



# Advanced Silverlight

Rob S. Miles | Microsoft MVP | University of Hull, UK  
Andy Wigley | Microsoft MVP | Appa Mundi

Session 3.0

**NOKIA**



# Course Schedule

- Session 1 – Tuesday, August 23, 2011
  - Building Windows Phone Apps with Visual Studio 2010
  - Silverlight on Windows Phone—Introduction
  - **Silverlight on Windows Phone—Advanced**
  - Using Expression to Build Windows Phone Interfaces
  - Windows Phone Fast Application Switching
  - Windows Phone Multi-tasking & Background Tasks
  - Using Windows Phone Resources (Bing Maps, Camera, etc.)
- Session 2 – Wednesday, August 24, 2011
  - Application Data Storage on Windows Phone
  - Using Networks with Windows Phone
  - Tiles & Notifications on Windows Phone
  - XNA for Windows Phone
  - Selling a Windows Phone Application

**NOKIA****Microsoft®**

# Topics

- Silverlight Project Templates
- ApplicationBar
- Page Navigation
- Data Binding
- Silverlight Toolkit

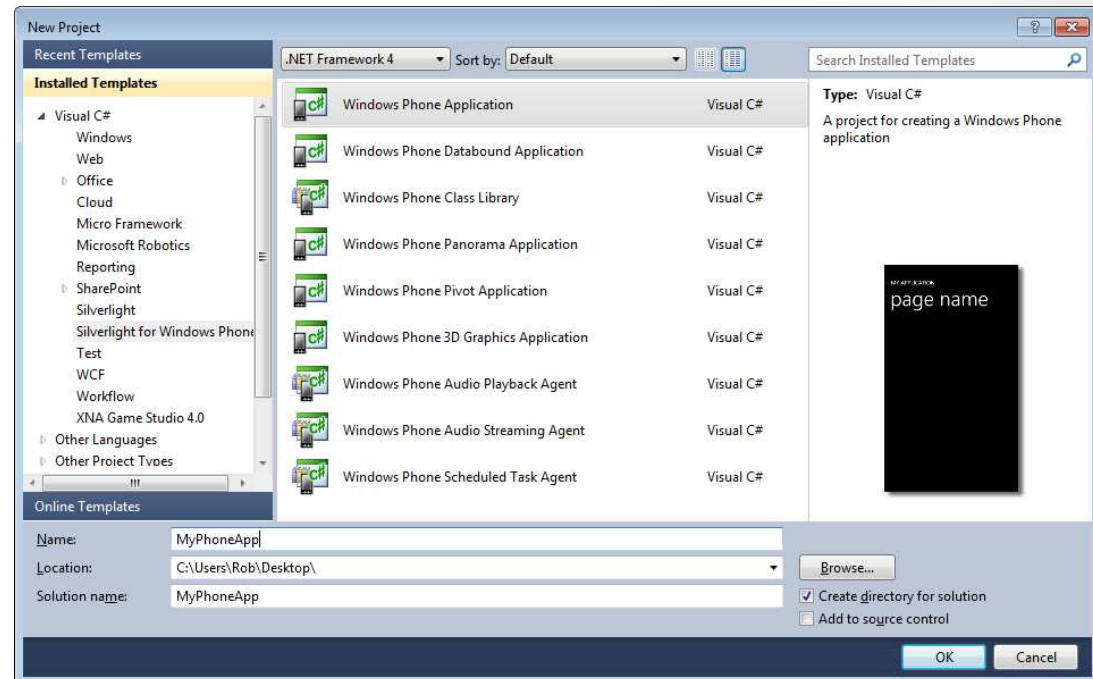
**NOKIA**

**Microsoft®**

# Silverlight Project Templates

# Project Templates and Components

- Visual Studio provides a set of project templates
- Each of them maps onto a particular style of application



# Application Templates

- “Windows Phone” application
  - Single page general purpose starter project
- “DataBound” application
  - Good example of a master-detail app that uses databinding
- Class Library
  - Project which has no UI intended to contain reusable classes
- “Pivot” application
  - User can “pivot” between different screens by flicking left and right
- “Panorama” application
  - A single panoramic background with pages of controls that the user can pan between



# Application Types



- The three UI project application types provide quite different user experiences
- Select the one that you feel is the most appropriate

**NOKIA****Microsoft**

# ApplicationBar



# Application Chrome

## System Tray and Application Bar

### System Tray

System owned indicator area that displays system-level status information

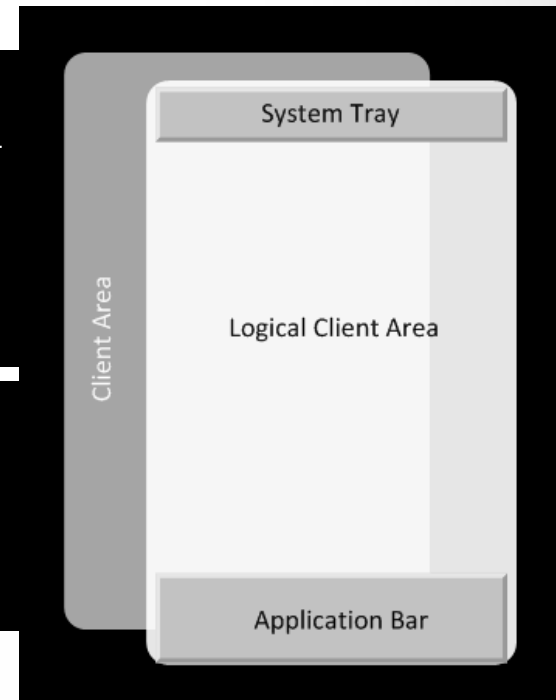
Apps can show/hide

```
Microsoft.Phone.Shell.SystemTray.IsVisible = false;
```

