ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

*International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*
*Vol. 2, Issue 2, February 2013*

# Creating Windows Phone 7 Applications

Chesti Altaff Hussain[1], Javvaji Venkata Naga Madhusudhana Rao[2], Navakoti kartheek[3]

Assistant Professor, Department of ECE, Bapatla Engineering College, Bapatla - 522101, India[1]

Final Year, Department of ECE, Bapatla Engineering College, Bapatla - 522101, India[2]

Final Year, Department of ECE, Bapatla Engineering College, Bapatla - 522101, India[3]

**Abstract**: Now a day's an electronic gadget, the mobile phone became a part of human day to day life, it occupied such an important place because, the smart phones now evolving innovatively are performing almost all the tasks of normal human being basic needs to make his life much easier than ever, in these phones we use different varieties of operating systems among them we here selected the "windows mobile platform operating system" which is user friendly and robust in the sense of security, we here implemented a tiny application in this operating system to display the "system time in digital format".

**Keywords:** Windows mobile platform operating system Smart phone**,** Microsoft Silver light tools, Visual studio 2010 version.

## I. INTRODUCTION

In this paper we show you the basics of writing applications for Windows Phone 7 using the C# programming language with the Silverlight and XNA 2D frameworks. This is .NET flat form. It is very easy to interface. It is very easy understanding. *Windows Phone 7 Silverlight Development Step by Step* by Andy Wigley & Peter Foot offers a more tools-oriented approach. Although Michael Stroh's *Windows Phone 7 Plain & Simple* is a guide to *using* the phone rather than developing for it, WINDOWS PHONE7 application creates a friendly environment to the user. Main advantages of this system is reducing power wastage saving time and also for security purposes. This application runs on windows mobile platform, the application has a graphical interface providing in convenient way. Microsoft is hard at work in developing its killer mobile OS. The new OS is now a total rewrite of the older Windows Mobile platform and sports many features that you have come to expect of a modern mobile OS.

Paper Organization:

Section I introduces the Requirement of this paper. Section II Gives the description of the tools used to develop the windows mobile phone application. Section III describes the working interface of windows emulators. Section IV shows how the proposed application was developed,and the necessary program code. Section V gives theinformation regarding the output of the code implemented.

## II. TOOLS

In the series of Windows Phone 7 articles on MobiForge. We will bring you on a development journey to explore the applications.And in this first installment we will explore the tools and see how you could start writing your application quickly

Various tools used for the development of windows phone 7 application are

- Visual studio 2010 express for windows phone
- Windows phone emulator resources
- Silverlight 4 tools for Visual studio
- XNA game studio 4.0
- Microsoft expression blend for windows phone

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

*International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*
*Vol. 2, Issue 2, February 2013*

