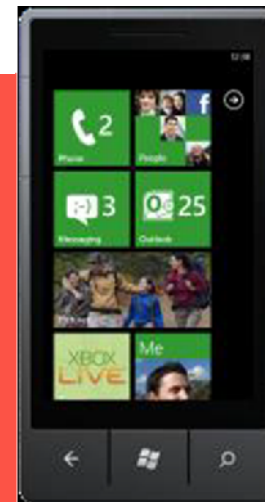




Advance Windows Phone Development



7

Akber Alwani

Window Phone 7 Development
EP.NET Professionals User Group

<http://www.epdotnet.com>

Agenda



Page Navigation
Application Bar and System tray
Orientation-Aware Pages
Pivots and Panorama
Isolated Storage
Q&A



Page Navigation

Graphics Composition

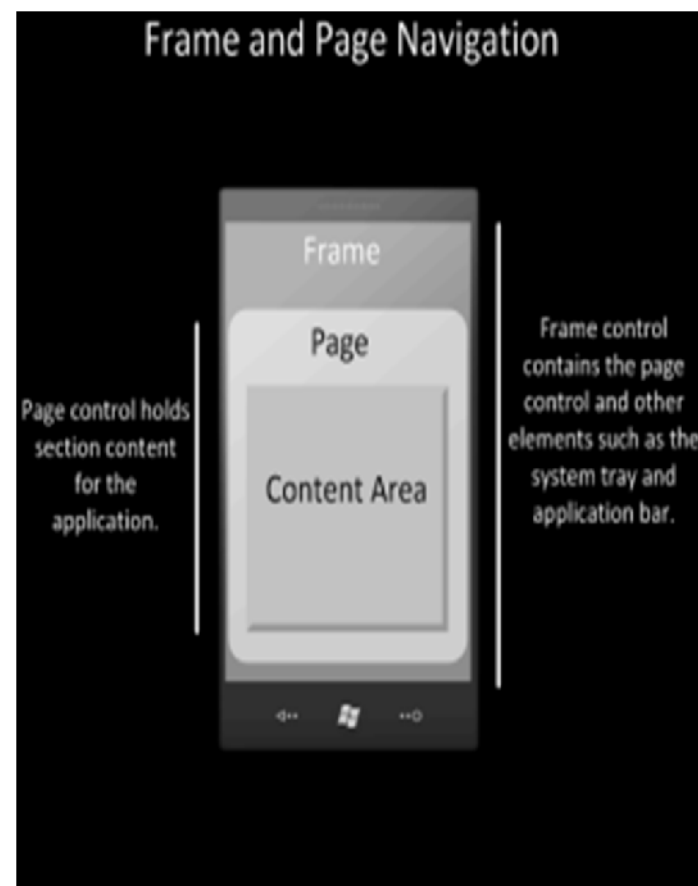


- Shell frame composes all UI into a single scene
- Central page management enables cross-app UI transitions and other effects
- System-wide Z-order enforcement emphasizes core phone functionality



Frame and Page

- **Frame**
 - Top-level container control
 - PhoneApplicationFrame class
 - Contain the page control and system elements such as system tray and application bar
- **Page**
 - Files the entire content region of the frame
 - PhoneApplicationPage-derived class
 - Contains a title
 - Optionally surfaces its own application bar