### Application Bar

Area where applications can display buttons for the most common tasks

Can display pop-up menu for less common tasks



# Application Bar

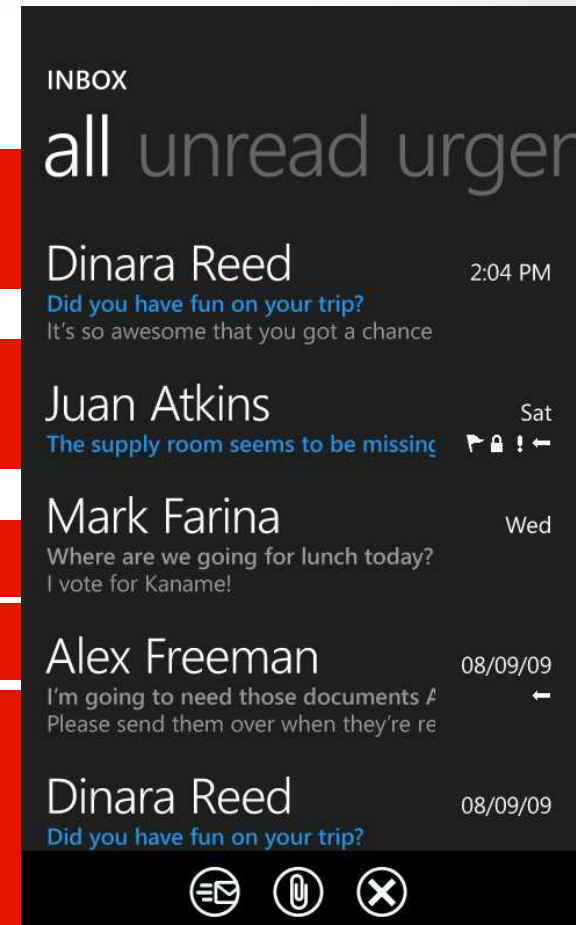
Use the **ApplicationBar** instead of creating your own menu system

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

Swipe up the bar to bring up the menu

Use white foreground on transparent background for icons  
System will colorize button according to users selected theme



NOKIA

Microsoft

# Application Bar in Xaml

```
<phone:PhoneApplicationPage
  x:Class="MyApp.MainPage"
  ... >

<phone:PhoneApplicationPage.ApplicationBar>
  <shell:ApplicationBar x:Name="AppBar" IsMenuEnabled="True">
    <shell:ApplicationBar.Buttons>
      <shell:ApplicationBarIconButton x:Name="NewContactButton"
        IconUri="Images/appbar.new.rest.png" Text="New"
        Click="NewContactButton_Click"/>
      <shell:ApplicationBarIconButton x:Name="SearchButton"
        IconUri="Images/appbar.feature.search.rest.png"
        Text="Find" Click="SearchButton_Click"/>
    </shell:ApplicationBar.Buttons>
    <shell:ApplicationBar.MenuItems>
      <shell:ApplicationBarMenuItem x:Name="GenerateMenuItem"
        Text="Generate Data" Click="GenerateMenuItem_Click" />
      <shell:ApplicationBarMenuItem x:Name="ClearMenuItem"
        Text="Clear Data" Click="ClearMenuItem_Click" />
    </shell:ApplicationBar.MenuItems>
  </shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar>
```

# App Bar & Landscape

`ApplicationBar` paints on side of screen in landscape

Has built-in animation when page switches orientation

**NOKIA****Microsoft**

# Application Bar Opacity

If Application Bar opacity is less than 1, displayed page will be the size of the screen  
Application Bar overlays screen content

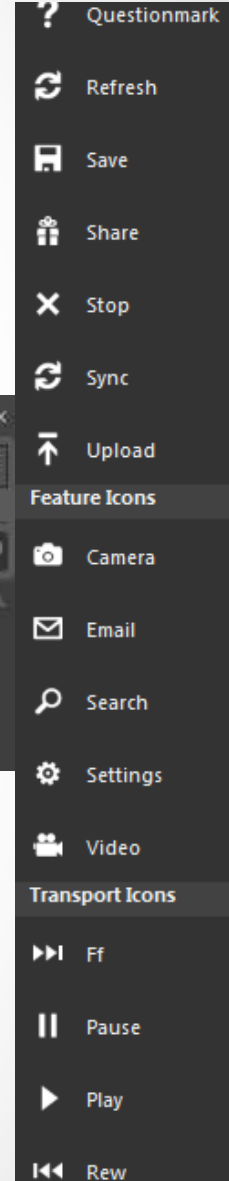
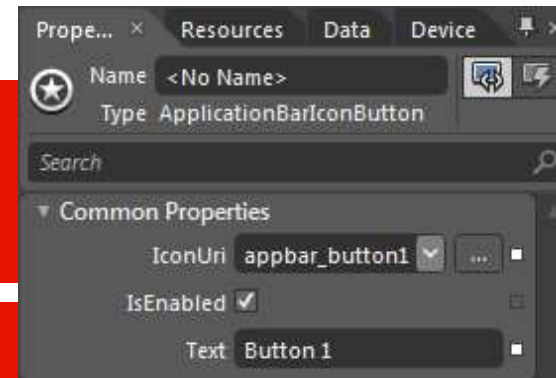
If Opacity is 1, displayed page is resized to the area of the screen not covered by the Application Bar



