Introduction to Silverlight 3
The Basics

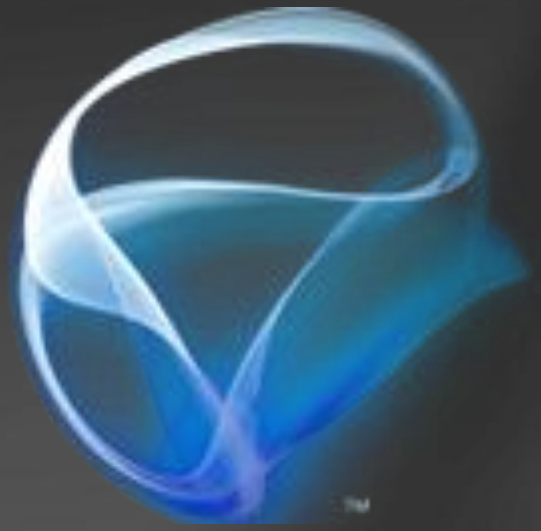
Lance J Collins
Course 6, Class of 2009
SDE
Visual Studio Platform Extensibility

Microsoft Silverlight

Highlights

- Silverlight 3 – What and Why?
- Tools
- Getting Started
- Layout
- Controls
- Control Customization
- Application Model

Silverlight defined…



Microsoft Silverlight is a cross-browser, cross platform and cross device plug-in for delivering the next generation of .NET based media experiences and rich interactive applications for the Web and devices

Microsoft Silverlight

Why Silverlight?

- Cross Platform, Cross Browser Support
- Uses C# and Visual Basic
  - Desktop application developers can program in a familiar environment
- Features
  - Sophisticated RIAs
  - HD video and advanced streaming techniques
  - Interactivity with high-resolution content

Silverlight Examples

Why Silverlight?

- Development Community
  - Forums
  - Tutorials
  - Guides
  - etc.
- Productivity Tools
  - Code editing, Auto completion, Refactoring
  - Visual Designers
  - etc.

Tools of the Trade

- Visual Studio
  - Targeted at developers
  - Fully featured Integrated Development Environment facilitates both coding and debugging
- Expression Blend
  - Targeted at designers, but still great even for developers
  - Allows editing of UI elements with immediate feedback

Getting Started with Silverlight 3

Easier than you may think

Install & Instantiate

- Install
  - One-time (system-wide)
  - Auto-updates
  - Fully customizable
  - No browser restart
- Instantiation
  - <OBJECT TAG>
    - Easier to customize
  - Silverlight.js
    - Scriptable control (http://code.msdn.microsoft.com/silverlightjs)

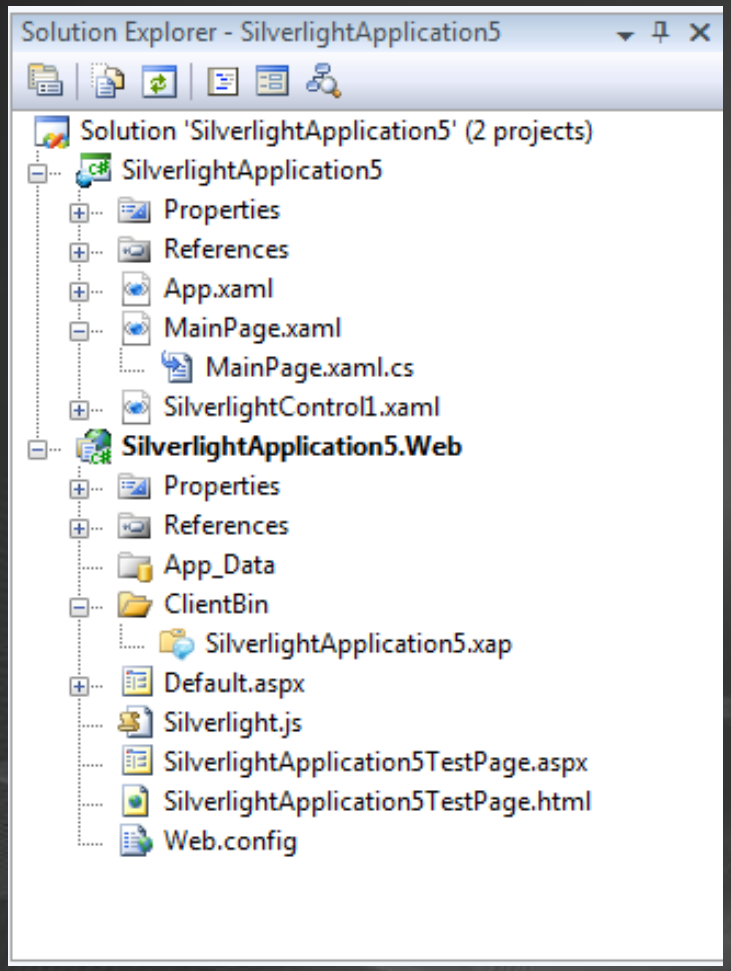
Good news

- Visual Studio does it all for you
  - Just create a Silverlight Project
- Let’s go create one…

