



# Microsoft

MCSD: Windows Store Style  
Apps Using C# Certification

70-484: Essentials of  
Developing Windows Store  
style Apps using C#  
Courseware

Version 1.0

[www.firebrandtraining.com](http://www.firebrandtraining.com)

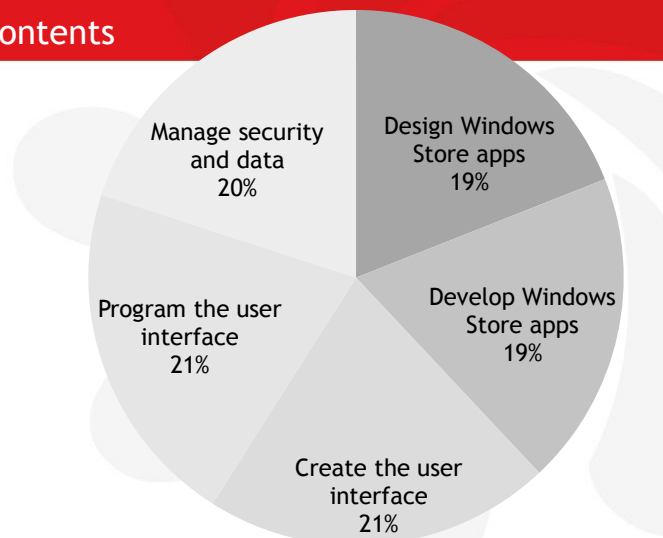
## Module 1

### Overview of the Windows 8 Platform and Windows Store Apps



## Course and Exam Contents

- 🔗 55 questions
- 🔗 130 minutes
- 🔗 3 case studies



Exam 70-484 – Essentials of Developing Windows Store Apps using C#  
<http://www.vic.ms/microsoft/windows-8/exam-70-484-essentials-of-developing-windows-store-apps-using-c/>

MSDN study links for exam 70-484: Essentials of Developing Windows Metro style Apps using C#  
<http://www.jayway.com/2012/10/25/msdn-study-links-for-exam-70-484-essentials-of-developing-windows-metro-style-apps-using-c/>



## Overview of the Windows 8 Platform and Windows Store Apps Contents

1.3

### Exam Topic: Design the UI layout and structure

- ❑ Evaluate the conceptual design and decide how the UI will be composed
- ❑ Design for the inheritance and re-use of visual elements (e.g., styles, resources)
- ❑ Design for accessibility
- ❑ Decide when custom controls are needed

### Exam Topic: Design for separation of concerns (SOC)

- ❑ Plan the logical layers of your solution to meet app requirements
- ❑ Design loosely coupled layers
- ❑ Incorporate WinMD components

Commanding design for Windows Store apps  
<http://msdn.microsoft.com/library/windows/apps/hh761499.aspx>

Plan your app  
<http://msdn.microsoft.com/library/windows/apps/hh465427.aspx>

Design principles  
<http://msdn.microsoft.com/library/windows/apps/hh779072.aspx>



## Structure of the Case Studies

1.4

### ✿Background

- e.g. you're building a Flickr-clone

### ✿Application structure

- e.g. your app calls an HTTP service, uses MVVM pattern

### ✿Business and technical requirements

- e.g. must work in low bandwidth
- e.g. must show toast notifications when a picture is favoured

### ✿MainWindow.xaml, MainWindow.xaml.cs

- Line numbers for reference, some missing code

### ✿DataModel.cs



## Layers High-Level

1.5

### 🌀 Presentation

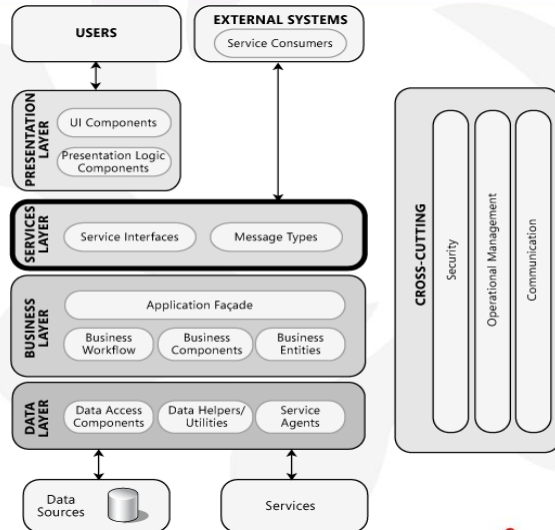
- Typically XAML for the user interface

### 🌀 Business

- Code logic and rules that define how an action should be performed

### 🌀 Data

- Code to retrieve and update the data store



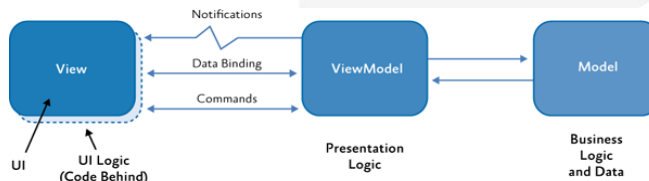
Chapter 5: Layered Application Guidelines  
<http://msdn.microsoft.com/en-us/library/ee658109.aspx>



## Layers Model-View-ViewModel (MVVM)

1.6

Layer	Component
Model	Encapsulates business logic and data; enforces business rules, retrieval and management of data
View	Encapsulates the UI and any UI logic; interacts with the view model through data binding, commands, and change notification events
ViewModel	Encapsulates presentation logic and state; queries, observes, and coordinates updates to the model, converting, validating, and aggregating data as necessary for display, retrieval and management of data bound to user interface; code that could affect the behavior of the user interface



Implementing the MVVM Pattern  
[http://msdn.microsoft.com/en-us/library/gg405484\(v=pandp.40\).aspx](http://msdn.microsoft.com/en-us/library/gg405484(v=pandp.40).aspx)



Layer	Description
View	XAML that defines the user interface, typically a Page whose DataContext is set to an instance of a ViewModel so that controls can use data binding; the code-behind for a View should not retrieve data itself to keep it as cross-platform as possible
ViewModel	A class that defines all the data required by a View, without any logic for populating it, to keep it as cross-platform as possible
Helper	A class that is used to populate a ViewModel, which can be specific to a platform such as Windows Phone, Windows Runtime, WPF, Silverlight, and so on



✦ If your Page's DataContext is a ViewModel with properties of type ICommand

```
public class MyContext {  
    public static RoutedCommand Open = new RoutedCommand();
```

```
    private void ExecutedCustomCommand(object sender,  
        ExecutedRoutedEventArgs e) {
```

```
    private void CanExecuteCustomCommand(object sender,  
        CanExecuteRoutedEventArgs e) {
```

```
        var customCommandBinding = new CommandBinding(  
            Open, ExecutedCustomCommand, CanExecuteCustomCommand);  
        this.CommandBindings.Add(customCommandBinding);
```

```
<Button Command="{Binding Open}">Open File</Button>
```

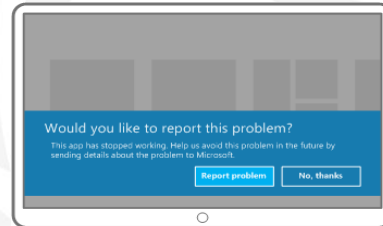
ButtonBase.Command property  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.primitives.buttonbase.command.aspx>



## Laying out your UI

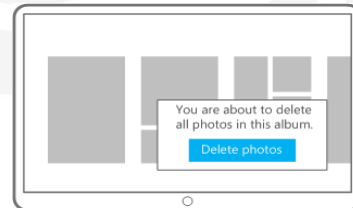
### Message dialogs

- They dim the app window and demand a user response before continuing
- Use only when you intend to stop the user and to demand response



### Flyouts

- Flyouts show temporary, dismissable UI related to what the user is currently doing
- Use flyouts to ask the user to confirm an action, to show a drop-down menu from a button the app bar, or to show more details about an item



Laying out your UI (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh465330.aspx>



## Reference

### Developing apps (C#/C++/VB)

- › App features, start to finish
- › Creating a UI
- › Defining app resources
- › Responding to user interaction
- › Working with data and files
- › Connecting to networks and web services
- › Managing user info
- › Launching, resuming and multitasking
- › Adding multimedia
- › Integrating devices, printers, and sensors
- › Adding trials and in-app purchases
- › Globalizing your app

## Developing Windows Store apps (C#/C++/VB)

165 out of 268 rated this helpful - [Rate this topic](#)



Learn how to create Windows Store apps using C#, C++, or Visual Basic. From the new Start screen to new UI layout and controls, the presentation of your app is about showing off what it's great at.