- **VISUAL STUDIO 2010 EXPRESS FOR WINDOWS PHONE**

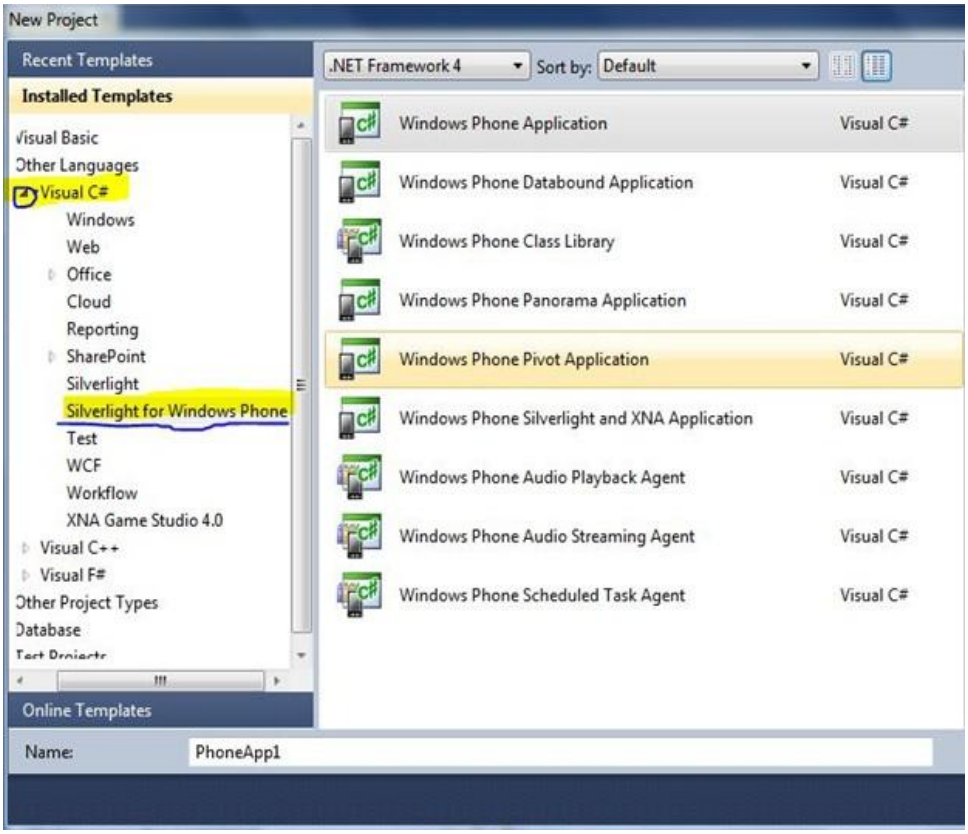


Figure 1: Iinterface for new application page.

In the above we are seeing interface for visual studio 2010 new project for Silverlight for windows phone and that is the language of visual c#. After we are clicking on the 'ok' button, new application page will be created.

After opening that page we seeing the phone interface and tool boxes and solution explorer and properties blocks. Take the windows phone control from the tool box and drop into the phone page. Automatically the properties will be displayed on the properties block. And solution explorer having the different files of our project.

- AppManifest.xml – application manifest file used for generating the XAP file.
- AssemblyInfo.cs – configuration file containing information for the assembly, such as title, description, company, GUID, etc.
- WMAAppManifest.xml – contains manifest information for the application, such as application capabilities, background image, etc
- App.xaml.cs – code-behind foe the App.xamlfile. This is where you implement the various methods to handle the application's life cycle.
- ApplicationIcon.png – the icon that is displayed when you install your application on the phone.
- Background.png – the image to display when your application is "pinned" to the start screen.
- Mainpage.xaml – the main page to load foe=r your application.
- Mainpage.xaml.cs – the code-behind for the main page of your application.
- SplashScreenImage.jpg – the splash screen that will be displayed when you first start your application.

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

*International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*
*Vol. 2, Issue 2, February 2013*

## III. WINDOWS PHONE EMULATOR RESOURCES

Your program will quickly build and in the status bar you'll see the text "Connecting to Windows Phone 7 Emulator…" The first time you use the emulator during a session, it might take a little time to start up. If you leave the emulator running between edit/build/run cycles, Visual Studio doesn't need to establish this connection again. Soon the phone emulator will appear on the desktop and you'll see the opening screen, followed soon by this little do-nothing Silverlight program as it is deployed and run on the emulator. On the phone you'll see pretty much the same image you saw in the design view. The phone emulator has a little floating menu at the upper right that comes into view when you move the mouse to that location. You can change orientation through this menu, or change the emulator size. By default, the emulator is displayed at 50% actual size, about the same size as the image on this page. When you display the emulator at 100%, it becomes enormous, and you might wonder "How will I ever fit a phone this big into my pocket?" The difference involves pixel density. Your computer screen probably has about 100 pixels per inch. (By default, Windows assumes that screens are 96 DPI.) The screen on an actual Windows


Figure 2: Emulator interface

Phone 7 device is more than 2½ times that. When you display the emulator at 100%, you're seeing all the pixels of the phone's screen, but at about 250% their actual size. You can terminate execution of this program and return to editing the program either though Visual Studio (using Shift-F5 or by selecting Stop Debugging from the Debug menu) or by clicking the Back button on the emulator. Don't exit the emulator itself by clicking the X at the top of the floating menu! Keeping the emulator running will make subsequent deployments go much faster.

While the emulator is still running, it retains all programs deployed to it. If you click the arrow at the upper-right of the Start screen, you'll get a list that will include this program identified by the text "Silverlight Hello Phone" and you can run the program again. The program will disappear from this list when you exit the emulator.

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

*International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*
*Vol. 2, Issue 2, February 2013*

## IV. CREATING WINDOWS PHONE 7 APPLICATION

Here first we are creating simple application that is display the time on the screen. Here first open the visual studio 2010, and select the c# language and select the tool Silverlight for windows phone 7, and take the application page. Here we are seeing the application page design interface and Xmalcode of the page.