Anatomy of a Silverlight Project

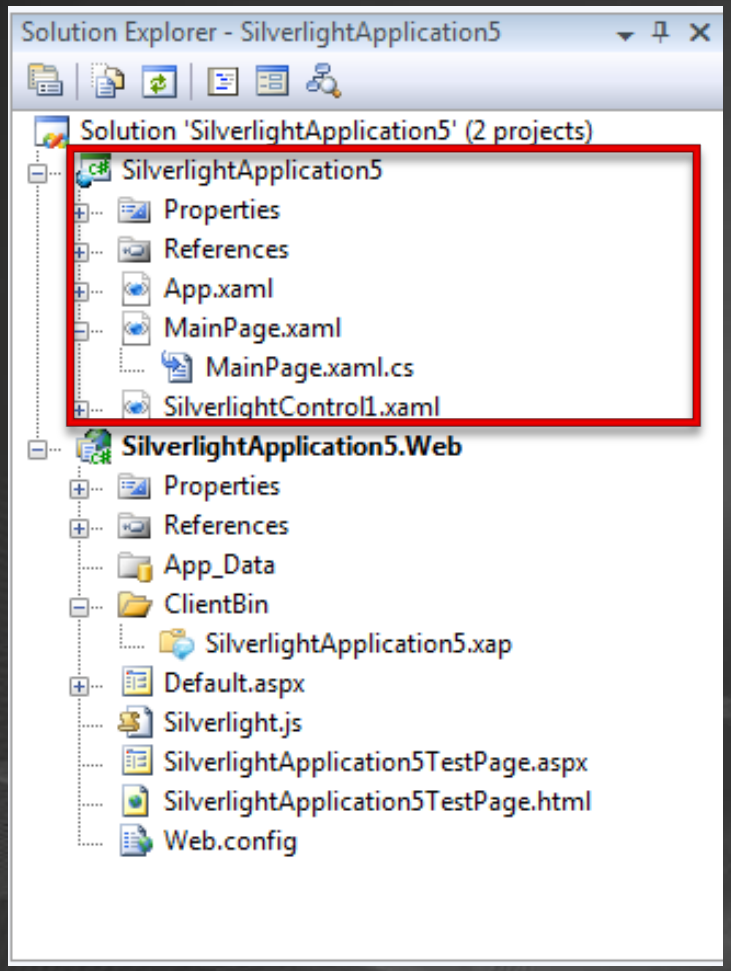
- Silverlight Application Project
  - Contains code, XAML, and resources that will be compiled into application
  - Compilation generates .xap file (ZIP file of compiled binaries, XAML, and resources)
- Web Host Project
  - Contains web site host for application with everything necessary to run your application
  - Running it will open up a page with your application