### In this article

- [Before you begin](#)
- [App features, start to finish](#)
- [Steps to developing Windows Store apps](#)
- [Related topics](#)

Use this link as your definitive reference for the course

Developing Windows Store apps (C#/C++/VB)  
<http://msdn.microsoft.com/en-us/library/windows/apps/br229566.aspx>



## Module 2

### Creating User Interfaces Using XAML



## XAML

### Why XAML?

#### ✦ Advantages of XAML and Windows Store Apps

- Powerful data binding and visualization, media support, 2D and 3D vector graphics, animation, flow and fixed documents
- Also used in Windows Phone and Windows Desktop applications (Windows Presentation Foundation)

#### ✦ XAML designer in Visual Studio

- Better XAML IntelliSense, event handling and code writing

#### ✦ XAML designer in Expression Blend

- Better visual property, timeline and animation editing



🌀 1990s: Visual Basic, C

- Contains: string (optionally also an integer)
- Displays: string



🌀 Early 2000s: .NET 1.0+

- Contains: object
- Displays: string (calls *ToString* method)

🌀 2006 and later: .NET 3.0+

- Contains: object
- Displays: object

```
<ListBoxItem>  
<Image ... >  
<Hyperlink ... >  
<RichTextBox ... >  
</ListBoxItem>
```



🌀 XAML is declarative code

- Easier for code generators and programmers to read and write
- Simply instantiates and controls .NET classes

🌀 XAML is an *alternative* to procedural language such as C# and Visual Basic, but is not *required*

- For example, this XAML could be written using C#

```
Button b1 = new Button();  
b1.Content = "OK";  
b1.Background = new SolidColorBrush(Colors.LightBlue);  
DockPanel.SetDock(b1, Dock.Top);
```

```
<Button Name="b1"  
  DockPanel.Dock="Top"> OK  
  <Button.Background>  
    LightBlue  
  </Button.Background>  
</Button>
```





## Namespaces and Instantiating Objects

## ✿ Default defined namespaces

```
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
```

## ✿ Importing namespaces

```
xmlns:sys="clr-namespace:System:assembly=mscorlib"
xmlns:aw="clr-namespace:Wpf.Examples"
```

## ✿ Instantiating objects

```
<sys:Double>98.1</sys:Double>
<aw:Product Name="Bike" ListPrice="12.34" Color="Red" />
```

XAML Namespaces and Namespace Mapping for WPF XAML  
<http://msdn.microsoft.com/en-us/library/ms747086.aspx>



## Setting Properties in Markup

## ✿ Two ways of setting a property in XAML

- As an attribute or as child element

```
<Button Background="LightBlue">
  ...
</Button>
```

```
<Button>
  <Button.Background>
    LightBlue
  </Button.Background>
  ...
</Button>
```

## ✿ Setting default properties

```
<Button Content="Click Me" />
```

```
<Button>Click Me</Button>
```

## ✿ Setting attached properties

- Objects can gain extra abilities by being children of a parent that defines attached properties

```
<Canvas>
  <Button Canvas.Top="20" Panel.ZIndex="1" />
```



## Setting Attached Properties in Code

- ✦ In markup, use the class name that defines the attached property

```
<Canvas>  
  <Button Canvas.Top="20" Panel.ZIndex="1" ... />
```

- ✦ In code, use static methods on the class that defines the attached property

```
Canvas.SetTop(Button1, 20);  
Panel.SetZIndex(Button1, 1);
```

- ✦ Can also read attached properties

```
double top = Canvas.GetTop(Button1);  
int zIndex = Panel.GetZIndex(Button1);
```



## Naming Objects

- ✦ x:Name or Name (but cannot use both)

- After x:Name is applied to a framework's backing programming model, the name is equivalent to the variable that holds an object reference or an instance as returned by a constructor

- ✦ x:Key

- Used for items that are being added as values to a dictionary, most often for styles and other resources that are being added to a ResourceDictionary
- There is actually no corresponding property on the object or even an attached dependency property being set, it is simply used by the XAML processor to know what key to use when calling Dictionary.Add

```
x:Name Directive  
http://msdn.microsoft.com/en-us/library/ms752290.aspx
```



## Dependency Properties and Routed Events

✦ CLR classes have simple properties and events

```
public string FirstName { get; set; }  
public event EventHandler Clicked;
```

✦ XAML has *dependency* properties and *routed* events

- Support data binding, styles, resources, animation, event tunnelling and bubbling, and other special features

```
public static readonly DependencyProperty Height =  
    DependencyProperty.Register("Height", typeof(int), ...  
public static readonly RoutedEvent TapEvent =  
   EventManager.RegisterRoutedEvent("Tap",  
    RoutingStrategy.Bubble, ...);
```

Dependency Properties Overview  
<http://msdn.microsoft.com/en-us/library/ms752914.aspx>

Routed Events Overview  
<http://msdn.microsoft.com/en-us/library/ms742806.aspx>



## Separation of Control Behaviour and Appearance

✦ XAML separates the behaviour of a control from its appearance

- Every control has a default appearance but this can be replaced

✦ For example, a button is something that can be clicked to trigger an action

- Although the default look may be a 3D silver-grey rectangle, a button could look like anything, may be animated, and so on
- Control templates allow this



# Routed Events Tunneling and Bubbling 2.11

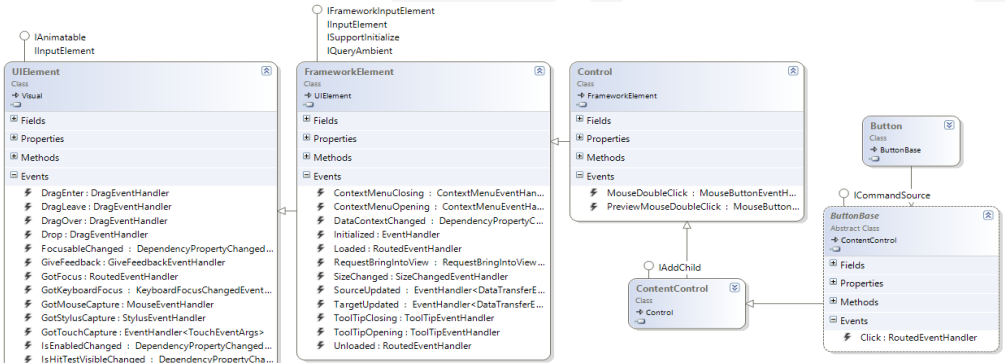
- ✦ How to tell the difference
  - By convention, Preview is a prefix for naming events that are registered to use tunnelling strategy
- ✦ Three supported strategies
  - Bubbling
  - Tunnelling
  - Direct

Understand Bubbling and Tunnelling in 5 minutes  
<http://www.wpffmentor.com/2008/11/understand-bubbling-and-tunnelling-in-5.html>



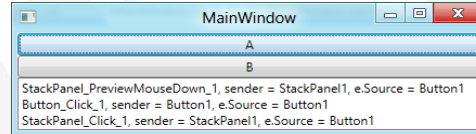
# Routed Events Inheritance Hierarchy 2.12

- UIElement.MouseDown, PreviewMouseDown, and so on
- Control.MouseDoubleClick, PreviewMouseDoubleClick
- ButtonBase.Click



Difference between *sender* and *e.Source*

- ✿ *sender* is the object that handled the event
- ✿ *e.Source* is the object that triggered the event



```
<StackPanel ButtonBase.Click="StackPanel1_Click"
PreviewMouseDown="StackPanel1_PreviewMouseDown" ...>
  <Button Content="A" Click="Button1_Click" ...>
  <Button Content="B" ...
```

```
Private Sub StackPanel1_Click( ... )
  ListBox1.Items.Add("StackPanel1_Click, sender = " &
  CType(sender, FrameworkElement).Name & ", e.Source = " &
  CType(e.Source, FrameworkElement).Name)
```

RoutedEventArgs.Source Property – <http://msdn.microsoft.com/en-us/library/system.windows.routedEventArgs.source.aspx>

*e.Handled*

- ✿ Set *e.Handled* to true to prevent routed events from tunnelling and bubbling to other event handlers
  - Setting *e.Handled* to true on a button's Click handler would prevent a parent panel from receiving the Click event...
  - ...unless the parent panel added its handler using code and passed true for the *handleEventToo* parameter

```
AddHandler(Button1.KeyDown, StackPanel1_KeyDown, true);
```

Marking Routed Events as Handled, and Class Handling  
<http://msdn.microsoft.com/en-us/library/ms747183>



## Module 3 Presenting Data



## Presenting Data Contents

### Exam Topic: Design and implement data presentation

- ❑ Choose and implement data controls to meet app requirements (e.g. ListView, GridView, and FlipView)
- ❑ Create data templates to meet app requirements

### Exam Topic: Apply the MVVM pattern to your app design

- ❑ Design and implement the appropriate data model to support business entities
- ❑ Design your viewmodel to support your view based on your model
- ❑ Develop a view to meet data-binding requirements
- ❑ Create view models using INotifyPropertyChanged, ObservableCollection, CollectionViewSource

### Exam Topic: Implement data binding

- ❑ Choose and implement data-bound controls
- ❑ Bind collections to items controls
- ❑ Implement the IValueConverter interface
- ❑ Create and set dependency properties
- ❑ Validate user input
- ❑ Enable filtering, grouping, and sorting data in the user interface



## Implementing INotifyPropertyChanged

✦ `INotifyPropertyChanged` is used to notify binding clients that a property value has changed

```
public class DemoCustomer : INotifyPropertyChanged {
    public event PropertyChangedEventHandler PropertyChanged;
    private void MyNotifyPropertyChanged(string name) {
        if (PropertyChanged != null) {
            PropertyChanged(this, new PropertyChangedEventArgs(name));
        }
    }
}
```

```
// in property setter
this.customerNameValue = value;
MyNotifyPropertyChanged("CustomerName");
```

✦ `CallerMemberName` simplifies code

```
private void MyNotifyPropertyChanged(
    [CallerMemberName] string name = "") {
```

```
MyNotifyPropertyChanged();
```

`INotifyPropertyChanged` Interface  
<http://msdn.microsoft.com/en-us/library/system.componentmodel.inotifypropertychanged.aspx>



## Data Binding with ObservableCollection<T>

✦ Represents a dynamic data collection that provides notifications when items get added, removed, or when the whole list is refreshed

- `ObservableCollection<T>` is specifically designed to support live updates in the user interface when data bound to a model so always use it in preference to any other collection

`ObservableCollection<T>` Class  
<http://msdn.microsoft.com/en-us/library/ms668604.aspx>



## ItemsControl.ItemTemplate

✦ Gets or sets the DataTemplate used to display each item

- If your ItemsControl is bound to a collection object and you do not provide a DataTemplate, the resulting UI of each item is a string representation of each object in the underlying collection

```
<ListBox ItemsSource="{Binding Source={StaticResource myToDoList}}">
  <ListBox.ItemTemplate>
    <DataTemplate>
      <StackPanel>
        <TextBlock Text="{Binding Path=TaskName}" />
        <TextBlock Text="{Binding Path=Description}" />
        <TextBlock Text="{Binding Path=Priority}" />
      </StackPanel>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```