# AppBar Design in Expression Blend

**Expression Blend** is the best tool for selecting from the available standard icons

Drop-down IconUri select box in AppBarIcon Properties shows all standard icons



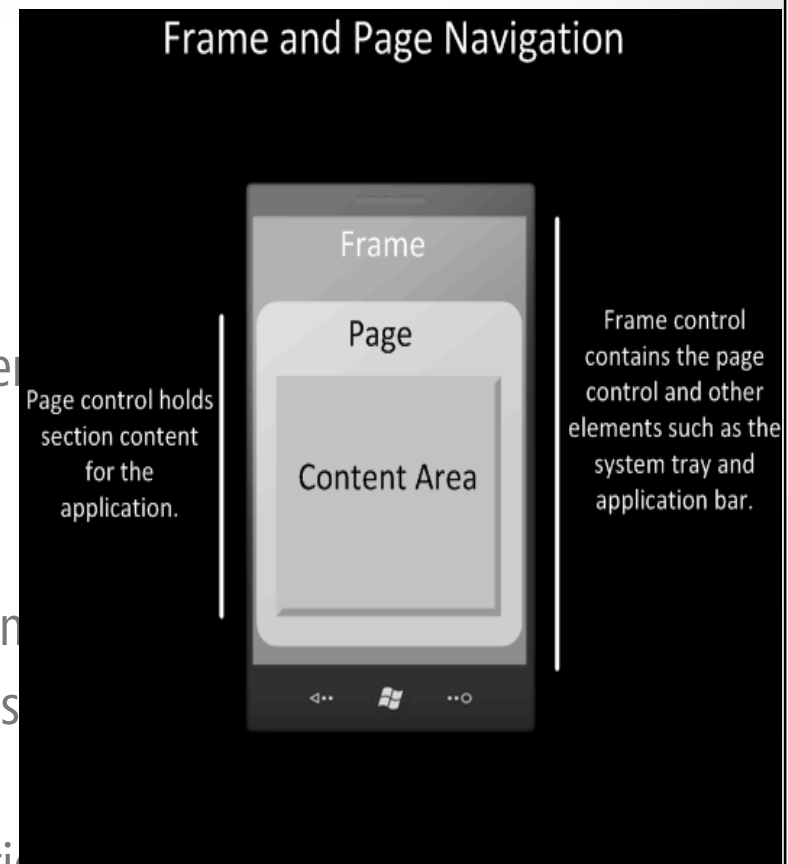
NOKIA

Microsoft®

# Page Navigation

# Frame and Page

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

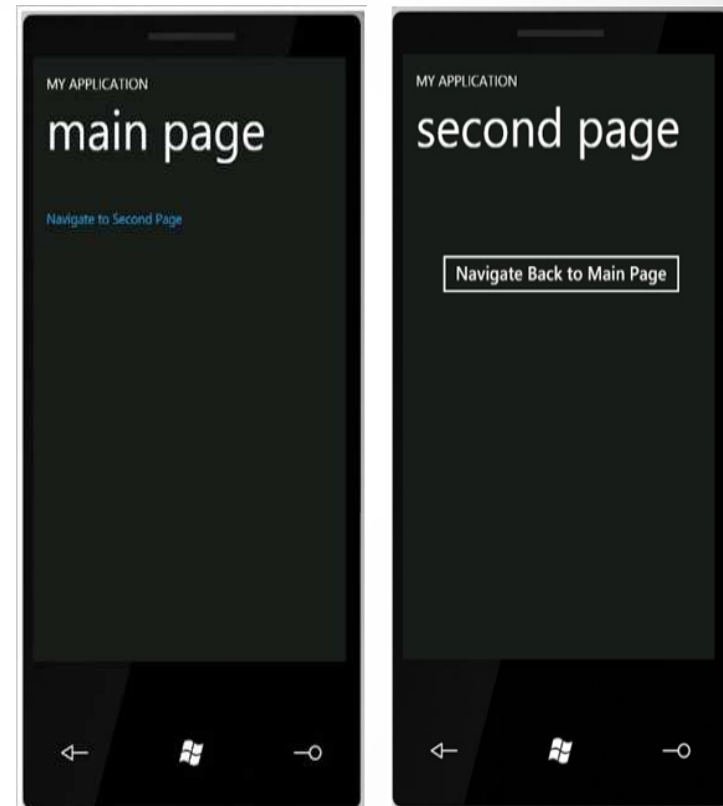




# Page Navigation

- 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, RoutedEventArgs e)  
{  
    NavigationService.Navigate(  
        new Uri("/SecondPage.xaml",  
            UriKind.RelativeOrAbsolute)  
    );  
}
```

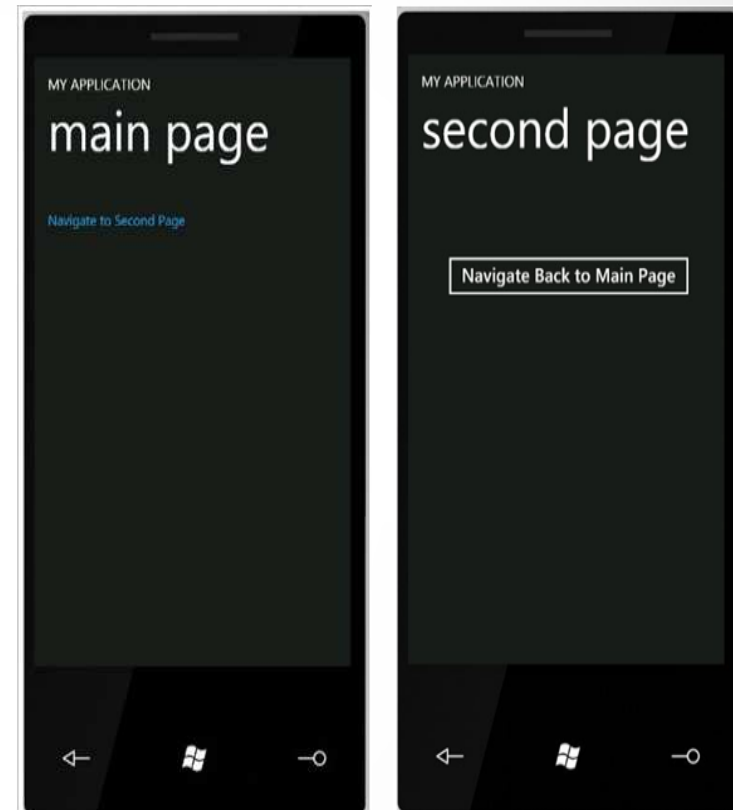
**NOKIA****Microsoft**

# Navigating Back

- Application can provide controls to navigate 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 required – built-in behavior

**NOKIA****Microsoft**

# Overriding the Back Key

- May need to override Back hardware key if 'back to previous page' is not logical behaviour
  - For example, when displaying a popup panel
  - User would expect Back key to close the panel, not the page

**NOKIA****Microsoft**

# Overriding the Back Key