Silverlight Application Solution



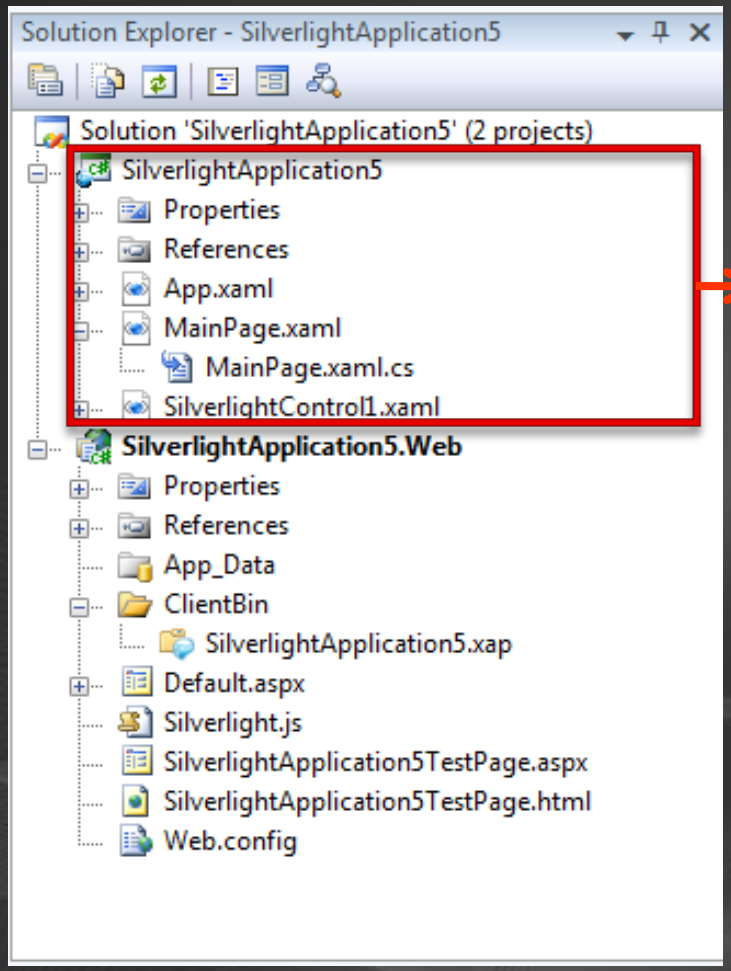
Microsoft Silverlight

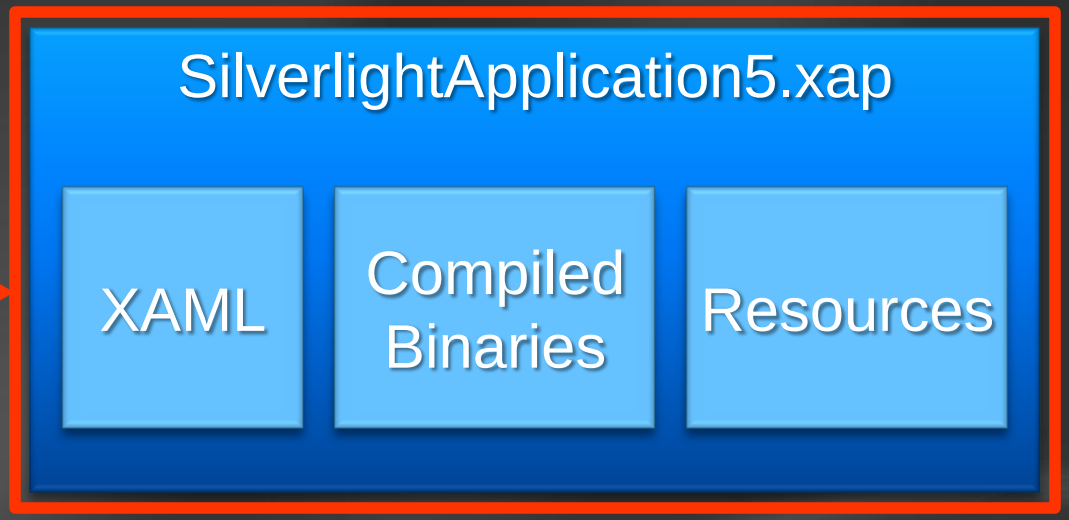
Compilation



Microsoft Silverlight

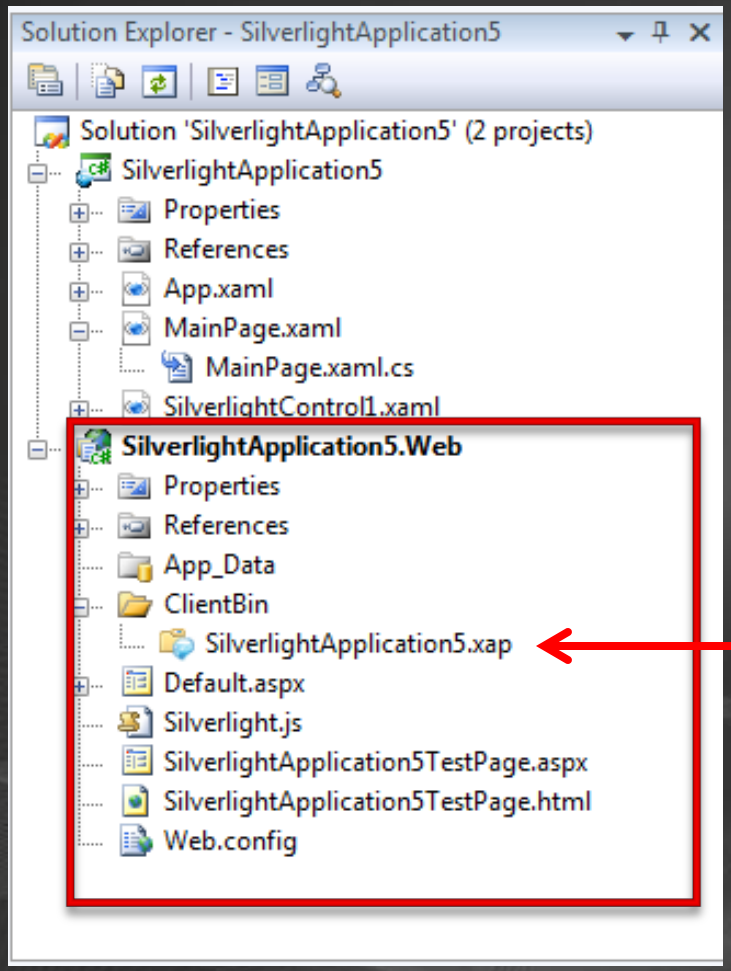
Compilation

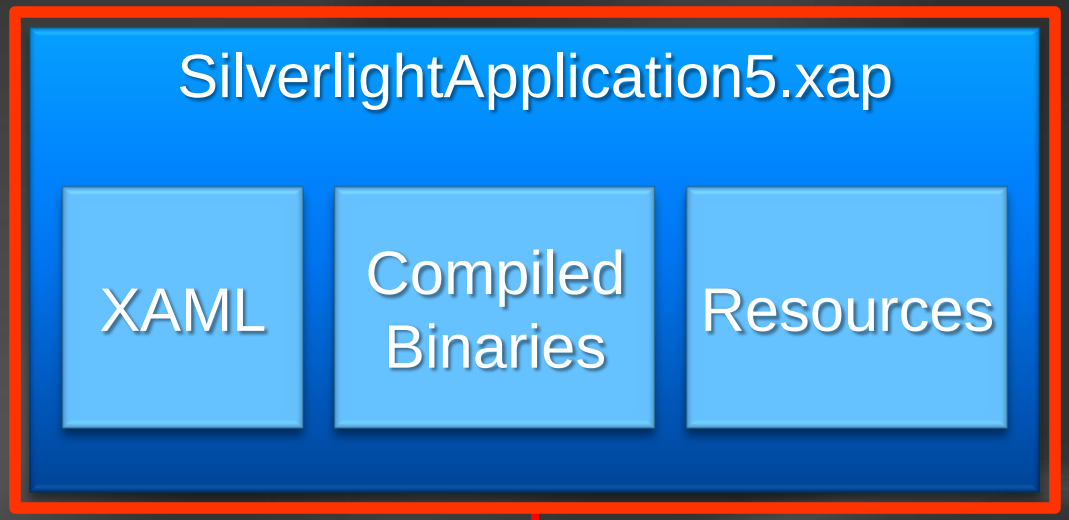




Microsoft Silverlight

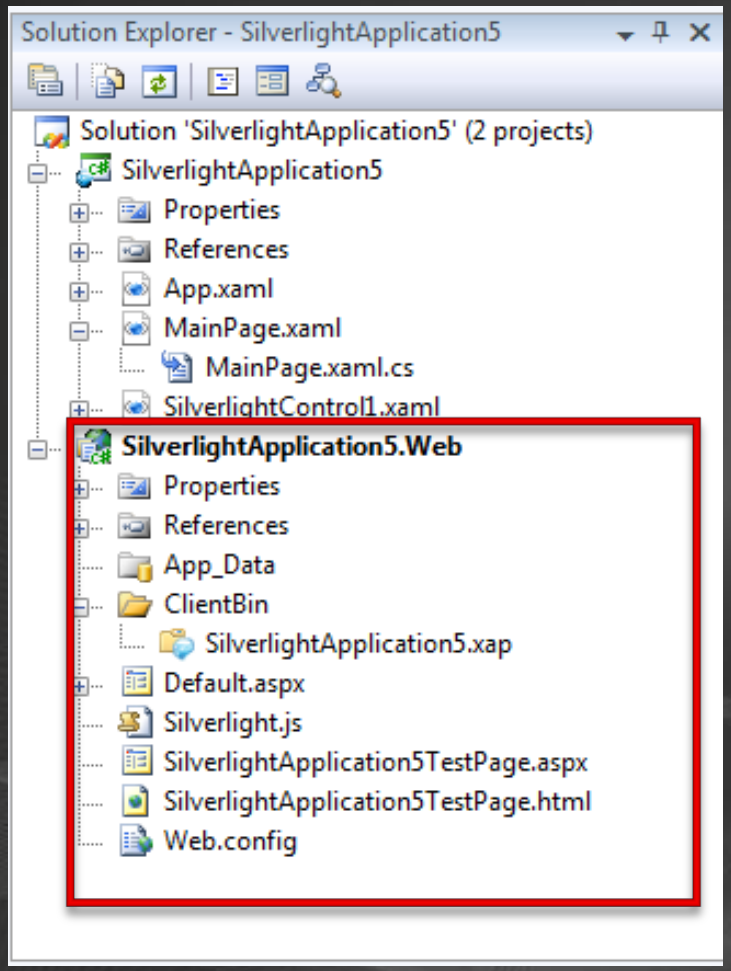
Compilation





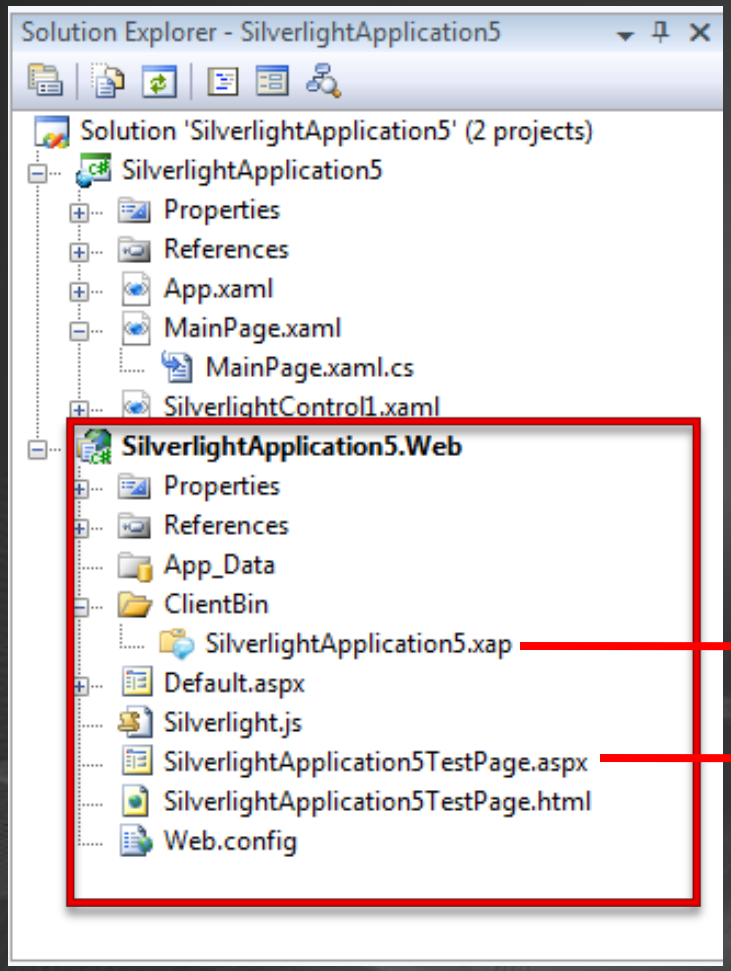
Microsoft Silverlight

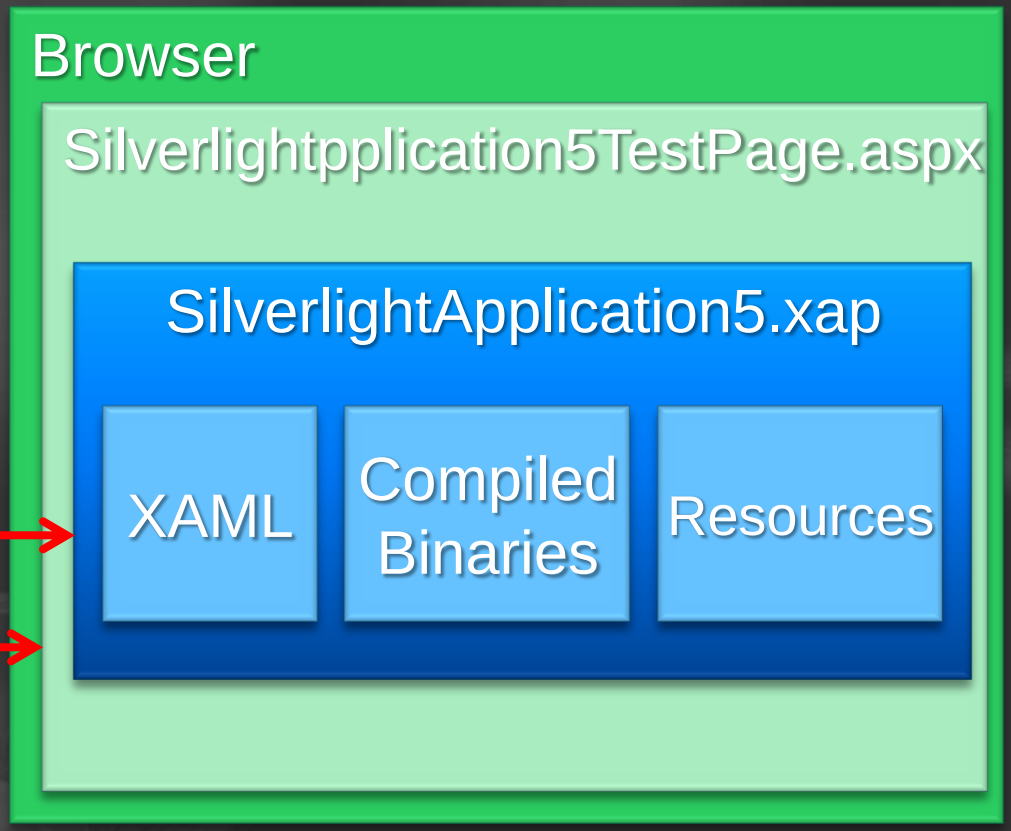
Compilation



Microsoft Silverlight

Running

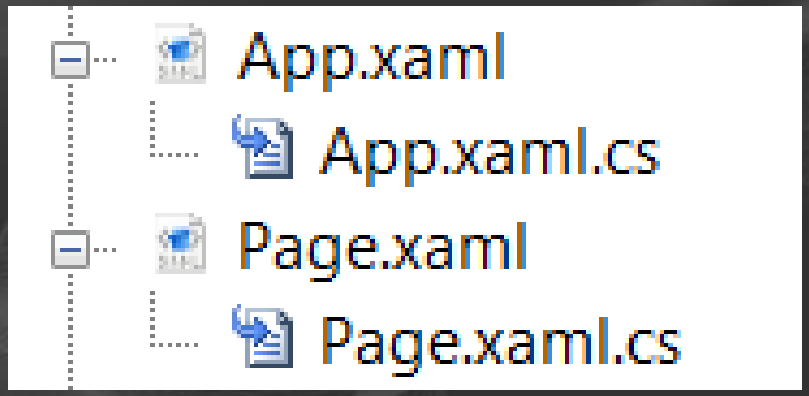




Microsoft Silverlight

XAML and Code-behinds

- XAML
  - User Interface Definition
- Code-Behind(C# or Visual Basic)
  - Logic for manipulating user interface
  - Event handlers



Microsoft Silverlight

XAML

- Extensible Application Markup Language
  - Declarative syntax
  - Separation of logic & user interface
- Any thing you can do from XAML you can do from code
  - Various things are simpler/cleaner to do in XAML

```
<Button Height="48"
Width="104"
Content="Button"/>
```

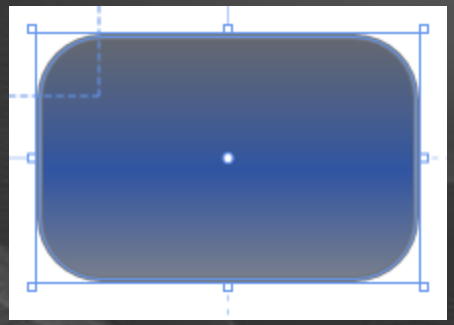
=

```
Button b = new Button ();
b.Height = 48 ;
b.Width = 104;
b.Content = “Button”
```

Microsoft Silverlight

XAML (toolable)

```