```
<ListBox ItemsSource="{Binding Source={StaticResource myToDoList}}"
        ItemTemplate="{StaticResource myTemplate}" />
```

ItemsControl.ItemTemplate Property

<http://msdn.microsoft.com/en-us/library/system.windows.controls.itemscontrol.itemtemplate.aspx>



## IValueConverter

✦ Provides a way to apply custom logic to a binding

- If you want to associate a value converter with a binding, create a class that implements the IValueConverter interface and then implement the Convert and ConvertBack methods
- Converters can change data from one type to another, translate data based on cultural information, or modify other aspects of the presentation
- Both the Convert and ConvertBack methods have a culture parameter that indicates the cultural information

IValueConverter Interface

<http://msdn.microsoft.com/en-us/library/system.windows.data.ivalueconverter.aspx>





## IValueConverter Example

```
[ValueConversion(typeof(DateTime), typeof(String))]
public class DateConverter : IValueConverter {
    public object Convert(object value, Type targetType,
        object parameter, CultureInfo culture) {
        DateTime date = (DateTime)value;
        return date.ToShortDateString();
    }
    public object ConvertBack(object value, Type targetType,
        object parameter, CultureInfo culture) {
        string strValue = value as string;
        DateTime resultDateTime;
        if (DateTime.TryParse(strValue, out resultDateTime))
        {
            return resultDateTime;
        }
        return DependencyProperty.UnsetValue;
    }
}
```

```
<src:DateConverter x:key="dateConverter"/>
```

```
<TextBlock Name="StartDateDTKey"
    Text="{Binding Path=StartDate, Converter={StaticResource dateConverter}}"
    Style="{StaticResource textStyleTextBlock}"/>
```

## Module 4

### Implementing Layout Using Windows 8 Built-In Controls



## Implementing Layout Using Windows 8 Built-In Controls

### Contents

#### Exam Topic: Create layout aware apps to handle view states

- Handle view state events from ViewStateManager
- Choose between style patterns for the different view states
- Set app orientation in the manifest

#### Exam Topic: Implement layout controls

- Implement the Grid control to structure your layout
- Set the number of rows/columns and size
- Enable zoom and scroll capabilities in layout controls
- Manage text flow and presentation

#### Exam Topic: Design and implement the app bar

- Determine what to put on the app bar based on app requirements
- Style and position app bar items
- Design the placement of controls on the app bar
- Handle AppBar events

Adding app bars (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/hh781230.aspx>

Quickstart: Adding layout controls (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/hh969155.aspx>

Defining layouts and views (Windows Store apps using C#/VB/C++ and XAML)  
[http://msdn.microsoft.com/library/windows/apps/hh465329\(v=win.10\).aspx](http://msdn.microsoft.com/library/windows/apps/hh465329(v=win.10).aspx)

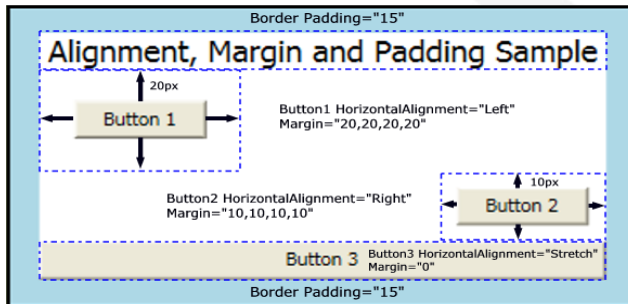


✦ Padding is similar to an “inner” Margin in most respects but only used in some classes

- Block, Border, Control, TextBlock

```
<Border Padding="15"
```

```
myBorder.Padding = new Thickness(15);
```



Alignment, Margins, and Padding Overview  
<http://msdn.microsoft.com/en-us/library/ms751709.aspx>



✦ StackPanels can stack their children horizontally or vertically (the default)

```
<StackPanel orientation="Horizontal">  
  <Border Background="SkyBlue">  
    <TextBlock>Stacked Item #1</TextBlock>  
  </Border>  
  <Border Background="CadetBlue">  
    <TextBlock>Stacked Item #2</TextBlock>  
  </Border>  
  <Border Background="LightGoldenRodYellow">  
    <TextBlock>Stacked Item #3</TextBlock>  
  </Border>  
</StackPanel>
```

Stacked Item #1 Stackd Item #2 Stackd Item #3

```
<StackPanel orientation="Vertical">  
  <Border Background="SkyBlue">  
    <TextBlock>Stacked Item #1</TextBlock>  
  </Border>  
  <Border Background="CadetBlue">  
    <TextBlock>Stacked Item #2</TextBlock>  
  </Border>  
  <Border Background="LightGoldenRodYellow">  
    <TextBlock>Stacked Item #3</TextBlock>  
  </Border>  
</StackPanel>
```

Stacked Item #1  
Stacked Item #2  
Stacked Item #3

StackPanel.Orientation Property  
<http://msdn.microsoft.com/en-us/library/system.windows.controls.stackpanel.orientation.aspx>



## AppBar

✦ An app bar is a UI element that presents navigation, commands, and tools to the user

- Appears at the top or bottom of the page, or both
- Hidden by default, shown when the user right clicks, presses Win+Z, or swipes from the top or bottom edge of the screen
- You can open and close the app bar programmatically by setting the `IsOpen` property
- You can respond to the app bar being opened or closed by handling the `Opened` and `Closed` events, for example, to provide context-sensitive controls

```
<Page.BottomAppBar>
  <AppBar x:Name="bottomAppBar" Padding="10,0,10,0"
    Opened="OnOpened" Closed="OnClosed">
    <Grid> ...
```

AppBar class  
<http://msdn.microsoft.com/library/windows/apps/hh701927>



## Detecting Changes in Size and View Mode

✦ Application View class

- Allows an app to obtain its current view state and attempt to unsnap the app if it is in a snapped state
- View: `FullScreenLandscape`, `Filled`, `Snapped`, `FullScreenPortrait`

✦ SizeChangedEventArgs and the SizeChanged event

- `NewSize` (read-only): Gets the new size of the object reporting the size change
- `PreviousSize` (read-only): Gets the previous size of the object reporting the size change
- `Size` struct represents an ordered pair of floating-point numbers that specify a height and width

ApplicationViewState enumeration  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.viewmanagement.applicationviewstate.aspx>

SizeChangedEventArgs class  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.sizechangedeventargs.aspx>



## Module 5 Handling Files in Windows Store Apps



## Handling Files in Windows Store Apps Contents

### 70-485 Exam Topic: Save and retrieve files from the file system

- Handle file streams
- Save and retrieve files by using StorageFile and StorageFolder classes
- Set file extensions and associations
- Save and retrieve files by using the file picker classes
- Compress files to save space
- Access libraries, including pictures, documents, and videos

Working with data and files (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/br229562.aspx>

Plan for network connectivity (Windows Store apps)  
<http://msdn.microsoft.com/library/windows/apps/jj863302.aspx>

Transferring a file from a network resource (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/hh757628.aspx>



## Protocol Activation

Windows allows an app to register to become the default handler for a certain URI scheme name

- You should only register for a URI scheme name if you expect to handle all URI launches for that type of URI scheme

```
<Package xmlns="http://schemas.microsoft.com/appx/2010/manifest">
  <Applications>
    <Application Id="AutoLaunch.App">
      <Extensions>
        <Extension Category="windows.protocol">
          <Protocol Name="alsdk" />
          <Logo>images\logo.png</Logo>
          <DisplayName>SDK Sample Protocol</DisplayName>
        </Extension>
      </Extensions>
    </Application>
  </Applications>
</Package>
```

How to handle protocol activation (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh779670.aspx>



## Module 6

### Windows Store App Process Lifetime Management



## Windows Store App Process Lifetime Management

### Contents

**Exam Topic: Design and implement Process Lifetime Management (PLM)**

- Choose a state management strategy
- Handle the suspend event
- Prepare for app termination
- Handle the Resume event
- Handle the OnActivated event
- Check the ActivationKind and previous state



## ApplicationExecutionState

State	When this state is seen	What your app should do
NotRunning	<p>The user first activates the app after:</p> <ul style="list-style-type: none"> <li>installing the app from the Store</li> <li>clicking End task in Task Manager while the app is running</li> <li>rebooting the computer</li> <li>logging off and back on</li> </ul> <p>The user closes the app through the close gesture or Alt+F4 and activates it within about 10 seconds of closing it.</p>	Display its initial UI and perform initialization tasks
Running	The app is activated through a secondary tile or one of the activation contracts and extensions while it is running	Respond to the activation event as appropriate

ApplicationExecutionState enumeration

<http://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.activation.applicationexecutionstate>



## ApplicationExecutionState

State	When this state is seen	What your app should do
Suspended	The app is activated through a secondary tile or one of the activation contracts and extensions while Windows is suspending it or after Windows has suspended it	Respond to the activation event as appropriate
Terminated	Windows successfully suspends the app and then it is terminated. For example, Windows can terminate a suspended app if the system is running low on resources.	Restore itself to the way the user left it, rather than starting fresh. Use data saved during app suspension. Refresh content or network connections that may have become stale.
ClosedByUser	The user closes the app through the close gesture or Alt+F4 and takes longer than 10 seconds to activate the app again	Display its initial UI and perform initialization tasks, rather than restoring its previous state





## Module 7

### Working with Resources, Styles, and Templates



## Working with Resources, Styles, and Templates

### Contents

**Exam Topic: Create and manage XAML styles and templates**

- Implement and extend styles and templates
- Implement gradients
- Modify styles based on event and property triggers
- Create shared resources and themes



### ✦ Style using key

- Control.Property

```
<Style x:key="myStyle">  
  <Setter Property="Control.Background" value="Blue" />  
</Style>
```

### ✦ Style using TargetType

- Property