```
<phone:PhoneApplicationPage
  x:Class="WindowsPhoneApplication4.Detail"
  ...
  BackKeyPress="PhoneApplicationPage_BackKeyPress"
  shell:SystemTray.IsVisible="True">
```

In code:

```
private void PhoneApplicationPage_BackKeyPress(object sender,
    System.ComponentModel.CancelEventArgs e)
{
    e.Cancel = true;    // Tell system we've handled it
}
```

# Passing Data Between Pages

- Can pass string data between pages using query strings

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

- On destination page

```
protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
{
    base.OnNavigatedTo(e);
    string msg = "";
    if (NavigationContext.QueryString.TryGetValue("msg", out msg))
        textBlock1.Text = msg;
}
```

# Passing Objects Between Pages

- Often, you will pass a data object from one page to another
  - E.g., user selects an item in a list and navigates to a Details page
- One solution is to store your ViewModel (that is, data) in your App class
  - Global to whole application
- Pass the selected item index in query string

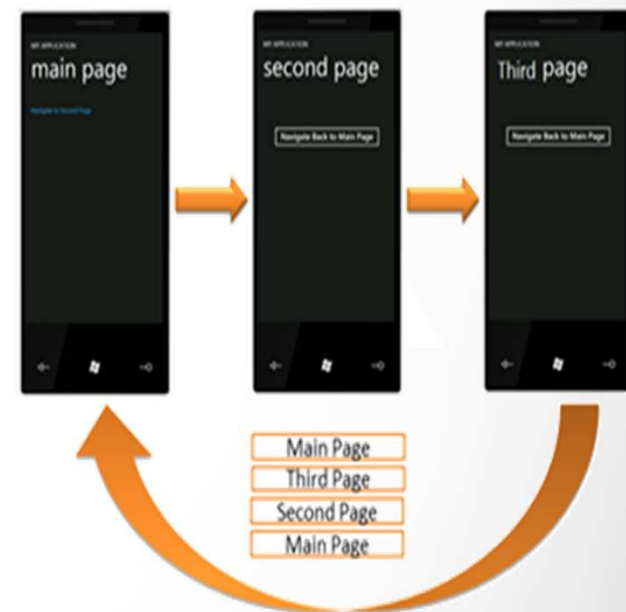
```
NavigationService.Navigate(  
    new Uri("/SecondPage.xaml?Param=" +  
    value,  
    UriKind.RelativeOrAbsolute))
```

**NOKIA****Microsoft**

# Handling Non Linear Navigation

- Design your app navigation strategy carefully!
- If you navigate from 'third page' to 'main page' and your user then presses the Back key, what happens?
  - User expects app to exit
  - App actually navigates back to Third Page
- Solution for Windows Phone 7.0 was complex code to handle back navigation correctly, or the Non-Linear Navigation Recipe library from AppHub
- Windows Phone 7.5 offers new API:
  - `NavigationService.RemovebackEntry()`

## NonLinear Navigation (A Loop)



# NavigationService.RemoveBackEntry()

- When 'Third Page' navigates back to MainPage, put a marker in the query string:

```
NavigationService.Navigate(new Uri("/MainPage.xaml?homeFromThird=true",  
UriKind.Relative));
```

- In OnNavigatedTo() in MainPage, look for the marker and if present, remove the 'Third Page' and 'SecondPage' from the navigation history stack:

```
protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)  
{  
    if (e.NavigationMode == System.Windows.Navigation.NavigationMode.New &&  
        NavigationContext.QueryString.ContainsKey( "homeFromThird" ))  
    {  
        NavigationService.RemoveBackEntry(); // Remove ThirdPage  
        NavigationService.RemoveBackEntry(); // Remove SecondPage  
    }  
}
```





Windows Phone

# Demo

## AppBar, Page Navigation and Pivot Control

25

# Data Binding

# Data Binding

- Simplest way to program UI controls is to write your own “glue” to get and set properties of controls
  - e.g. `textBox1.Text = "Hello, world";`
- In complex applications, such code quickly becomes unwieldy and error prone.
- Use Silverlight data binding to link your UI to a class in your application that contains your application data
  - A data class that is a source for data binding is called a **ViewModel**
- UI controls can get their display values automatically from properties of the viewmodel class
  - Changing the property, updates the display
  - User input can automatically update the bound property of the viewmodel class



# Data Binding in XAML

```
<TextBlock x:Name="ContentText"  
    Text="{Binding LineThree, Mode=OneWay}" TextWrapping="Wrap"  
    Style="{StaticResource PhoneTextTitle3Style}"/>
```

- Properties of controls can be bound to a public property of a data object
  - In the example above, the **Text** property of the TextBlock is bound to the **LineThree** property of some data source
- Define the data source by setting:
  - The **DataContext** property of any containing FrameworkElement-derived class (a containing control, the page, or the frame), **or**
  - The **ItemsSource** property of a List control



# Data Binding Modes

```
<TextBlock x:Name="ContentText"  
    Text="{Binding LineThree, Mode=OneWay}"  
    TextWrapping="Wrap"  
    Style="{StaticResource PhoneTextTitle3Style}"/>
```

- The Mode property determines how changes are synchronized between the target control and data source
  - OneTime – Control property is set once to the data value and any subsequent changes are ignored
  - OneWay – Changes in the data object are synchronized to the control property, but changes in the control are not synchronized back to the data object
  - TwoWay – Changes in the data object are synchronized to the control property and vice-versa



# INotifyPropertyChanged

- Data objects that take part in OneWay or TwoWay binding must implement INotifyPropertyChanged