<Rectangle Height="124" Width="191" Stroke="#FF74767B" RadiusY="31" RadiusX="31" StrokeThickness="3">
    <Rectangle.Fill>
      <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
          <GradientStop Color="#FF7A7F8D" Offset="1"/>
          <GradientStop Color="#FF3155A3" Offset="0.548"/>
          <GradientStop Color="#FF46547A" Offset="0"/>
       </LinearGradientBrush>
     </Rectangle.Fill>
  </Rectangle>
```



Microsoft Silverlight

XAML → Objects

```
<UserControl x:Class="SilverlightApplication2.Page"
   xmlns="…"   xmlns:x=".. " Width="400" Height="300">
   <Grid x:Name="LayoutRoot" Background="White">
   </Grid>
</UserControl>
```

Design-time

```
public partial class Page :
System.Windows.Controls.UserControl {

      internal System.Windows.Controls.Grid LayoutRoot;
```

Compile-time

```
public void InitializeComponent()
{
   LoadFrom(this, Page.xaml); //Pseudocode
   LayoutRoot = this.FindName("LayoutRoot"); //Pseudocode
}
```

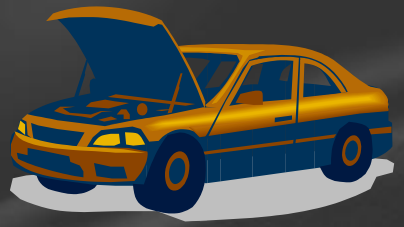
Run-time

Microsoft Silverlight

XAML / Code-Behind

demo

Silverlight 3 Fundamentals



Microsoft Silverlight

Dependency Properties

- Special properties attached to an instance of an object (the owner)
- Declared as public static field on type that owns it