```
<Style x:key="myStyle" TargetType="{x:Type Label}">  
  <Setter Property="Background" value="Blue" />  
</Style>
```



### ✦ Define a style

```
<Style TargetType="TextBox" x:key="DullTextBox">  
  <Setter Property="Foreground" value="Black" />  
  ...  
</Style>
```

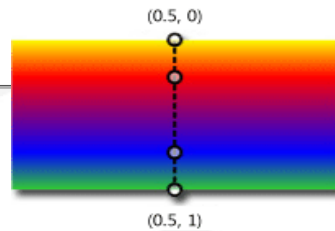
### ✦ Inherit and override

```
<Style TargetType="TextBox" x:key="BrightTextBox">  
  BasedOn="{StaticResource DullTextBox}"  
  <Setter Property="Foreground" value="Yellow" />  
  ...  
</Style>
```



- ✦ StartPoint and EndPoint are co-ordinates in a 2D space, across-down

```
<Rectangle width="200" height="100">  
  <Rectangle.Fill>  
    <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">  
      <GradientStop Color="Yellow" Offset="0.0" />  
      <GradientStop Color="Red" Offset="0.25" />  
      <GradientStop Color="Blue" Offset="0.75" />  
      <GradientStop Color="LimeGreen" Offset="1.0" />  
    </LinearGradientBrush>  
  </Rectangle.Fill>  
</Rectangle>
```



LinearGradientBrush class  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.media.lineargradientbrush.aspx>



- ✦ Ensure resources are the correct type

```
<Page.Resources>  
  <Button Foreground="{StaticResource MyBrush}" ...
```

- ✦ Foreground needs to be a Brush

```
<SolidColorBrush x:Key="MyBrush"  
  color="{StaticResource MyColor}" ...
```

- ✦ Color needs to be a Color

```
<Color x:Key="MyColor">Red</Color>
```



- ✦ Do not have to have a unique key
  - Uses last one found



- ✦ **Template** of a Control (instance of a **ControlTemplate**) decides how a control looks, while the **ContentTemplate** and **ItemTemplate** (**DataTemplates**) decide how the content and items *in* the control look

```
<Button Template={StaticResource CT1}  
      ContentTemplate={StaticResource DT1} ...
```

```
<ListBox Template={StaticResource CT2}  
        ItemTemplate={StaticResource DT2} ...
```

```
<Window.Resources>  
  <ControlTemplate x:Key="CT1">  
    <ContentPresenter /> ...
```

```
<ControlTemplate x:Key="CT2">  
  <ItemsPresenter /> ...
```

```
<DataTemplate x:Key="DT1"><Border ... >  
  <ContentPresenter Content="{Binding}" />
```



## Module 8 Designing and Implementing Navigation in a Windows Store App



## Designing and Implementing Navigation in a Windows Store App Contents

### **Exam Topic: Design and implement navigation in an app**

- Handle navigation events, check navigation properties, and call navigation functions by using the Navigation framework
- Design navigation to meet app requirements
- Semantic Zoom



## Navigation Design

### Hub pages

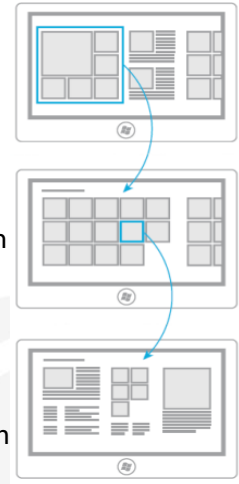
- Here content is displayed in a rich horizontally panning view allowing users to get a glimpse of what's new and available

### Section pages

- Here content consists of individual items, each of which has its own Detail page. Section pages may also take advantage of grouping and a panorama style layout

### Detail pages

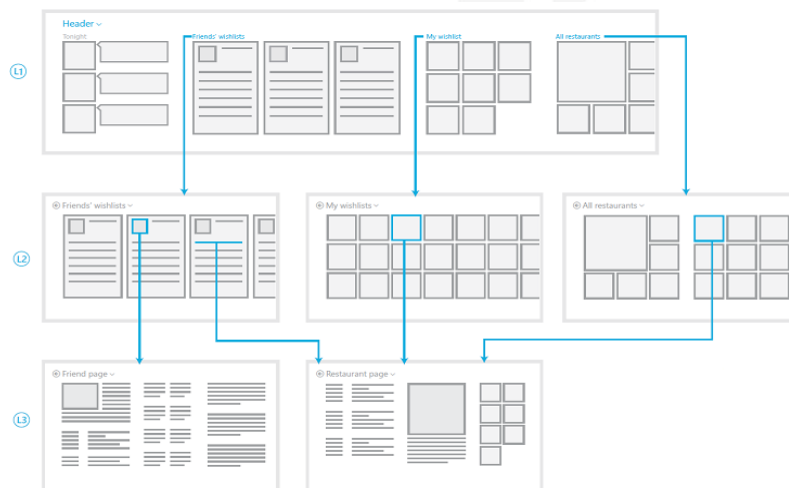
- Here the details of individual items are displayed, the format of which may vary tremendously depending upon the particular type of content



Navigation design for Windows Store apps  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh761500.aspx>



## Navigation Design



## Frame Navigation

✦ Frame is a content control that provides the ability to navigate to and display content

- Frame can be hosted within other content, as with other controls and elements

✦ Frame methods

- GoBack or GoForward: to the most recent item in back or forward navigation history
- Navigate: asynchronously to the specified source content
- When content is navigated to, Frame records the navigation as an entry in navigation history; an entry is added to back navigation history when either a new navigation occurs, by calling the Navigate method, or by navigating to an entry in forward navigation history, by calling GoForward

Frame Class  
<http://msdn.microsoft.com/en-us/library/system.windows.controls.frame.aspx>



## NavigationMode property and enumeration

✦ Specifies the type of navigation that is taking place

- Back: Navigating back to the most recent content in back navigation history; occurs when the GoBack method is called
- Forward: Navigating to the most recent content on forward navigation history; occurs when the GoForward method is called
- New: Navigating to new content; occurs when the Navigate method is called, or when Source property is set
- Refresh: Reloading the current content; occurs when the Refresh method is called

NavigationMode Enumeration  
<http://msdn.microsoft.com/en-us/library/system.windows.navigation.navigationmode.aspx>



## OnNavigatedTo event

✚ Invoked when the Page is loaded and becomes the current source of a parent Frame

- NavigationEventArgs: event data that can be examined by overriding code; representative of the pending navigation that will load the current Page
- Usually the most relevant property to examine is Parameter, an object that potentially passes parameters to the navigation target (may be null)

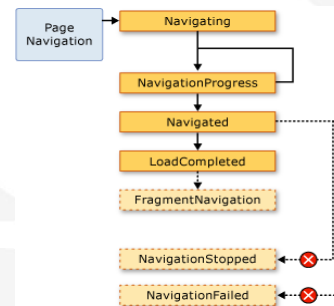
Page.OnNavigatedTo method  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.page.onnavigatedto.aspx>



## Lifetime of Navigation

✚ The lifetime of a navigation can be tracked through the following events:

- Navigating
- Navigated
- NavigationProgress
- NavigationFailed
- NavigationStopped
- LoadCompleted
- FragmentNavigation
- Not all events are raised each time that a navigation occurs; the set of events that are raised is determined by the type of navigation that occurs (content or content fragment) and how the navigation completes (canceled, stopped, or failed)





## Navigation Cache Mode

✦ Gets or sets the navigation mode that indicates whether this Page is cached, and the period of time that the cache entry should persist

- Disabled: The page is never cached and a new instance of the page is created on each visit
- Required: The page is cached and the cached instance is reused for every visit regardless of the cache size for the frame
- Enabled: The page is cached, but the cached instance is discarded when the size of the cache for the frame is exceeded

Page.NavigationCacheMode Property  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.page.navigationcachemode>



## Semantic Zoom

✦ Enables the user to zoom between two different views of the same content

```
<SemanticZoom>
  <SemanticZoom.ZoomedOutView>
    <!-- Put the GridView for the zoomed out view here. -->
  </SemanticZoom.ZoomedOutView>
  <SemanticZoom.ZoomedInView>
    <!-- Put the GridView for the zoomed in view here. -->
  </SemanticZoom.ZoomedInView>
</SemanticZoom>
```

- These controls can be any two controls that implement the ISemanticZoomInformation interface
- There are two controls that implement this interface: ListView and GridView

Quickstart: adding SemanticZoom controls (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh781234.aspx>



## Module 9 Implementing Windows 8 Contracts



### Implementing Windows 8 Contracts Contents (1 of 2)

**Exam Topic: Design for charms and contracts**

- Choose the appropriate charm based on app requirements
- Design your app in a charm- and contract-aware manner
- Configure app manifest for correct permissions

**Exam Topic: Implement search**

- Provide search suggestions using the SearchPane class
- Search for and launch other apps
- Provide and constrain search within an app, including inside and outside of Search charm
- Provide search result previews
- Implement activation from within search

**Exam Topic: Implement Share in an app**

- Use the DataTransferManager class to share data with other apps
- Accept sharing requests by implementing activation from within Share
- Limit the scope of sharing using the DataPackage object
- Implement in-app Share outside of Share charm



### Exam Topic: Manage app settings and preferences

- ❑ Choose which app features are accessed in AppSettings
- ❑ Add entry points for AppSettings in the Settings window
- ❑ Create settings flyouts using the Popup control
- ❑ Add settings to Popup
- ❑ Store and retrieve settings from the roaming app data store

### Exam Topic: Access and display contacts

- ❑ Call the ContactsPicker class
- ❑ Filter which contacts to display
- ❑ Display a set number of contacts
- ❑ Create and modify contact information
- ❑ Select specific contact data

Quickstart: Selecting user contacts (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/jj152724.aspx>

Managing user info (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/br229572.aspx>



### ✦ Triggers for search

- When the user searches your app while it is the main app on screen, the system fires a QuerySubmitted event
- When the user searches your app while it is not the main app on screen, the system fires an Activated event and stores the arguments for this event by using SearchActivatedEventArgs

✦ To use the query text and current filter to populate your page with results, you should add code to the Filter\_SelectionChanged method in the search result page code (SearchResultsPage.xaml.cs by default).

Adding search to an app (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/jj130767.aspx>

Guidelines and checklist for search (Windows Store apps)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh465233.aspx>



## Search How to Add Query Suggestions

9.5

### ✦ By default no filters are shown on your search results page

- You should add filters that make sense for your app and users, by adding code to the LoadState method

### ✦ Suggestions

- You can register for the SuggestionsRequested event and build your own list of suggestions that is made up of suggestions (IEnumerable<string>) you retrieved from another source