```
public class ItemViewModel : INotifyPropertyChanged
{
    private string lineOne;
    public string LineOne
    {
        get { return lineOne; }
        set { if (value != lineOne) {
                lineOne = value;
                NotifyPropertyChanged("LineOne"); } }
    }
    public event PropertyChangedEventHandler PropertyChanged;
    private void NotifyPropertyChanged(String propertyName)
    {
        if (null != PropertyChanged) PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}
```



# Binding to Lists

```
<ListBox x:Name="MainListBox" ItemsSource="{Binding Items}">  
...  
</ListBox>
```

- List controls can bind to collections of items
- For one way or two way databinding to work, this must be an ObservableCollection
- Items inside an ObservableCollection need to implement INotifyPropertyChanged
  - Provides event notifications to Silverlight data binding whenever a data object property in the collection changes, or an item is added to or removed from the collection



# Observable Collections

```
public class MainViewModel
{
    public MainViewModel()
    {
        // Insert some test data into the collection
        Items = new ObservableCollection<ItemViewModel>() {
            new ItemViewModel() { LineOne = "runtime one",
                                   LineTwo = "Maecenas praesent",
                                   LineThree = "Facilisi faucibus ", },
            new ItemViewModel() { LineOne = "runtime two",
                                   LineTwo = "Dictumst eleifend",
                                   LineThree = "Suscipit torquent ", },
        };
    }

    public ObservableCollection<ItemViewModel> Items { get; private set; }
}
```



# MVVM

- MVVM stands for Model – View – ViewModel
- MVVM is an architectural pattern that employs Databinding and strict separation of concerns
  - Model – a class or classes that exposes the data of your application, either fetched from local data storage or externally such as a web service
  - ViewModel – a class or classes that has properties and methods that can be used to databind to a View
  - View – a class or classes that implement the presentation functionality of your application, displaying data and accepting user input. A View should contain no application logic and is bound to a ViewModel class
- See:
  - <http://channel9.msdn.com/blogs/kreekman/techdays-2010-understanding-the-model-view-viewmodel-pattern>
  - <http://galasoft.ch/mvvm/>

**NOKIA****Microsoft®**



# Demo

# Data Binding



34

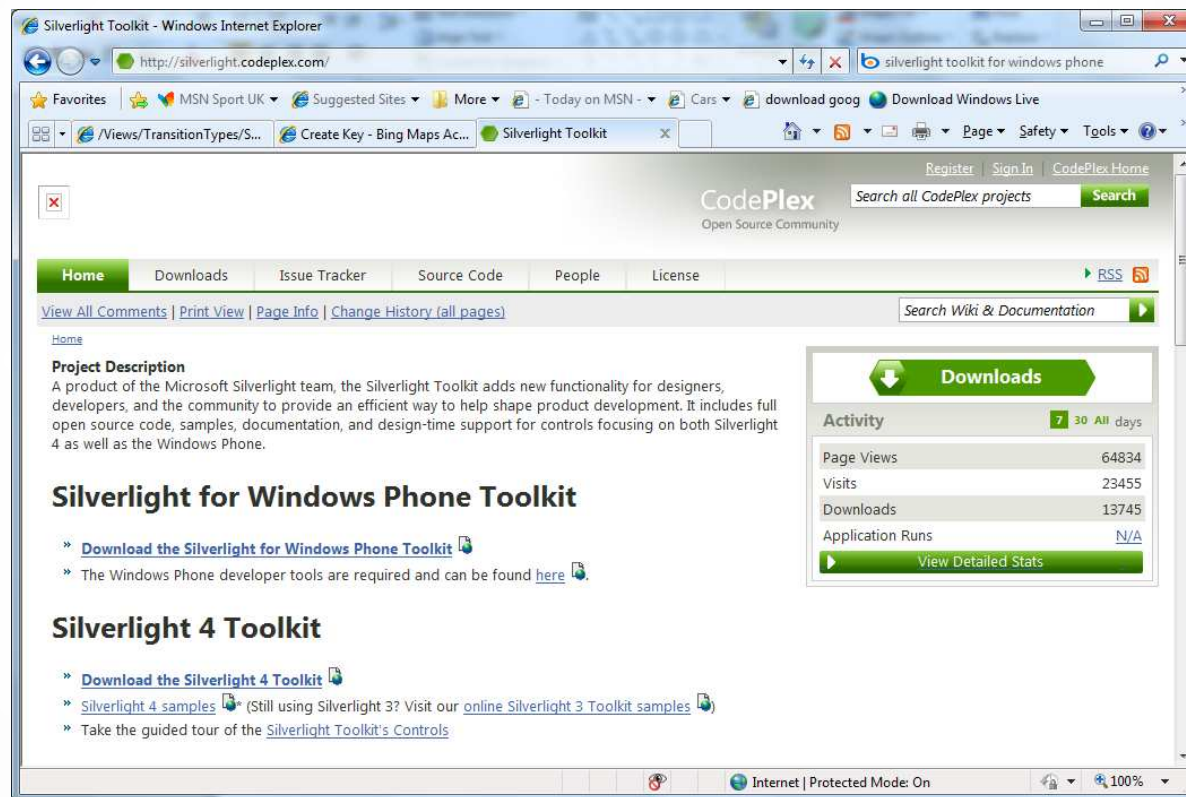
# The Silverlight Toolkit

# Silverlight Toolkit for Windows Phone

- A product of the Microsoft Silverlight team
- The Silverlight Toolkit adds new functionality 'out of band' from the official product control set
- Includes full open source code, samples, documentation, and design-time support for controls for Windows Phone
- Refresh every 3 months or so
  - Bug fixes
  - New controls

# How to Get the Silverlight Toolkit

- <http://silverlight.codeplex.com>



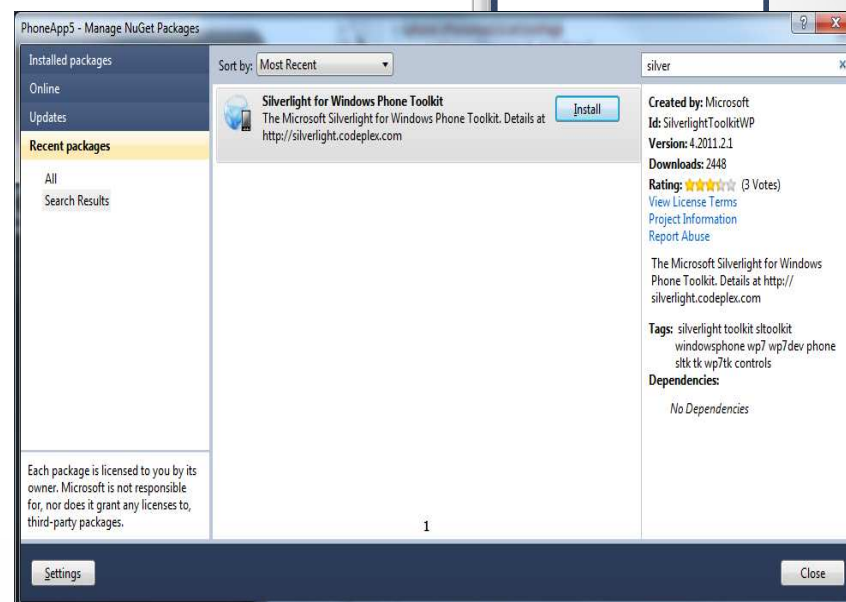
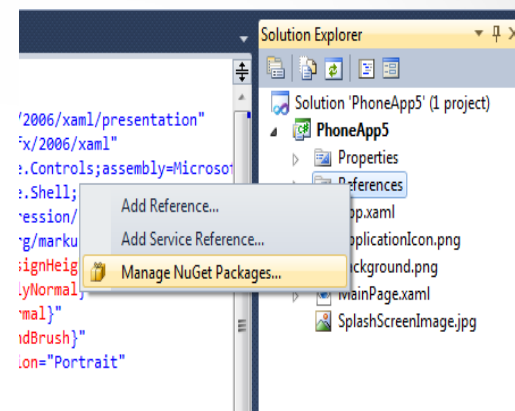
NOKIA

3Microsoft®

# NuGet



- Package management system for .NET
- Simplifies incorporating 3rd party libraries
- Developer focused
- Free, open source
- Install NuGet using the Visual Studio Extension Manager
- Use NuGet to add libraries such as the Silverlight Toolkit to projects



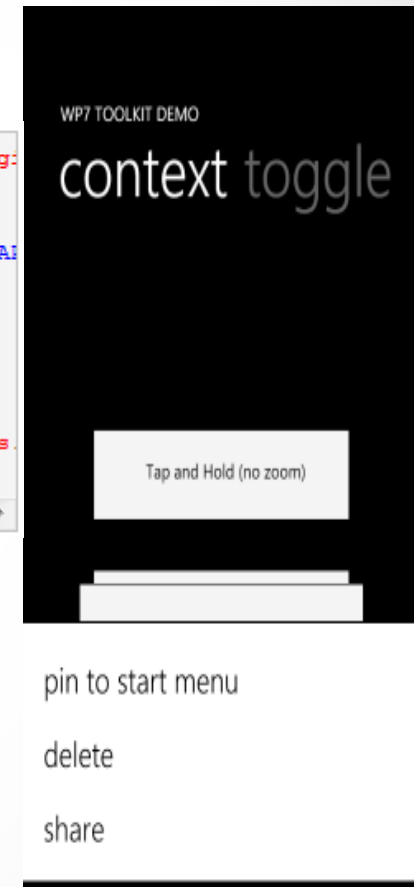
NOKIA

Microsoft

# Controls in the Silverlight Toolkit

# ContextMenu