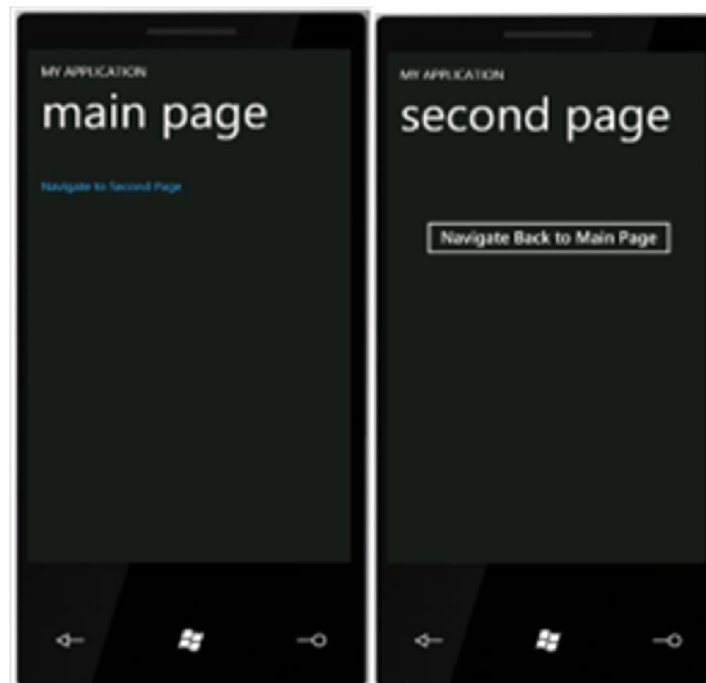




Page Navigation

- Silverlight on Windows Phone uses a Page-based navigation model
 - Similar to web page model
 - Each page identified by a URI
 - Each page is essentially stateless

```
private void hyperlinkButton1_Click(object sender,
RouteEventArgs e)
{
    NavigationService.Navigate(new Uri("/SecondPage.xaml",
UriKind.RelativeorAbsolute) )
}
```

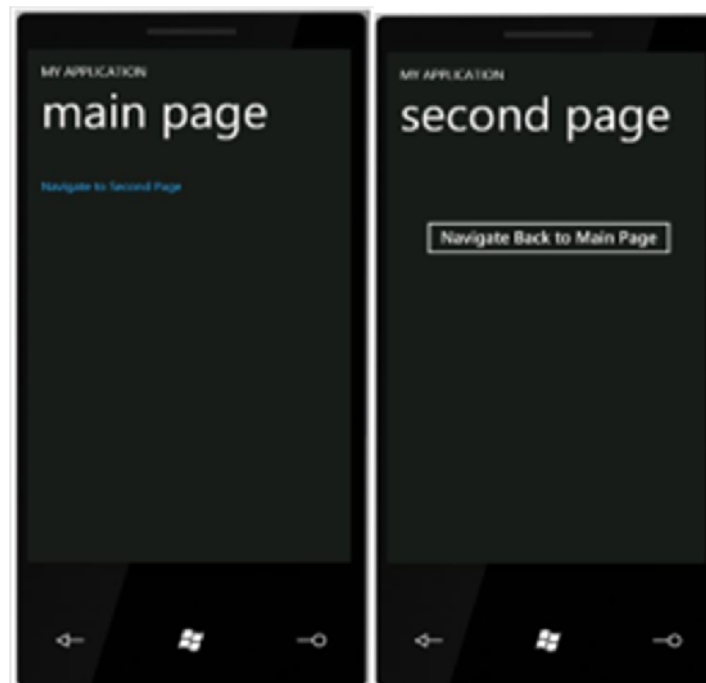




Page Navigation

- Silverlight on Windows Phone uses a Page-based navigation model
 - Similar to web page model
 - Each page identified by a URI
 - Each page is essentially stateless

```
private void hyperlinkButton1_Click(object sender,
RouteEventArgs e)
{
    NavigationService.Navigate(new Uri("/SecondPage.xaml",
UriKind.RelativeorAbsolute) )
}
```



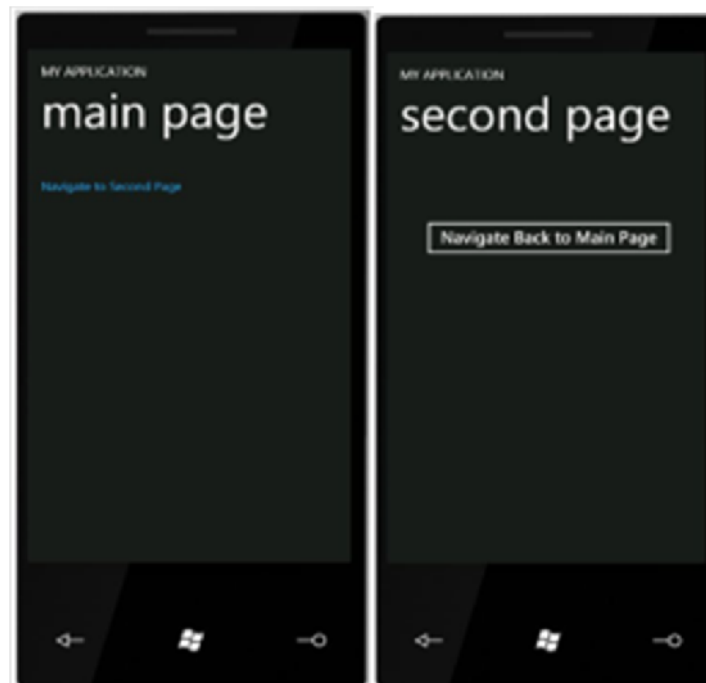


Navigation Back

- Application can provide controls on navigation back to preceding page

```
private void Button1_Click(object sender, RoutedEventArgs e)
{
    NavigationService.GoBack();
}
```