```
var pane = SearchPane.GetForCurrentView();
pane.SuggestionsRequested += (sender, args) => {
    args.Request.SearchSuggestionCollection
        .AppendQuerySuggestions(suggestions); };
```

Quickstart: Adding search to an app (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh868180.aspx>

SearchSuggestionCollection class  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.search.searchsuggestioncollection.aspx>



## Sharing DataRequest

9.6

### ✦ Your app receives a DataRequest object when a DataRequested event occurs

- With this object, your app can supply data to a target app by using a DataPackage object, use a deferral object to call a function, or inform the target app that an error occurred

```
private void ShareSourceLoad() {
    var dataTransferManager = DataTransferManager.GetForCurrentView();
    dataTransferManager.DataRequested += new TypedEventHandler<
        DataTransferManager, DataRequestedEventArgs>(this.DataRequested);
}
private void DataRequested(DataTransferManager sender, DataRequestedEventArgs e)
{
    DataRequest request = e.Request;
    request.Data.Properties.Title = "Share Text Example";
    request.Data.Properties.Description = "An example of how to share text.";
    request.Data.SetText("Hello world!");
}
```

DataRequest class  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.datatransfer.datarequest>



- ✦ Use the `FailWithDisplayText` method when your app is unable to supply a `DataPackage` to a target app
  - This method cancels the share operation and supplies a text message to the target app
  - The target app can then display this text to the user to explain why the share operation failed

`DataRequest.FailWithDisplayText`  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.datatransfer.datarequest.failwithdisplaytext>



- ✦ You can create an elegant sharing experience by using a custom shortcut called a QuickLink
  - A QuickLink acts as a link to your app that's customized for a specific set of user actions
  - A QuickLink doesn't actually store any data, it contains an identifier that, when selected, is sent to your app, so your app needs to store the data somewhere—such as in the cloud, or on the user's computer—along with its associated ID
- ✦ Email sharing is an example
  - Users have to supply the email addresses of the people they want to share with but adding them repeatedly becomes tedious. A better experience is to create a QuickLink so the user can repeat their choices in one step

How to create a QuickLink (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh973050.aspx>



### ✦ An app can control the Settings Charm pane

- Add or remove commands, receive a notification when the user opens the pane, or open the pane programmatically

SettingsPane	Description
CommandsRequested event	Occurs when the user opens the settings pane. Listening for this event lets the app initialize the setting commands and pause its UI until the user closes the pane
GetForCurrentView method	Gets a SettingsPane object that is associated with the current app view
Show method	Displays the Settings Charm pane to the user
Edge property	Gets a value indicating whether the Settings charm appears on the left or right edge of the screen

```
SettingsPane.GetForCurrentView().CommandsRequested += onCommandsRequested;
```

SettingsPane class  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.applicationsettings.settingspane>



### ✦ ApplicationData.RoamingSettings gets the application settings container in the roaming app data store

- Name of each setting can be 255 characters in length at most
- Each setting can be up to 8K bytes in size and each composite setting can be up to 64K bytes in size
- The sync engine may limit the total size of settings and files that can roam

```
var roamingSettings =  
    Windows.Storage.ApplicationData.Current.RoamingSettings;  
roamingSettings.Values["exampleSetting"] = "Hello World";  
object value = roamingSettings.Values["exampleSetting"];  
if (value == null) { ... } else { ... }  
roamingSettings.Values.Remove("exampleSetting");
```

ApplicationData.RoamingSettings  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.storage.applicationdata.roamingsettings>



### ✦ Use a SettingsCommand to represent a settings entry

- This can be appended to the ApplicationCommands vector

```
void onCommandsRequested(SettingsPane settingsPane,
    SettingsPaneCommandsRequestedEventArgs eventArgs) {
    var handler = new UICommandInvokedHandler(onSettingsCommand);
    var generalCommand = new SettingsCommand(
        "generalSettings", "General", handler);
    eventArgs.Request.ApplicationCommands.Add(generalCommand);
    var helpCommand = new SettingsCommand("helpPage", "Help", handler);
    eventArgs.Request.ApplicationCommands.Add(helpCommand);
}
```

```
void onSettingsCommand(UICommand command) {
    SettingsCommand settingsCommand = (SettingsCommand)command;
    rootPage.NotifyUser("You selected " + settingsCommand.Label,
        NotifyType.StatusMessage); // NotifyUser is a custom method
}
```

Guidelines for app settings (Windows Store apps)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh770544.aspx>

SettingsCommand class  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.applicationsettings.settingscommand.aspx>



### ✦ Launch the Contact Picker for selecting contacts

- PickSingleContactAsync and PickMultipleContactsAsync
- Use the SelectionMode property to control whether your app receives all of a contact's information, or just specific fields
- DesiredFields is an IList<string> of field names your app wants

```
using windows.ApplicationModel.Contacts;

var picker = ContactPicker();
picker.CommitButtonText = "Select";
picker.SelectionMode = ContactSelectionMode.Fields;
picker.DesiredFields.Add(KnownContactField.Email);
IReadOnlyList<ContactInformation> contacts =
    await picker.PickMultipleContactsAsync();
```

Quickstart: Selecting user contacts (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/jj152724.aspx>

ContactPicker.DesiredFields  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.contacts.contactpicker.desiredfields.aspx>



## Common Exam Questions

- ✦ You will often get questions about getting a reference to a pane for contracts
  - Use `GetForCurrentView()` method, not a constructor

**DataTransferManager.GetForCurrentView**  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.datatransfer.datatransfermanager.getforcurrentview>

**SearchPane.GetForCurrentView**  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.search.searchpane.getforcurrentview>

**SettingsPane.GetForCurrentView**  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.applicationsettings.settingspane.getforcurrentview>





## Module 10 Implementing Tiles and User Notifications



### Implementing Tiles and User Notifications Contents

**Exam Topic: Create and manage tiles**

- Create and update tiles and tile contents
- Create and update badges (TileUpdateManager class)
- Respond to notification requests
- Choose an appropriate tile update schedule based on app requirements

**Exam Topic: Notify users by using toast**

- Enable an app for toast notifications
- Populate toast notifications with images and text using the ToastUpdateManager class
- Play sounds with toast notifications
- Respond to toast events
- Control toast duration

Creating tiles (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/hh868260.aspx>

Pinning secondary tiles (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/hh868258.aspx>

Sending toast notifications (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/library/windows/apps/hh868266.aspx>



### ✦ TileUpdateManager creates TileUpdater objects used to change and update Start menu tiles

- Provides access to the XML content of the system-provided tile templates so that you can customize that content for use in updating your tiles
- By default, local tile and badge notifications do not expire and push, periodic, and scheduled notifications expire after three days. It is a best practice to set an expiration time, particularly on local tile and badge notifications

```
TileNotification tileNotification = new TileNotification(tileXml);  
tileNotification.ExpirationTime = DateTimeOffset.UtcNow.AddSeconds(10);  
TileUpdateManager.CreateTileUpdaterForApplication()  
.Update(tileNotification);
```

Quickstart: Sending a tile update (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh868253.aspx>



### ✦ Enables the tile to queue up to five notifications

- When queuing is enabled, a maximum of five tile notifications can automatically cycle on the tile
- Be careful about enabling cycling unless your app explicitly wants to use it, otherwise you can potentially have outdated notifications cycling through
- Generally, the queue is FIFO (first in, first out), so that when it is full and a new notification arrives, the oldest notification is removed
- However, notifications can be given a Tag, which allows a new notification with that same Tag to replace its older notification in the queue, regardless of its position

TileUpdater.EnableNotificationQueue  
<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.notifications.tileupdater.enablenotificationqueue.aspx>

How to use the notification queue with local notifications (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh868234.aspx>



- ✦ A badge is a number or glyph that is displayed in the lower right corner of a tile to indicate an app's status
  - The badge is an overlay on the tile, not a part of the tile itself
  - The badge is manipulated through its own APIs and schema and is updated through its own notifications

```

XmlDocument badgeXml = BadgeUpdateManager
    .GetTemplateContent(BadgeTemplateType.BadgeNumber);
XmlElement badgeElement = (XmlElement)badgeXml.SelectSingleNode("/badge");
badgeElement.SetAttribute("value", "7");
BadgeNotification badge = new BadgeNotification(badgeXml);
BadgeUpdateManager.CreateBadgeUpdaterForApplication().Update(badge);
    
```

Quickstart: Sending a badge update (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh868225.aspx>



## Toast Two Ways to Make Toast



Integer nisl sapien, sagittis non  
 viverra a. É ullamcorper quis sapien.  
 Etiam est nisl, faucibus dolor amet...

- ✦ The ToastImageAndText01 template must look like this

```

<toast>
  <visual>
    <binding template="ToastImageAndText01">
      <image id="1" src=""/>
      <text id="1"></text>
    </binding>
  </visual>
</toast>
    
```

- ✦ Easiest to create by getting a pre-defined template

```

var toastXml = ToastNotificationManager.GetTemplateContent(
    ToastTemplateType.ToastImageAndText01);
    
```

- ✦ Or create it manually

```

var bindingElem = visualElem.CreateElement("binding");
bindingElem.SetAttribute("template",
    ToastTemplateType.ToastImageAndText01.ToString());
    