```
public void SetCanvasWidthTwice(double width)
{
    MyCanvas.SetValue(Canvas.WidthProperty, width);
    MyCanvas.Width = width;
}
```

Dependency Properties Uses

- Animation
- Bindings
- Styling
- Property Inheritance
- Attached properties

DP value precedence

- Active animations, or animations with a Hold behavior.
- Local value.
- TemplatedParent template properties.
- Style setters.
- Default value. Any given dependency property may have a default value

Attached Dependency Properties

- Special type of property
- Decouples Owner/Consumer from object setting it

```
<Canvas x:Name="MyCanvas" Background="White">
    <TextBlock x:Name="TextBlock1"
        Canvas.Top="30"
        Canvas.Left="20"
        Text="My Location is x = 20, y = 30" />
</Canvas>
```

```
public void SetTextBlockDistanceFromTop(double top)
{
    TextBlock1.SetValue(Canvas.TopProperty, top);
}
```

Dynamic Layout

- Global Layout properties
  - Width, MinWidth, MaxWidth, ActualWidth
  - Height, MinHeight, MaxHeight, ActualHeight
  - Margin and Padding
- Panels
  - Canvas (Top, Left Attached Properties)
  - Grid (Row, Column Attached Properties)
  - StackPanel (Horizontal or Vertical Orientation)
  - Border