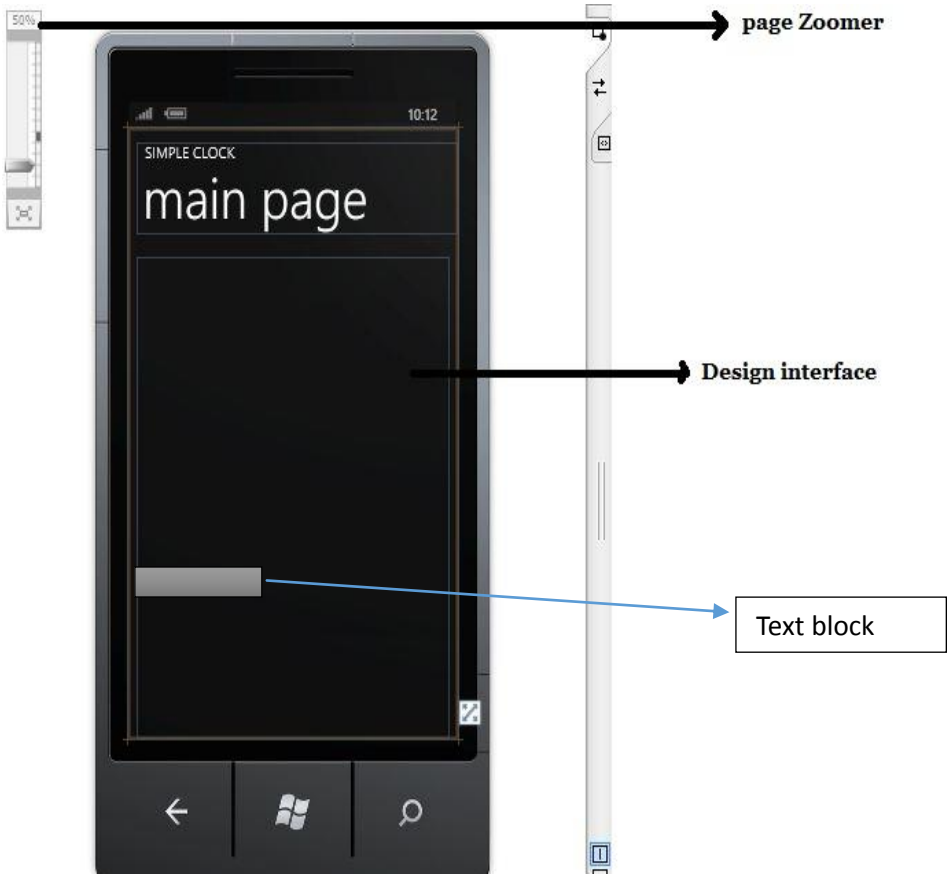


Figure3 : Design page interface

After taking the text box, that box was dropped on the design page, and change the properties of the design page and text block. In the properties block. The design page XMAL code will be generated automatically in background. That is shown in below.

```
<phone:PhoneApplicationPage
x:Class="SilverlightSimpleClock.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"
FontFamily="{StaticResourcePhoneFontFamilyNormal}"
FontSize="{StaticResourcePhoneFontSizeNormal}"
 Foreground="{StaticResourcePhoneForegroundBrush}"
SupportedOrientations="PortraitOrLandscape" Orientation="Portrait"
```

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875



```
mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
shell:SystemTray.IsVisible="True" Loaded="PhoneApplicationPage_Loaded">

<!--LayoutRoot contains the root grid where all other 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="SIMPLE CLOCK" Style="{StaticResourcePhoneTextNormalStyle}"/>
<TextBlock x:Name="PageTitle" Text="main page" 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 Name="txtblk"
HorizontalAlignment="Center"
VerticalAlignment="Center" />
</Grid>
</Grid>

<!-- Sample code showing usage of ApplicationBar
<phone:PhoneApplicationPage.ApplicationBar>
<shell:ApplicationBarIsVisible="True" IsMenuEnabled="True">
<shell:ApplicationBarIconButton x:Name="appbar_button1" IconUri="/Images/appbar_button1.png" Text="Button
1"></shell:ApplicationBarIconButton>
<shell:ApplicationBarIconButton x:Name="appbar_button2" IconUri="/Images/appbar_button2.png" Text="Button
2"></shell:ApplicationBarIconButton>
<shell:ApplicationBar.MenuItems>
<shell:ApplicationBarMenuItem x:Name="menuItem1" Text="MenuItem 1"></shell:ApplicationBarMenuItem>
<shell:ApplicationBarMenuItem x:Name="menuItem2" Text="MenuItem 2"></shell:ApplicationBarMenuItem>
</shell:ApplicationBar.MenuItems>
</shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar>
   -->
</phone:PhoneApplicationPage>
```

This code will be displayed on the XMAL page. In the above code we will see the all properties of the application page and totally XMAL code of the design page. After this design we have to write the working code. That is application page working code. These design issues all are in the Main page.Xmal. The working code will be written in Main page.Xmal.cs page. This will be generatedwhen double click on the design page.

```
using System;
usingSystem.Windows.Threading;
usingMicrosoft.Phone.Controls;
```