```

Quickstart: Sending a toast notification (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh868254.aspx>



## Toast Customizing Duration and Audio

10.7

✦ You can optionally set a display duration for your toast

- There are two values: "short" (the default) and "long"
- Use "long" only if your notification is part of a scenario such as an incoming call or appointment reminder

```
IXmlNode toastNode = toastXml.SelectSingleNode("/toast");  
(XmlElement)toastNode.SetAttribute("duration", "long");
```

✦ Windows plays a sound when your toast is displayed

- You can optionally specify a different sound from a system-provided set of sounds, or no sound at all

```
XmlElement audio = toastXml.CreateElement("audio");  
audio.SetAttribute("src", "ms-winsoundevent:Notification.IM");
```

```
audio.SetAttribute("silent", "true");
```

Quickstart: Sending a toast notification (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh868254.aspx>



## Toast Activating Your App With Toast

10.8

✦ When your app is activated through a toast notification, it needs to be given information related to the content of the toast

- It uses the launch attribute to specify this activation information

```
((XmlElement)toastNode).SetAttribute("launch",  
"{\"type\":\"toast\",\"param1\":\"12345\",\"param2\":\"67890\"}");
```

```
<toast launch="{&quot;myContext&quot;:&quot;12345&quot;}">  
  <visual>  
    <binding template="ToastImageAndText01">  
      <image id="1" src="ms-appx:///images/redwide.png"  
        alt="red graphic"/>  
      <text id="1">Hello world!</text>
```

```
protected override void OnLaunched(LaunchActivatedEventArgs args) {
```

How to handle activation from a toast notification (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh868212.aspx>



## Module 11

### Designing and Implementing a Data Access Strategy



## Designing and Implementing a Data Access Strategy

### Contents

Topic	Slide
OData	3
WCF Data Services	7
HTTP Methods	9
OData .NET Clients	12
HTTP Clients	16
HTTP Responses	19
SOAP vs REST	21
REST and Web API	26

#### Exam Topic: Choose an appropriate data access strategy

- Choose the appropriate data access strategy (file based, web service, remote storage, including Windows Azure storage) based on requirements

#### Exam Topic: Retrieve data remotely

- Use HttpClient to retrieve web services
- Set the appropriate HTTP verb for REST
- Consume SOAP/WCF services
- Use WebSockets for bi-directional communication
- Handle the progress of data requests

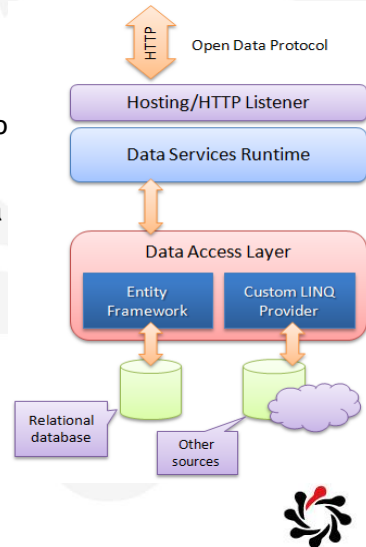


- ✦ OData is a standard for building HTTP services that follow standards for querying the data model
  - It defines a query syntax using URIs similar to SQL
- ✦ Two technologies for creating an OData service
  - WCF Data Services (.NET 3.5 +)
  - ASP.NET Web API OData (.NET 4.5 +)

WCF Data Services and OData At-a-Glance  
<http://msdn.microsoft.com/en-us/data/aa937697>

WCF Data Services  
<http://msdn.microsoft.com/en-us/data/odata.aspx>

WCF Data Services Blog  
<http://blogs.msdn.com/b/astoriateam/>



- ✦ To select or order by multiple columns use a comma-separated list
- ✦ Case-sensitive!
- ✦ Must use \$ prefix for keywords
  - \$select, \$filter, \$orderby, \$expand
  - \$top, \$skip
  - /\$count: return int
  - \$inlinecount: a count is included with the feed
  - \$links
  - \$metadata

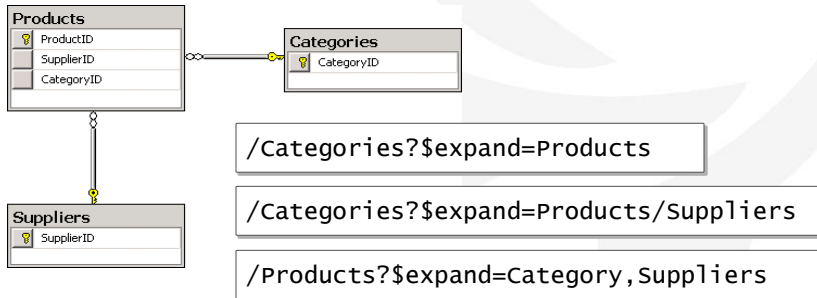
```
http://.../AW.svc/Contacts?  
  $select=FirstName,LastName,Age&  
  $filter=State eq 'CA' and Price gt 500&  
  $orderby=LastName,Age
```

OData: URI Conventions  
<http://www.odata.org/documentation/uri-conventions#QueryStringOptions>



The syntax of a \$expand query option is a comma-separated list of Navigation Properties

- Each Navigation Property can be followed by a forward slash and another Navigation Property to enable identifying a multi-level relationship



Expand System Query Option (\$expand)  
<http://www.odata.org/documentation/uri-conventions#ExpandSystemQueryOption>



URI	Returns
<code>/Customers('ALFKI')/ContactName</code>	An XML element that contains the ContactName property value for a specific Customer
<code>/Customers('ALFKI')/ContactName/\$value</code>	Only the string "Maria Anders" without the XML element
<code>/Customers('ALFKI')/Orders</code>	All the orders that are related to a specific Customer
<code>/Orders(10643)/Customer</code>	A reference to the Customer entity to which a specific Order entity belongs
<code>/Orders?\$filter=not endswith(ShipPostalCode,'100')</code>	All the orders the postal codes of which do not end in 100
<code>/Categories(1)/\$links/Products</code>	Links to the data instead of the actual data e.g. <code>&lt;uri&gt;http://.../Products(4)&lt;/uri&gt;</code>
<code>/Categories?\$select=Name, Products&amp;\$expand=Products</code>	Must select Products if expanding Products

Accessing Data Service Resources (WCF Data Services)  
<http://msdn.microsoft.com/en-us/library/dd728283.aspx>



✚ In any web project

- Project - Add New Item - WCF Data Service
- Create a context class that represents your data
  - ADO.NET Entity Data Model is easiest
  - Or any class that has properties of type IQueryable<T> where T is an “entity” (and optionally implements IUpdatable)
- Use context class in DataService<TContext>
- Set permissions

```
public class BlogService : DataService<BlogContext> {  
    public static void InitializeService(  
        DataServiceConfiguration config) {  
        config.SetEntitySetAccessRule("Blogs", EntitySetRights.All);  
        config.SetServiceOperationAccessRule(  
            "MyServiceOperation", ServiceOperationRights.All);  
        config.DataServiceBehavior.MaxProtocolVersion =  
            DataServiceProtocolVersion.V2; } }
```



✚ WCF Data Services enables an application to intercept request messages so that you can add custom logic

- Define a query interceptor for the Orders entity set

```
[QueryInterceptor("Orders")]  
public Expression<Func<Order, bool>> OnQueryOrders()  
{  
    return o => o.Customer.ContactName ==  
        HttpContext.Current.User.Identity.Name;  
}
```

- Check operations to determine type of change

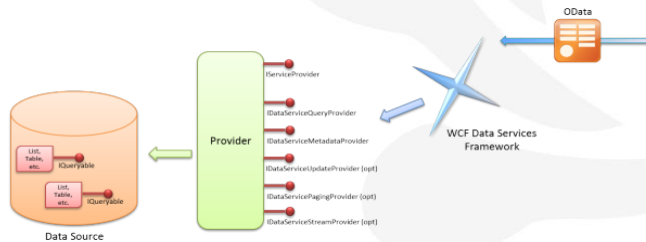
```
[ChangeInterceptor("Products")]  
public void OnChangeProducts(  
    Product product, UpdateOperations operations)
```

Interceptors (WCF Data Services)  
<http://msdn.microsoft.com/en-us/library/dd744842.aspx>





- ✦ A Data Service Provider is simply a .NET class that sits between the Data Services Framework and the underlying data source that's being exposed



```
public class MyDataSource : IUpdatable {  
    public IQueryable<Product>Products { get { ...  
}
```

Custom Data Service Providers  
<http://msdn.microsoft.com/en-us/data/gg191846.aspx>



- ✦ To update a column of a record without overwriting other columns, use MERGE verb and only pass the changed column values

```
MERGE /AW.svc/Contacts(23)  
Host: Adventureworks.com  
Content-Type: application-json  
{ State: 'CA' }
```

- ✦ Use SaveChangesOptions.ReplaceOnUpdate for PUT

✦ Warning!

- By default the WCF Data Services client library passes all properties in MERGE, not just the ones that have changed

WCF Data Services: Optimizing bandwidth usage and performance with updates  
<http://blogs.infosupport.com/wcf-data-services-optimizing-updates-in-the-client-library/>



- ✦ To enable CRUD operations, IIS must allow the following methods on the .svc extension
  - PUT
  - DELETE



- ✦ Some network intermediaries block HTTP verbs like DELETE or PUT or MERGE
  - “Verb tunnelling” or “POST tunnelling” gets around this
- ✦ Uses HTTP POST to “wrap” another verb

```
POST /Categories(5)
HTTP/1.1
Host: Adventureworks.com
X-HTTP-Method: DELETE
```

- ✦ To enable on client