Layout

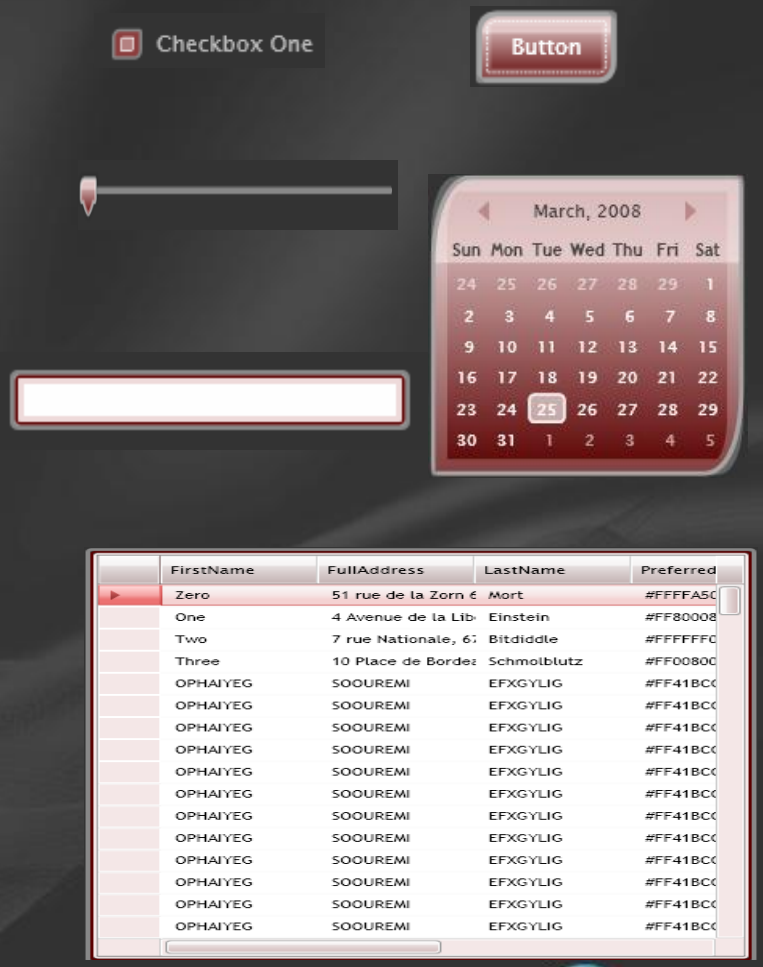
demo

Controls in Silverlight 3

Building Blocks of A Silverlight Application

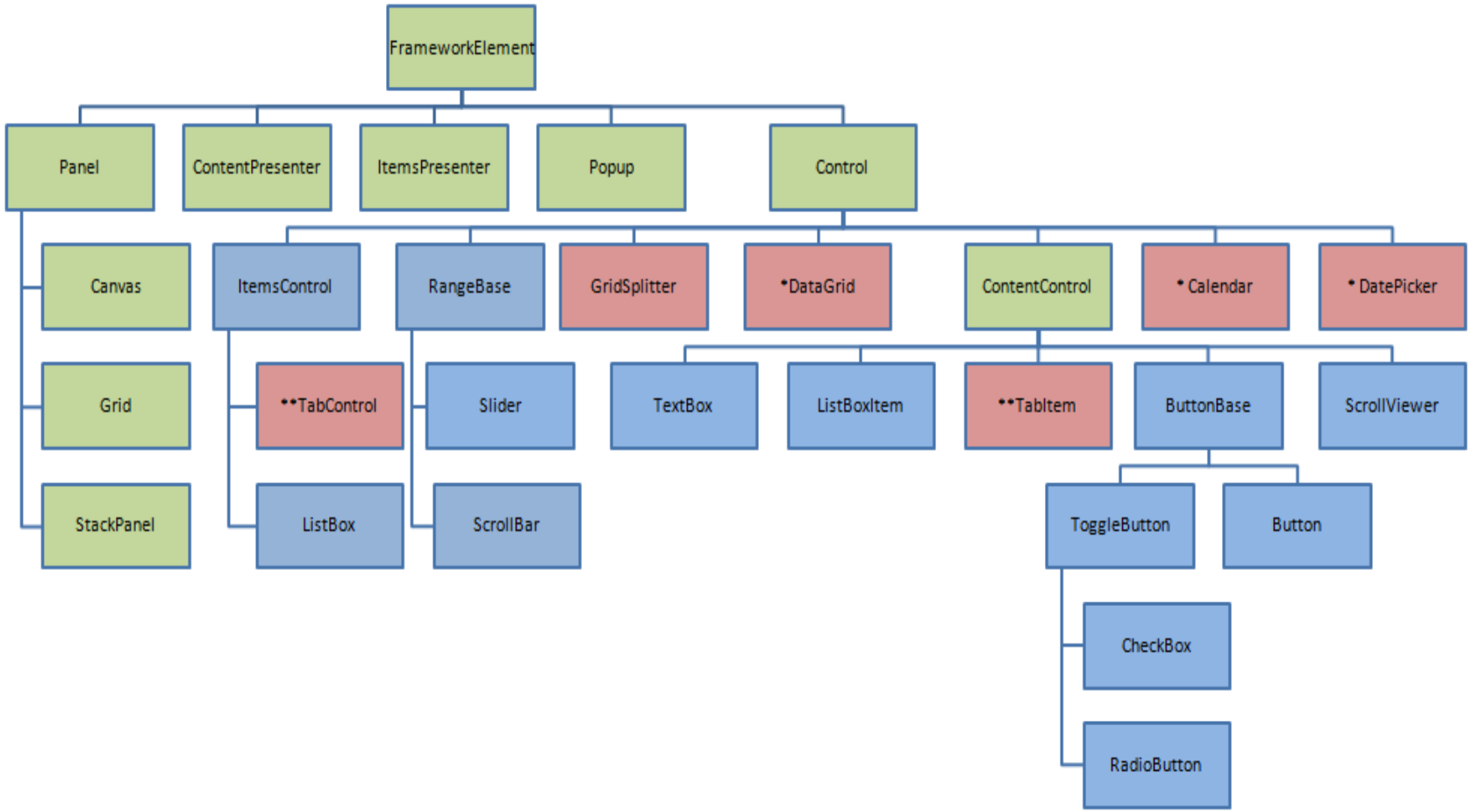
Controls

- Microsoft ships the usual suspects
  - With source
  - Modeled after WPF
  - Extensible
- Silverlight toolkit adds to this set
- Rich partner ecosystem for domain specific controls