```
1: <Canvas HorizontalAlignment="Left" VerticalAlignment="Top" Width="345" Height="91" Margin="0,0,0,0">
2:     <toolkit:ContextMenuService.ContextMenu>
3:         <toolkit:ContextMenu>
4:             <toolkit:MenuItem Header="pin to start menu" Click="OnMenuClicked" Tag="START" />
5:             <toolkit:MenuItem Header="delete" Click="OnMenuClicked" Tag="DELETE" />
6:             <toolkit:MenuItem Header="share" Click="OnMenuClicked" Tag="SHARE" />
7:         </toolkit:ContextMenu>
8:     </toolkit:ContextMenuService.ContextMenu>
9:     <Rectangle Fill="#FFF4F4F5" Height="91" Stroke="Black" Width="345"/>
10:    <TextBlock TextWrapping="Wrap" Text="Tap and Hold (zoom)" Foreground="Black" Canvas.Left="10" Canvas.Top="10" />
11: </Canvas>
```





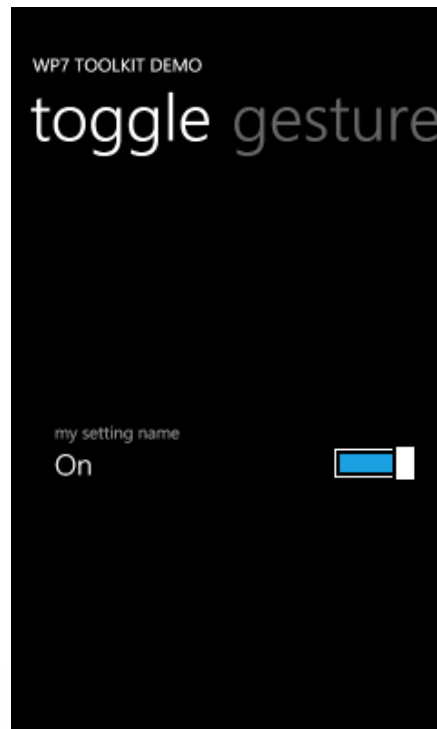
# DatePicker and TimePicker