```
DataServiceContext.UsePostTunneling = true;
```

2.2.5.8 X-HTTP-Method  
[http://msdn.microsoft.com/en-us/library/dd541471\(prot.10\).aspx](http://msdn.microsoft.com/en-us/library/dd541471(prot.10).aspx)



- ✦ DataServiceContext does not support lazy loading so you must use the LoadProperty method to explicitly load related entities

```
context.LoadProperty(order, "LineItems");  
foreach(var item in order.LineItems) {
```

- ✦ Or use Expand method to pre-load (“eager loading”)

```
... from o in aw.Orders.Expand("LineItems") ...
```

DataServiceContext.LoadProperty Method - <http://msdn.microsoft.com/en-us/library/system.data.services.client.dataservicecontext.loadproperty.aspx>



- ✦ To find out how a LINQ to OData query will translate into an OData URL use RequestUri

```
var query = from p in db.Products  
            where p.Color == "Red"  
            select p;  
string uri =  
    ((DataServiceQuery)query).RequestUri.ToString();
```

```
http://localhost:1034/AW.svc/Products()  
?$filter=Color eq 'Red'
```



✦ Create an event handler for SendRequest

```
context.SendingRequest += new EventHandler  
<SendingRequestEventArgs>(OnSendingRequest);
```

✦ Add the header

```
private static void OnSendingRequest(  
    object sender, SendingRequestEventArgs e) {  
    // Add an Authorization header that contains an  
    // OAuth WRAP access token to the request.  
    e.RequestHeaders.Add("Authorization",  
        "WRAP access_token=\"123456789\"");  
}
```

How to: Set Headers in the Client Request (WCF Data Services)  
<http://msdn.microsoft.com/en-us/library/gg258441.aspx>



✦ By using the new async and await keywords, you can use resources to create an asynchronous method almost as easily as you create a synchronous method

async modifier, Task<T> return type, Async suffix for name

```
async Task<int> AccessTheWebAsync()  
{  
    HttpClient client = new HttpClient();  
    Task<string> getStringTask =  
        client.GetStringAsync("http://msdn.microsoft.com");  
    DoIndependentWork(); // executes while async op works  
    string urlContents = await getStringTask;  
    return urlContents.Length;  
}
```

Waits until task is complete, control returns to the caller of AccessTheWebAsync



✦ Provides common methods for sending data to and receiving data from a resource identified by a URI

Method	Description
DownloadData	Downloads resource as a Byte array from the URI specified
DownloadDataAsync	Downloads resource as a Byte array from the URI specified as an asynchronous operation
DownloadDataTaskAsync	Downloads resource as a Byte array from the URI specified as an asynchronous operation using a task
DownloadFile, ...	Downloads resource with the specified URI to a local file
DownloadString, ...	Downloads the requested resource as a String
UploadData, ...	Uploads the data (a byte array) as ...
UploadFile, ...	Uploads a local file as ...
UploadValues, ...	Uploads a NameValueCollection as ...

WebClient Class  
[http://msdn.microsoft.com/en-us/library/system.net.webclient\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.net.webclient(v=vs.110).aspx)



✦ Uploads the specified name/value collection to the resource identified by the specified URI

- For an HTTP resource, the POST method is used
- If the Content-type header is null, the UploadValues method sets it to "application/x-www-form-urlencoded"

```
var mywebClient = new WebClient();
var nvc = new NameValueCollection();
nvc.Add("Name", name);
nvc.Add("Address", address);
nvc.Add("Age", age);
byte[] responseArray = mywebClient.UploadValues(uri, nvc);
// Encoding.ASCII.GetString(responseArray)
```

WebClient.UploadValues Method (String, NameValueCollection)  
<http://msdn.microsoft.com/en-us/library/9w7b4fz7.aspx>



## HTTP Clients HttpResponseMessage Properties

11.19

Property	Description
Content	Gets or sets the content of a HTTP response message
Headers	Gets the collection of HTTP response headers
IsSuccessStatusCode	Gets a value that indicates if the HTTP response was successful (read-only bool)
ReasonPhrase	Gets or sets the reason phrase which typically is sent by servers together with the status code
RequestMessage	Gets or sets the request message which led to this response message
StatusCode	Gets or sets the status code of the HTTP response
Version	Gets or sets the HTTP message version

HttpResponseMessage Properties  
[http://msdn.microsoft.com/en-us/library/system.net.http.httpresponsemessage\\_properties.aspx](http://msdn.microsoft.com/en-us/library/system.net.http.httpresponsemessage_properties.aspx)



## HTTP Clients HTTP Request Authorization

11.20

### From StackOverflow

- “For a web page that exists, but for which a user that does not have sufficient privileges, (they are not logged in or do not belong to the proper user group), what is the proper HTTP response to serve?”

### To determine if a request needs to be authorized

- Check the StatusCode or ReasonPhrase properties of the HttpResponseMessage
- 401 Unauthorized: If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials
- 403 Forbidden: The server understood the request, but is refusing to fulfill it

403 Forbidden vs 401 Unauthorized HTTP responses  
<http://stackoverflow.com/questions/3297048/403-forbidden-vs-401-unauthorized-http-responses>



## SOAP Services versus RESTful Services

- ✿ When developers talk about “web” services, they usually mean SOAP services, for example, ASMX files
  - Actually, SOAP services can use any protocol, since SOAP defines a message format, not an architectural style, which is why WCF supports so many bindings
- ✿ RESTful services are true web services since they are built on the architecture of the web
- ✿ Proponents of RESTful services are sometimes called RESTafarians and are often quite passionate about how “evil” SOAP is and how “beautiful” REST is
- ✿ In WCF 3.5 Microsoft embraced REST so that you can use WCF to create both SOAP and RESTful services
  - In ASP.NET 4.5 Microsoft added Web API which is even better



## 21<sup>st</sup> Century Service-Oriented Architecture

- ✿ In 2000 Roy Fielding wrote a doctoral dissertation
  - The web is the world’s largest and most scalable distributed application platform
  - He described the architecture of the web and distilled from it an architectural style based on the factors that led to its success
  - He named this style REST and suggested its use to build services
- ✿ WCF isn’t tied to SOAP so Microsoft was able to quickly embrace REST once its simple power was understood
- ✿ RESTafarians believe that REST should be your first choice when building services



### ✿ Principles

- Addressable resources (URIs)
- Uniform interface for interacting with resources (HTTP verbs: GET, POST, DELETE, PUT, etc.)
- Standard resource formats (HTML, JPEG, XML, JSON, etc.)
- Statelessness between clients and servers (provides scalability and manageability)
- Hyperlinking for navigation between resources (relationships)

### ✿ GET

- The cacheability of the GET verb contributes to the scalability of the web
- GETs are also considered “safe” in the sense that they should not cause side effects i.e. they don’t change resources



### ✿ SOAP doesn’t follow the architecture of the web at all

- Rather than URIs, SOAP uses *actions*, which are a thin veneer over method calls
- SOAP services usually have only one URI and many different actions
- SOAP is really an interoperable cross-platform remote procedure call (RPC) system

### ✿ When using HTTP, SOAP only uses one HTTP verb, POST

- POSTs cannot be cached, so it’s not as scalable as GET

### ✿ But SOAP wasn’t designed for the web and goes out of its way to be protocol independent





- ✦ REST services combine nouns (e.g. resources defined by URIs) with verbs (e.g. GET, DELETE)

```
PUT /Aw.svc/Products(123)
Host: http://localhost:801
Content-Length: 223
Content-Type: application/xml
```

```
<Product>
  <ProductID>123</ProductID>
```

- ✦ SOAP services use a message to contain the nouns (e.g. the payload in the body) with verbs (the action in the header)

```
<s:Envelope xmlns:s=" ... >
  <s:Header>
    <To>http://.../Sample.svc</To>
    <Action>AddProduct</Action>
```

```
<s:Body>
  <Product>
    <ProductID>123</ProductID>
    <ProductName>Fish</ProductName>
```



- ✦ Choose “common sense” URIs so developers can quickly work out how to access any resource and your service becomes almost “self-documenting”

- Design your service API as if you were designing the URLs for a web site i.e. make them logical enough that an end user could work out how to use them if shown a few examples

Task	HTTP Method	Relative URI
Retrieve all entities	GET	/api/orders
Retrieve single entity	GET	/api/orders/ <i>id</i>
Retrieve by custom	GET	/api/orders?category= <i>category</i>
Create new entity	POST	/api/orders
Update entity	PUT (or POST)	/api/orders/ <i>id</i>
Remove entity	DELETE	/api/orders/ <i>id</i>



✿ With REST there is no need to design the semantics of the actions you want to perform because HTTP already defines them for you

- GET (or HEAD): retrieves a resource (or without the body)
- POST: usually means create a new resource (although the effect of this verb is not defined in the HTTP standards)
- PUT: replace a resource with a new version (idempotent - the effect of calling it many times is the same as calling it once)
- DELETE: remove a resource (idempotent)
- CONNECT, TRANSFER, OPTIONS, TRACE

✿ The Atom protocol defines

- MERGE: a more efficient way of updating an existing resource

Method Definitions  
<http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>



✿ REST has no restrictions on resource formats

✿ A REST service's resource types are technically known as *media types*

✿ The media type is returned in the HTTP header as the Content-Type

- XML: application/xml (text/xml is old standard)
- Feeds: application/rss+xml or application/atom+xml
- HTML and XHTML: text/html or application/xhtml+xml
- JavaScript Object Notation: application/json
  - JSON is most popular because it is a more compact format

Internet Media Types  
[http://en.wikipedia.org/wiki/Internet\\_media\\_type](http://en.wikipedia.org/wiki/Internet_media_type)



✦ Achieving zero-lag connectivity between Web clients and servers requires going beyond the HTTP protocol

- The new WebSocket Protocol aims to overcome a structural limitation of the HTTP protocol that makes it inefficient for Web applications hosted in browsers to stay connected to the server over a persistent connection
- Great for real-time updates like stock prices

Understanding the Power of WebSockets  
<http://msdn.microsoft.com/en-us/magazine/hh975342.aspx>



## Module 12

### Responding to Mouse and Touch



## Responding to Mouse and Touch

### Contents

- Exam Topic: Manage input devices**
- Capture Gesture library events
  - Create custom gesture recognizers
  - Listen to mouse events or touch gestures
  - Manage Stylus input and inking



## Common Interactions in Windows 8

### ✦ Learning

- The press and hold gesture displays detailed info or teaching visuals (for example, a tooltip or context menu) without committing to an action



### ✦ Commanding

- The tap gesture invokes a primary action, for example launching an app or executing a command



### ✦ Panning

- The slide gesture is used primarily for panning interactions but can also be used for moving, drawing, or writing



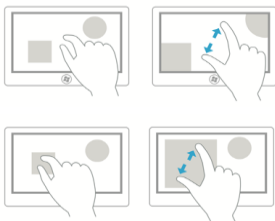
### ✦ Rotating

- Simulates the experience of rotating a piece of paper on a flat surface



## Common Interactions in Windows 8

### ✦ Optical zoom and resizing



- Optical zoom adjusts the magnification level of the entire content area to get a more detailed view of the content
- In contrast, resizing is a technique for adjusting the relative size of one or more objects within a content area without changing the view into the content area

### ✦ Semantic Zoom

- Semantic Zoom is a touch-optimized technique for presenting and navigating structured data or content within a single view



✦ With touch interactions, your Windows Store app can use physical gestures to emulate the direct manipulation of UI elements

- Pointer events are used to get basic contact info such as location and device type, extended info such as pressure and contact geometry, and to support more complex interactions
- Gesture events are used to handle static single-finger interactions such as tapping and press-and-hold (double-tap and right-tap are derived from these basic gestures)
- Manipulation events are used for dynamic multi-touch interactions such as pinching and stretching, and interactions that use inertia and velocity data such as panning/scrolling, zooming, and rotating

Gestures, manipulations, and interactions (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/jj883700.aspx>



✦ You can respond to simple touch and mouse gestures by handling high-level events such as Tapped and Holding

- Pointer events such as `PointerMoved` can be used to support simple, one-finger interactions such as sliding