Microsoft Silverlight

Basic Controls



Microsoft Silverlight

Controls

Quick look

Microsoft Silverlight

Hierarchy

- Used for more than laying out visuals
  - Resources
  - Bubbling Events
  - Property Value Inheritance

Resource Dictionaries

- Resource Dictionaries are collections of reusable, shared objects accessible within a specific scope (self and children nodes)
- Each item in a resource dictionary must define a key through the x:Key attribute
- Referencing a resource:
  - Background="{StaticResource *BackgroundColorKey*}"

Resource Dictionaries

- Each resource is accessible within a certain scope
  - Element Resources - scope is itself and children
  - App Resources – scope is the whole application. Defined in App.xaml.
- Resource Dictionaries can be defined in external files and merged in.

Resources In Action

demo

Bubbling Events

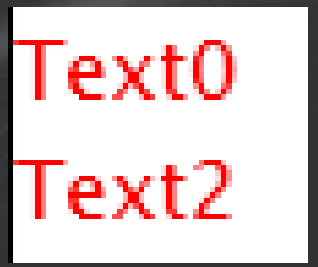
- Events that are routed (or forwarded) up the visual tree until
  - It reach ‘root’ of the tree or
  - e.Handled = true ;

KeyDown, KeyUp,
MouseLeftButtonDown, MouseLeftButtonUp, MouseMove
BindingValidationError

Property Value inheritance

- When a Dependency Property inherits its value from a parent in the Visual Tree
  - Foreground
  - Control.Font*
  - Data Context

```
<ContentControl Foreground="Red" Height="49"  >
    <StackPanel HorizontalAlignment="Left"   >
            <TextBlock Text="Text0"  />
            <TextBlock Text="Text2"  />
     </StackPanel>
</ContentControl>
```



Databinding

- Allows updates to properties when bound property changes
- Two ends:
  - Source property – property referenced by the binding (can be generic property of object, but need to implement INotifyPropertyChanged)
  - Target property – property on which binding is declared (must be dependency property)
- Syntax: Text="{Binding Path=Title}"

Databinding

- If no source object is specified, DataContext is considered the source object.
- Two options:
  - Bind to property with the same type
  - Bind to property of different type and specify IValueConverter
- Modes: OneTime, OneWay (default), TwoWay

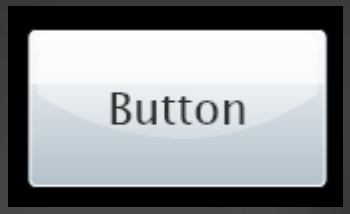
Databinding

demo

Control customization & extensibility

- Styling
  - Set existing properties
  - Centralizes change
  - Maximizes reuse
- Templating/Skinning
  - Define new parts
  - Creative freedom
  - Maximum Visual customization without code

Control Customization



Styling

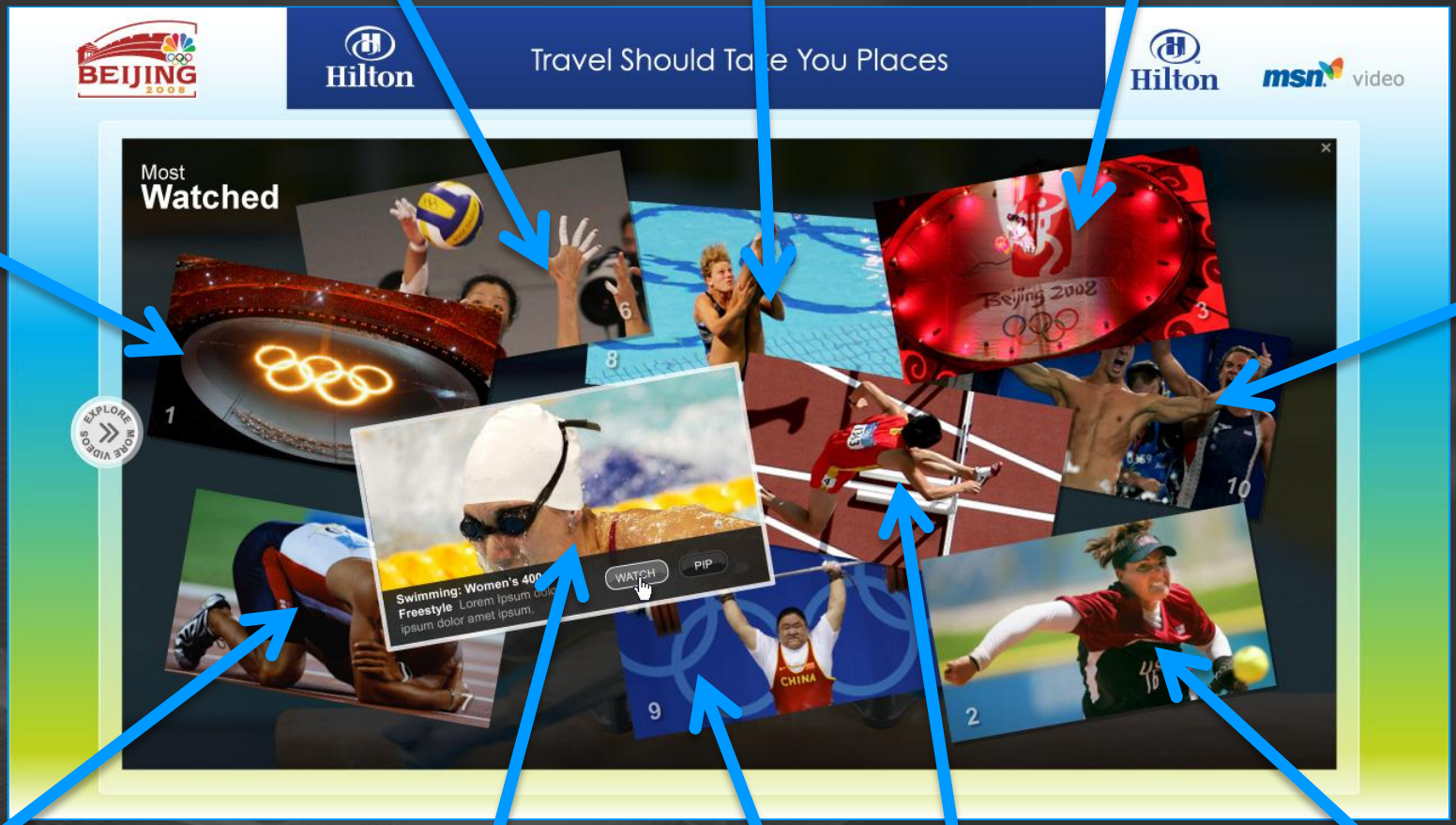
- Set existing properties
- Centralizes change
- Maximizes reuse