- The hardware Back key will also navigate back to preceding page
 - No code require- built-in behavior





Passing Data Between Pages

- Can pass string data between pages using query strings:

```
1. private void passParam_Click(object sender, RoutedEventArgs e)
2. {
3.     NavigationService.Navigate(new Uri("/SecondPage.xaml?msg=" +
4.         textBox1.Text, UriKind.Relative));
5. }
```

- On destination page:

```
1. protected override void OnNavigatedTo(
2.     System.Windows.Navigation.NavigationEventArgs e)
3. {
4.     base.OnNavigatedTo(e);
5.     string msg = "";
6.     if (NavigationContext.QueryString.TryGetValue("msg", out msg))
7.         textBlock1.Text = msg;
8. }
```



Passing Objects Between Pages

- Often, you will need to pass a data object from one page to another
 - For example, the user selects an item in a list and navigates to
- One solution for this is to store your ViewModel (i.e. data) on App Class
 - Global to whole application
- Pass the selected item index in query string





Navigation

Demo



Application Bar



Application Chrome

System Tray and Application Bar

■ System Tray

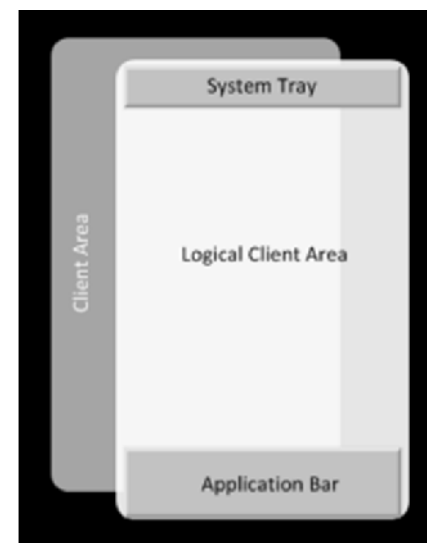
- System owned indicator area that displays system-level status information

- App can show/hide

Microsoft.Phone.Shell.[SystemTray](#).Invisible= [false](#);

■ Application Bar

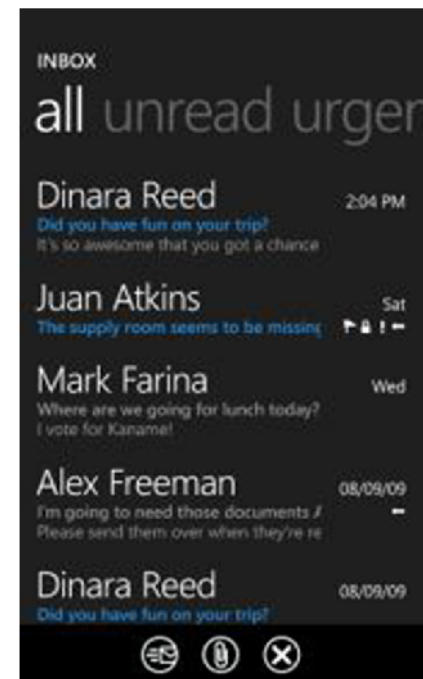
- Area where application can display buttons for the most common tasks
- Can display pop-up menu for less common tasks



Application Bar



- Use the ApplicationBar instead of creating your own menu system
- Up to 4 buttons plus optional menu
 - Swipe up the bar to bring up the menu
- Don't fill all 4 slots if not needed
- Use white foreground on transparent background for icons
 - System will colorize button according to users selected theme





Application Bar in Xaml

```
1 <phone:PhoneApplicationPage
2   xClass="MyApp.MainPage"
3   ... >
4
5 <phone:PhoneApplicationPage.ApplicationBar>
6   <shell:ApplicationBar xName="AppBar" IsMenuEnabled="True">
7     <shell:ApplicationBar.Buttons>
8       <shell:ApplicationBarIconButton xName="NewContactButton"
9         IconUri="Images/appbarnewrest.png" Text="New"
10        Click="NewContactButton_Click"/>
11      <shell:ApplicationBarIconButton xName="SearchButton"
12        IconUri="Images/appbarfeature.searchrest.png"
13        Text="Find" Click="SearchButton_Click"/>
14    </shell:ApplicationBar.Buttons>
15    <shell:ApplicationBar.MenuItems>
16      <shell:ApplicationBarMenuItem xName="GenerateMenuItem"
17        Text="Generate Data" Click="GenerateMenuItem_Click" />
18      <shell:ApplicationBarMenuItem xName="ClearMenuItem"
19        Text="Clear Data" Click="ClearMenuItem_Click" />
20    </shell:ApplicationBar.MenuItems>
21  </shell:ApplicationBar>
22 </phone:PhoneApplicationPage.ApplicationBar>
```



