height="75"
Grid.Column="0" Grid.Row="0"/>
    <TextBlock x:Name="txtOSVersion" Height="150"
Grid.Column="1" Grid.Row="0"/>
    <TextBlock Text="CLR Version" Height="75"
Grid.Column="0" Grid.Row="1"/>
    <TextBlock x:Name="txtCLR" Text=" "
Height="150" Grid.Column="1" Grid.Row="1"/>
    <TextBlock Text="HasShutDownStarted"
Height="75" Grid.Column="0" Grid.Row="2"/>
    <TextBlock x:Name="txtHasShut" Text=" "
Height="75" Grid.Column="1" Grid.Row="2"/>
    <TextBlock Text="TickCount" Height="75"
Grid.Column="0" Grid.Row="3"/>
    <TextBlock x:Name="txtCount" Height="75"
Grid.Column="1" Grid.Row="3"/>
</Grid>

```

← Exp No. MC12(Experiment Title)  
← Mobile Computing (Application Title)  
← Software Specification (Page Title)

### MainPage.xaml.cs

```

using Microsoft.Phone.Controls;
using System.Text;

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

```

← String Builder refers System.Text

```

        StringBuilder str=new StringBuilder();
        str.AppendFormat("Platform\t{0}",
System.Environment.OSVersion.Platform.ToString());
        str.Append("\n");
        str.AppendFormat("Build\t{0}",
System.Environment.OSVersion.Version.Build.ToString());
        str.Append("\n");
        str.AppendFormat("Major\t{0}",
System.Environment.OSVersion.Version.Major.ToString());
        str.Append("\n");

str.AppendFormat("Minor\t{0}",
System.Environment.OSVersion.Version.Minor.ToString());
        str.Append("\n");
        str.AppendFormat("Revision\t{0}",
System.Environment.OSVersion.Version.Revision.ToString());
        txtOSVersion.Text = str.ToString();
        txtHasShut.Text =
System.Environment.HasShutdownStarted.ToString();
        StringBuilder str1 = new StringBuilder();
        str1.AppendFormat("Build\t{0}",
System.Environment.Version.Build.ToString());
        str1.Append("\n");
        str1.AppendFormat("Major\t{0}",
System.Environment.Version.Major.ToString());
        str1.Append("\n");
        str1.AppendFormat("Minor\t{0}",
System.Environment.Version.Minor.ToString());
        str1.Append("\n");
        str1.AppendFormat("Revision\t{0}",
System.Environment.Version.Revision.ToString());
        txtCLR.Text = str1.ToString();
        txtCount.Text =
System.Environment.TickCount.ToString();

    }
}
}

```

← Fetching the various Subversion of OS.

← Fetching the Boolean value, whether the device is going to Shut Down or Not.

← Fetching the various Subversion of Common Language Runtime (i.e. CLR).

← Fetching the Time span in milisec since the system has started.

## Screenshots



Fig. No1 Main Page

**Observations:** It is observed by the developer that system specific information is also served as a vital parameter during the analysis of the various mobile devices.