```
<Rectangle PointerPressed="TestRectangle_PointerPressed"  
  PointerReleased="TestRectangle_PointerReleased"  
  PointerExited="TestRectangle_PointerExited" />
```

✦ For multi-touch interactions such as pinching, and interactions that use inertia and velocity data such as dragging, you use the manipulation events

- The information provided is touch data such as position, translation delta, and velocity; it is your responsibility to convert this information into the equivalent interaction

Quickstart: Touch input (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/465387.aspx>



## Using Manipulation Events

- ✦ Any gesture in a Windows Store app consists of a series of manipulation events
  - Each gesture starts with a ManipulationStarted event, such as when a user touches the screen
  - Next, one or more ManipulationDelta events are fired, for example, if you touch the screen and then drag your finger across the screen, multiple ManipulationDelta events are fired
  - Finally, a ManipulationCompleted event is raised when the interaction is finished
- ✦ The following example creates a Rectangle that can be dragged across the screen:

```
<Rectangle Name="TestRectangle"
  width="200" Height="200" Fill="Blue"
  ManipulationMode="All"/>
```



## Using Manipulation Events

```
// Global Transform used to change the position of the Rectangle.
private TranslateTransform dragTranslation;

public MainPage()
{
    InitializeComponent();
    // Add handler for the ManipulationDelta event
    TestRectangle.ManipulationDelta += Drag_ManipulationDelta;
    dragTranslation = new TranslateTransform();
    TestRectangle.RenderTransform = this.dragTranslation;
}

void Drag_ManipulationDelta(object sender,
    ManipulationDeltaRoutedEventArgs e)
{
    // Move the rectangle.
    dragTranslation.X += e.Delta.Translation.X;
    dragTranslation.Y += e.Delta.Translation.Y;
}
```



## ManipulationModes enumeration

Member	Description
TranslateX, TranslateY	Permit manipulation actions that translate the target on the X or Y axis
TranslateRailsX, TranslateRailsY	Permit manipulation actions that translate the target on the X or Y axis but using a rails mode
Rotate	Permit manipulation actions that rotate the target
Scale	Permit manipulation actions that scale the target
TranslateInertia	Apply inertia to translate actions
RotateInertia	Apply inertia to rotate actions
ScaleInertia	Apply inertia to scale actions
All	Enable all manipulation interaction modes except those supported through Direct Manipulation
System	Enable system-driven touch interactions supported through Direct Manipulation

Specifies how a target UI element interprets manipulation events. This enumeration is flagwise to facilitate setting multiple modes.



## RenderTransform

✎ Gets or sets transform information that affects the rendering position of this element

- A render transform does not regenerate layout size or render size information
- Render transforms are typically intended for animating or applying a temporary effect to an element, for example, the element might zoom when focused or moused over, or might jitter on load to draw the eye to that part of the user interface

✎ MatrixTransform, RotateTransform, ScaleTransform (shrink or grow), SkewTransform, TransformGroup, TranslateTransform (move)

Transform Class Inheritance Hierarchy  
<http://msdn.microsoft.com/en-us/library/system.windows.media.transform.aspx#inheritancecontinued>

UIElement.RenderTransform Property  
<http://msdn.microsoft.com/en-us/library/system.windows.uielement.rendertransform.aspx>





- ✿ Make an app that can create handwritten notes, drawings, and annotations, by adding support for a pen or stylus

```
using Windows.UI.Input.Inking;
```

```
<Canvas x:Name="InkCanvas" Background="white" Margin="62,0,62,10" />
```

```
InkManager _inkManager = new InkManager();
```

```
InkCanvas.PointerPressed += new PointerEventHandler(InkCanvas_PointerPressed);  
InkCanvas.PointerMoved += new PointerEventHandler(InkCanvas_PointerMoved);  
InkCanvas.PointerReleased += new PointerEventHandler(InkCanvas_PointerReleased);  
InkCanvas.PointerExited += new PointerEventHandler(InkCanvas_PointerReleased);
```

Responding to pen and stylus interactions (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/jj883711.aspx>

Quickstart: Capturing ink data (Windows Store apps using C#/VB/C++ and XAML)  
<http://msdn.microsoft.com/en-us/library/windows/apps/974457.aspx>



## Module 13 Planning for Windows Store App Deployment



### Planning for Windows Store App Deployment Contents

**Exam Topic: Plan for an app deployment**

- Plan a deployment based on Windows Store app certification requirements
- Prepare an app manifest (capabilities and declarations)
- Sign an app
- Plan the requirements for an enterprise deployment



## Windows Store app certification requirements

1. Windows Store apps provide value to the customer
2. Windows Store apps can display ads but are more than just ads or websites
3. Windows Store apps behave predictably
  - For example, your app must provide visual feedback when users touch interactive elements
  - For example, your app must support a snapped layout
4. Windows Store apps put the customer in control
5. Windows apps are appropriate for a global audience
6. Windows Store apps are easily identified and understood
7. Desktop apps must follow additional requirements

Windows 8 app certification requirements  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh694083.aspx>



## Windows Store app certification requirements

### ✦ Requirements do not include:

- Which language you write the code in (you can use C#, Visual Basic, HTML5 and JavaScript, C++ and so on)
- Using all the features of the device, e.g., connecting to the Internet, taking photos, GPS, and so on

### ✦ You can display ads, but it must not display *only* ads

- Ads in your apps must comply with our content policies
- Your app must not use its description, tiles, notifications, app bar, or the swipe-from-edge interactions to display ads
- The elements of your app's description, such as screenshots, text, and promotional images must describe your app and not contain additional advertising
- Ads must not execute program code that didn't come from the ad provider



✦ The Manifest Designer has the following tabs:

- Application UI: how your app tile appears on the user's Start screen, whether your app can display notifications to users, and the appearance of the app's splash screen
- Capabilities: specify the resources that your web app can access programmatically, such as the Pictures library or connected devices such as a webcam
- Declarations: extend or customize standard Windows features for use in your Windows store apps. For example, an app can declare that it handles files that have certain file extensions
- Packaging: specifies a unique identity for the package, including name, version, and publisher information

Using the Manifest Designer (Windows Store apps)  
<http://msdn.microsoft.com/en-us/library/windows/apps/br230259.aspx>

App packages and deployment (Windows Store apps)  
<http://msdn.microsoft.com/en-us/library/windows/apps/hh464929.aspx>



## Signing an App

✦ App package's digital signature file: AppxSignature.p7x

- The app package signature ensures that the package and contents haven't been modified after they were signed
- If the signing certificate validates to a Trusted Root Certification Authorities Certificate, the signature also identifies who signed the package
- If you don't use Microsoft Visual Studio 2012 to create and sign your app packages, you need to create and manage your own code signing certificates by using MakeCert.exe and Pvk2Pfx.exe, and then sign your app by using SignTool.exe

```
SignTool sign /fd hashAlgorithm /a /f signingCert.pfx /p  
password filepath.appx
```

How to create an app package signing certificate  
<http://msdn.microsoft.com/en-us/library/windows/apps/jj835832.aspx>

How to sign an app package using SignTool  
<http://msdn.microsoft.com/en-us/library/windows/apps/jj835835.aspx>



## Deploying Enterprise Apps

✦ If you're writing a proprietary line-of-business app, it's likely that you want to deploy it yourself—a process called *sideloading*

- Windows Server 2012 and Windows 8 Enterprise editions are classified as “enterprise sideloading enabled” meaning that the PCs are ready to receive the apps that you deploy outside of the Windows Store
- If you are deploying apps to Windows 8 Pro, Windows RT, or Windows 8 Enterprise, you can configure them for sideloading apps by activating the product key for enterprise sideloading

Deploying enterprise apps  
<http://msdn.microsoft.com/en-us/library/windows/apps/jj657971.aspx>