Application Bar

Demo

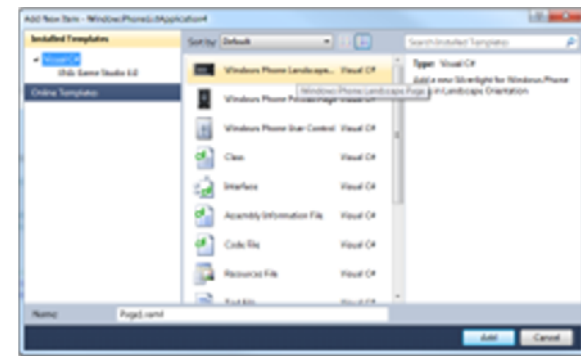


Orientation-Aware Pages

Orientation Support



- In Visual-Studio 'Add New Item', you can add a Landscape Page or a Portrait Page
- In fact, any page can be either – or both
- PhoneApplicationPage.SupportedOrientation property states what orientations the page supports
 - Portrait
 - Landscape
 - PortraitOrLandscape
- If set to PortraitOrLandscape, page will re-orientate itself automatically when user rotates the phone
- You cannot force a page to re-orientate in code
- You can force a page to always use a specific orientation by setting the SupportedOrientatio property to the value you require





Orientation-Aware Pages

```
1. <phone:PhoneApplicationPage
2.     x:Class="OrientationSample.MainPage"
3.     ...
4.     SupportedOrientations="PortraitOrLandscape" Orientation="Portrait"
5.     mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768">
```

- Can use a ScrollView to Scroll content when in landscape

```
1. <ScrollViewer x:Name="ContentGrid" Grid.Row="1" VerticalScrollBarVisibility="Auto">
2.     <StackPanel Background="{StaticResource PhoneBackgroundBrush}" Height="632"
3.         Width="466">
4.         <!--Adding various controls and UI elements.-->
5.         <Button Content="This is a Button" Width="254" />
6.         <Rectangle Width="100" Height="100" Margin="12,0" HorizontalAlignment="Left"
7.             Fill="{StaticResource PhoneAccentBrush}"/>
8.         ...
9.     </StackPanel>
10. </ScrollViewer>
```