```
1: <toolkit:DatePicker />
```



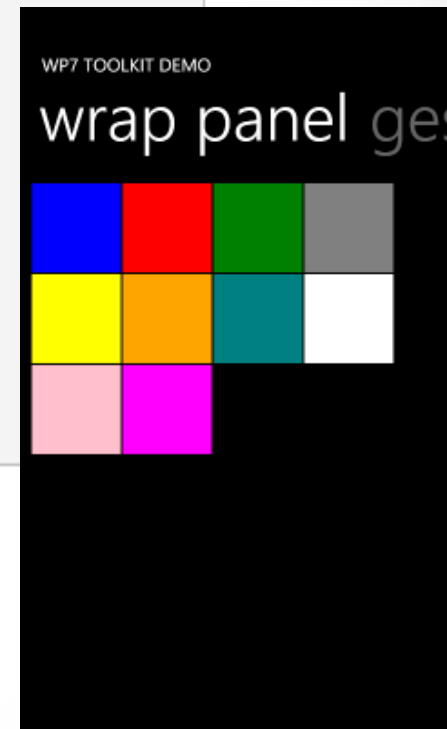
# ToggleSwitch

```
1: <toolkit:ToggleSwitch Header="my setting name" Height="118" Margin="0,0,-24,-34" Width="
```

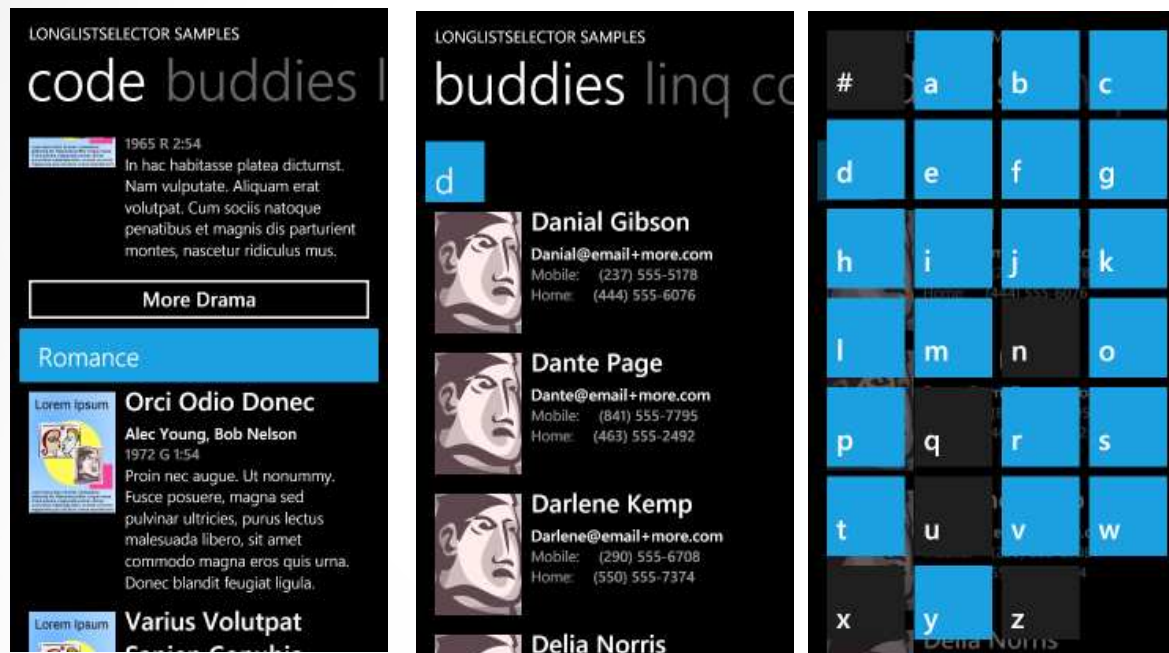


# WrapPanel

```
1: <toolkit:WrapPanel Orientation="Horizontal">
2:     <Rectangle Fill="Blue" Width="100" Height="100" Stroke="Black" />
3:     <Rectangle Fill="Red" Width="100" Height="100" Stroke="Black" />
4:     <Rectangle Fill="Green" Width="100" Height="100" Stroke="Black" />
5:     <Rectangle Fill="Gray" Width="100" Height="100" Stroke="Black" />
6:     <Rectangle Fill="Yellow" Width="100" Height="100" Stroke="Black" />
7:     <Rectangle Fill="Orange" Width="100" Height="100" Stroke="Black" />
8:     <Rectangle Fill="Teal" Width="100" Height="100" Stroke="Black" />
9:     <Rectangle Fill="White" Width="100" Height="100" Stroke="Black" />
10:    <Rectangle Fill="Pink" Width="100" Height="100" Stroke="Black" />
11:    <Rectangle Fill="Magenta" Width="100" Height="100" Stroke="Black" />
12: </toolkit:WrapPanel>
```



# LongListSelector



## ListBox++

- Flat lists
- Grouped lists – with headers
- Jump List

Supports full UI and data virtualization

NOKIA

Microsoft®

# ListPicker



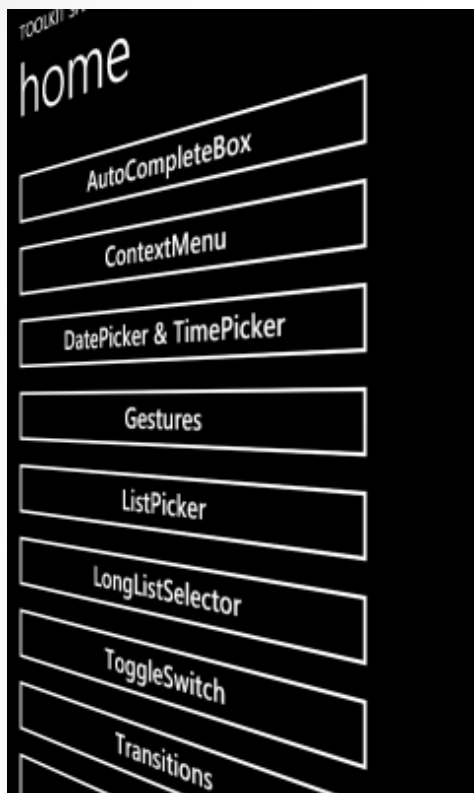
## WP7 ComboBox

- Dropdown list for small number of items
- Full screen selector for longer lists

**NOKIA****Microsoft**

# Page Transitions and TiltEffect

# Page Transitions



- Easy way to add page transitions to your app similar to those in the built-in apps
- Different transitions are included:
  - Roll, Swivel, Rotate, Slide and Turnstile
- Start by using the **TransitionFrame** control from the Silverlight Toolkit instead of the default PhoneApplicationFrame
  - Set in InitializePhoneApplication() method in App.xaml.cs:

```
//RootFrame = new PhoneApplicationFrame();  
RootFrame = new TransitionFrame();
```

# Enabling Transitions on a Page

- Declare namespace for Silverlight Toolkit assembly

```
<phone:PhoneApplicationPage  
  xmlns:toolkit="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls.Toolkit"
```

- Under <phone:PhoneApplicationPage> root element, add transition you want

```
  <toolkit:TransitionService.NavigationInTransition>  
    <toolkit:NavigationInTransition>  
      <toolkit:NavigationInTransition.Backward>  
        <toolkit:TurnstileTransition Mode="BackwardIn"/>  
      </toolkit:NavigationInTransition.Backward>  
      <toolkit:NavigationInTransition.Forward>  
        <toolkit:TurnstileTransition Mode="ForwardIn"/>  
      </toolkit:NavigationInTransition.Forward>  
    </toolkit:NavigationInTransition>  
  </toolkit:TransitionService.NavigationInTransition>  
  <toolkit:TransitionService.NavigationOutTransition>  
    <toolkit:NavigationOutTransition>  
      <toolkit:NavigationOutTransition.Backward>  
        <toolkit:TurnstileTransition Mode="BackwardOut"/>  
      </toolkit:NavigationOutTransition.Backward>  
      <toolkit:NavigationOutTransition.Forward>  
        <toolkit:TurnstileTransition Mode="ForwardOut"/>  
      </toolkit:NavigationOutTransition.Forward>  
    </toolkit:NavigationOutTransition>  
  </toolkit:TransitionService.NavigationOutTransition>
```

**NOKIA****Microsoft**



# TiltEffect

- Add additional visual feedback for control interaction
- Instead of simple states such as Pressed or Unpressed, controls with TiltEffect enabled provide motion during manipulation
  - For example, Button has a subtle 3D effect and appears to move into the page when pressed and bounce back again when released

- Easy to enable TiltEffect for all controls on a page

```
<phone:PhoneApplicationPage  
    xmlns:toolkit="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls.Toolkit"  
    toolkit:TiltEffect.IsTiltEnabled="True">
```

- Also can apply to individual controls

```
<ToggleButton Content="Button" toolkit:TiltEffect.IsTiltEnabled="True"/>
```



Windows Phone

# Demo

## Page Transitions and TiltEffect

50

# Gesture Support

# Silverlight Toolkit GestureService

- Library that allows you to attach gesture recognition to UIElements for common gestures:
  - Flick, Pinch

```
1: <Image x:Name="GesturedImage" Source="dividbyzero.jpg" HorizontalAlignment="Center" VerticalAlignment="Top"
2:     RenderTransformOrigin="0.5,0.5">
3:     <Image.RenderTransform>
4:         <ScaleTransform x:Name="ImageScaling" ScaleX="1" ScaleY="1" />
5:     </Image.RenderTransform>
6:     <toolkit:GestureService.GestureListener>
7:         <toolkit:GestureListener PinchDelta="OnPinchDelta" />
8:     </toolkit:GestureService.GestureListener>
9: </Image>
```

```
1: private void OnPinchDelta(object sender, PinchGestureEventArgs e)
2: {
3:     ImageScaling.ScaleX = e.DistanceRatio;
4:     ImageScaling.ScaleY = e.DistanceRatio;
5: }
```

# Touch Events in Windows Phone OS 7.1

- In 7.1, UIElement gets support for touch events that previously were only supported by the Silverlight Toolkit Gesture Service:
  - Tap
  - DoubleTap
  - Hold
- These augment the manipulation events already in 7.0:
  - ManipulationStarted
  - ManipulationDelta
  - ManipulationCompleted



## Tip: Capturing Manipulation Events

- Common problem: "Why are my manipulation events not firing?"
  - Usually, this is because the element where you have hooked the manipulation events does not have a background,
  - In this case, manipulation events bubble up right through your element to whatever lies behind

```
<Border ManipulationStarted="BorderManipulationStarted"  
        ManipulationDelta="BorderManipulationDelta"  
        Background="Transparent">  
    <!--Stuff-->  
</Border>
```



Windows Phone

# Demo

## Silverlight Toolkit Gesture Service

55

# Review

- Navigation to pages is performed on the basis of uri (Uniform Resource Indicator) values
- The back button normally navigates back to the source page, but this can be overridden
- The uri can contain query strings
- The Silverlight Toolkit provides additional controls and libraries
- Data binding allows a display to automatically reflect the state of a data object







# Windows® Phone

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

**Microsoft®**