These all are libraries. And writing the code we are using the different types of methods. In the above we are using the DispatcherTimer is the variable, and method is DateTime.Now.Tostring ();

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

*International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*
*Vol. 2, Issue 2, February 2013*

```
MainPage.xaml   MainPage.xaml.cs
SilverlightSimpleClock.MainPage        OnTimerTick(object sender, EventArgs args)
using System;
using System.Windows.Threading;
using Microsoft.Phone.Controls;

namespace SilverlightSimpleClock
{
    public partial class MainPage : PhoneApplicationPage
    {
        public MainPage()
        {
            InitializeComponent();

            DispatcherTimer tmr = new DispatcherTimer();
            tmr.Interval = TimeSpan.FromSeconds(1);
            tmr.Tick += OnTimerTick;
            tmr.Start();
        }

        void OnTimerTick(object sender, EventArgs args)
        {
            txtblk.Text = DateTime.Now.ToString();
        }
    }
}
```

Figure 4 : working code of the clock

After writing the code save the project. And run with the emulator. Emulator will be display the output of the application page.

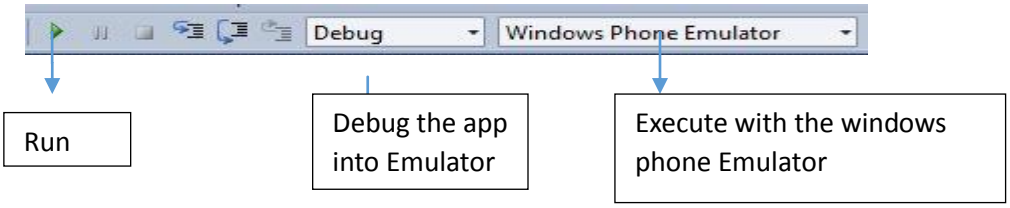


Figure 5: Execute interface

## V.OUTPUT OF THE APPLICATION PAGE

Run with the Emulator. And see the output.


Figure 6: output in Emulator

ISSN (Print) : 2320 – 3765
ISSN (Online) : 2278 – 8875

*International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*
*Vol. 2, Issue 2, February 2013*

## VI. CONCLUSION

The application we designed here based on windows mobile phone platform operating system, is a quite simple one, and if we wish to enhance the features of the application.We can use the application in real time world, such as controlling of home appliances using a very convenient smart and a flexible interface including good graphics makes our daily life much ease and comfortable, we showed here a very tiny example which can embrace the enthusiastic.

## REFERENCES

1. Charles Petzold, " Programming Windows Phone 7",pp.393-589,2010
2. Brian Faucher, "Windows phone7 application development",pp.3-37,2011
3. Nick Randolph,Christopher Fairbairn, "Professional Windows Phone 7 Application Development: Building Applications and Games Using Visual Studio, Silverlight, and XNA",pp.123-149,2010
4. Rob Cameron, "Pro Windows Phone 7 Development",edition.2,pp.1-305,2011
5. Windows Phone 7 Application Development - from internet Microsoft windows.
6. Phone 7 development center-general thesis submitted documents from open source internet.
7. Geoffrey Hunt ,lebogangmadise, "coding workshop for windows phone", 2012
8. Nick lecrenski, karl Watson, Robert fonseca-ensor," Beginning windows phone 7 application development",pp.1-99, 2011.
9. Henry Lee ,EugenebChuvyrov ,"Windows phone application Development", pp 1-15.
10. Nick Radolph, Christopher Fairbairn ,"Professional Windows Phone 7 Application Development",pp 1-13.
11. Adam Dawes ,"Windows phone 7 Game Development",pp 1-150.
12. MichealStroh,"PlainAnd Simple Windows Phone 7".
13. Alex Horner,ScottDensmore,DominicBetts,Fredricore,Jose Gallardo Salzar,"Windows Phone 7 Developers Guide".
14. Jonathan Marbutt,RobbSchieferJr ,"Windows Phone 7 Silverlight Cook Book".
15. Jon Westfall,"Windows Phone 7 Made simple"pp 1-200.

## BIOGRAPHY

**Chesti Altaff Hussain** is working as an Assistant Professor in the Department Of E.C.E, Bapatla Engineering College, Bapatla-522101, Guntur District, Andhra Pradesh, India.

**Javvaji Madhusudhana Rao** is studying Final Year B.Tech in the Department Of E.C.E, Bapatla Engineering College, Bapatla-522101, Guntur District, Andhra Pradesh, India.

**Navakoti Kartheek** is studying Final Year B.Tech in the Department Of E.C.E, Bapatla Engineering College, Bapatla-522101, Guntur District, Andhra Pradesh, India.