Handling Orientation
Changes

Demo



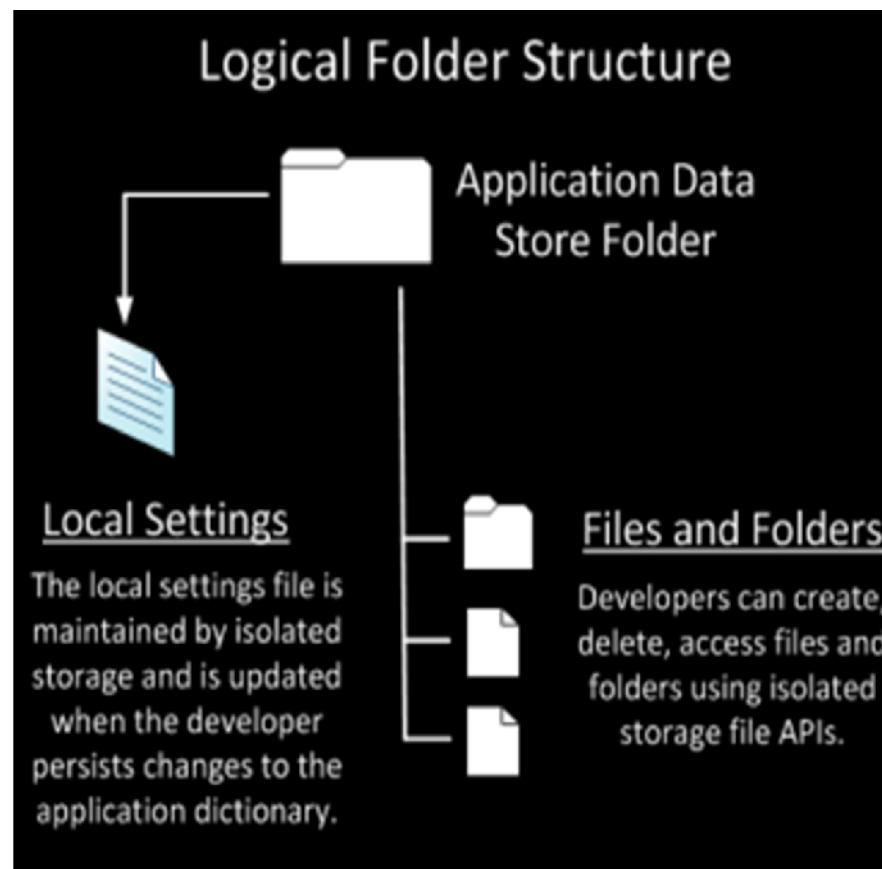
Storing Data in Isolated Storage

Settings and other data



Isolated Storage

- All I/O operations restricted to isolated storage
 - Use Isolated File storage to create a files and folders structure hierarchy
 - Use Isolated Settings storage to store application settings





Isolated Storage Classes

- `IsolatedStorageFile`
 - Represent an isolated storage area containing files and directories
- `IsolatedFileStream`
 - Exposes a files stream access to a file stored within isolated storage
- `IsolatedStorageSettings`
 - Dictionary `<(Of <(Tkey,Tvalue)>)>` that stores key-value pairs in isolated storage.



Isolated Storage

```
using System.IO;
```

```
using System.IO.IsolatedStorage;
```

- Before you can use the isolated storage classes directly you need to bring in a couple of namespaces
- You don't need to add any references to your project though
- The isolated storage can be managed and used as a chunk of filestore
 - Named files and folders

Saving Data



```
private void saveGameToIsolatedStorage(string message)
{
    using (IsolatedStorageFile isf =
        IsolatedStorageFile.GetUserStoreForApplication())
    {
        using (IsolatedStorageFileStream rawStream =
            isf.CreateFile(filename))
        {
            StreamWriter writer = new StreamWriter(rawStream);
            writer.WriteLine(message); // save the message
            writer.Close();
        }
    }
}
```



Reading Data

```
string loadString(string filename)
{
    string result = null;
    using (IsolatedStorageFile isf =
            IsolatedStorageFile.GetUserStoreForApplication())
    {
        if (isf.FileExists(filename)) {
            try {
                using (IsolatedStorageFileStream rawStream =
                    isf.OpenFile(filename, System.IO.FileMode.Open)) {
                    StreamReader reader = new StreamReader(rawStream);
                    result = reader.ReadLine();
                    reader.Close();
                }
            } catch {}
        }
    }
    return result;
}
```

Application Settings



```
1.  IsolatedStorageSettings appSettings =  
        IsolatedStorageSettings.ApplicationSettings;  
2.  
3.  string shortName;  
4.  try  
5.  {  
6.      shortName = (string)appSettings["shortName"];  
7.  }  
8.  catch (System.Collections.Generic.KeyNotFoundException)  
9.  {  
10.     // No preference is saved - use/set default  
11.     shortName = "NotSet";  
12.     appSettings["shortName"] = shortName;  
13.  }  
14.  this.shortNameTextBox.Text = shortName;
```



Isolated Storage

Demo



Quota Management

- There is no quotas on Windows Phone!
- Applications must make careful use of space
 - Use only what is necessary
 - Be transparent about storage usage
- Manage application data
 - Delete temporary data and files when no longer required
 - Consider synchronizing or archiving data to cloud to reduce device storage

Q&A



Apps navigate between pages using NavigationService

You can pass data objects between pages by using Query String

An Application Bar can have up to four icons

You can change the display orientation of a Page in code at runtime

System auto orientates LandscapeOrPotrait forms on phone rotation

IsoloatedStorage is subject to 100MB Quota per application

You can create directories and files in isolagestorage

Download All Presentation



<http://bit.ly.epdotnet-wp7>



XNA Game



Pivots and Panorama



Pivots & Panorama

- Best controls to build applications
 - Easy to implement new Windows 7 user experience
- Out-of-the-box behavior matches UX guidelines and system behavior
 - Graphically similar (lists of stuff, off-screen content)
- Break up data into consumable chunks for user
- Physically similar (cyclical, left-to-right, gestures, backstack)