Templating

- Define parts
- Creative freedom
- Visual customization without code

Microsoft Silverlight

Motivation for UserControl



Silverlight

UserControl

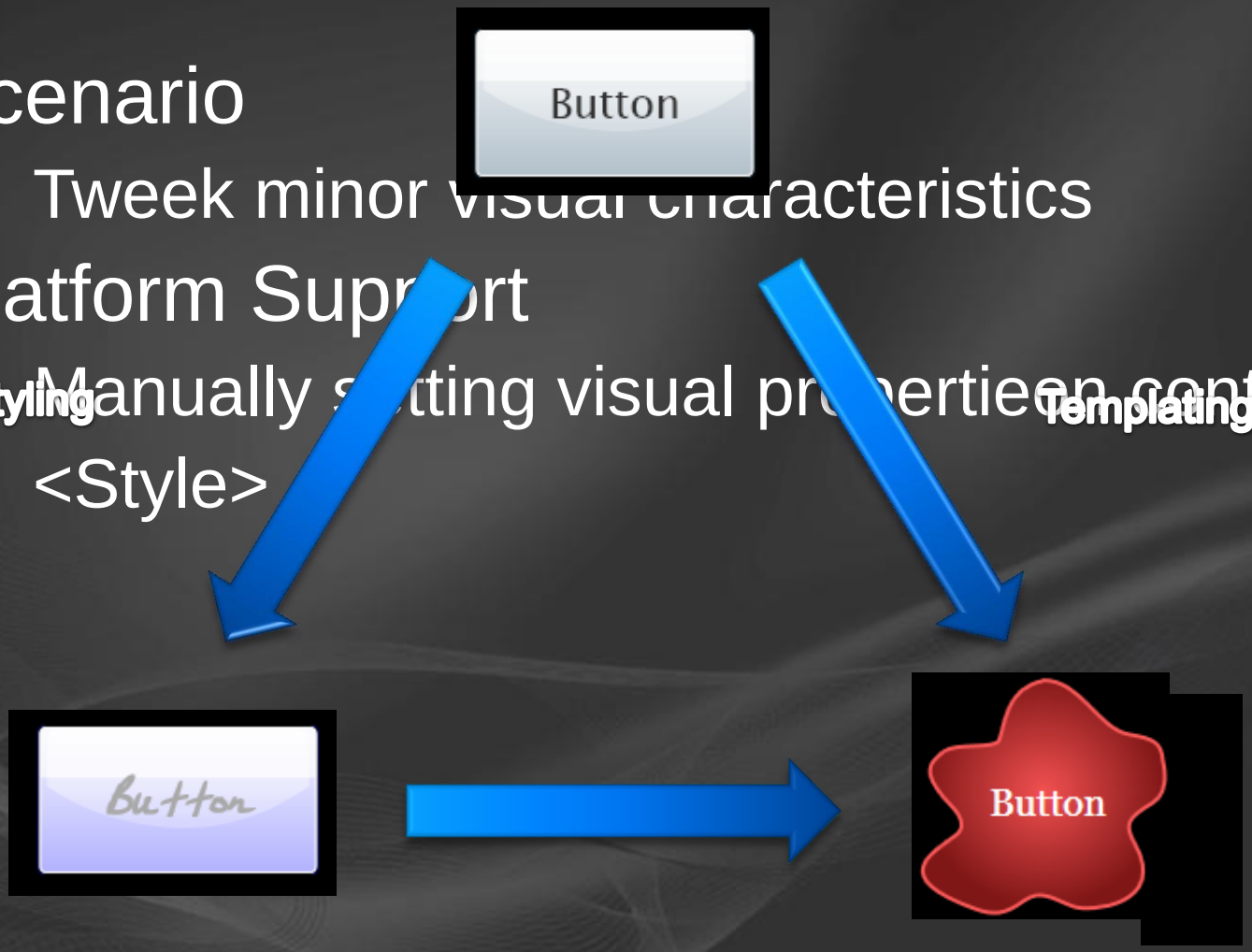
- Scenario
  - Fixed look with some logic
  - Split large page into smaller chunks
  - Reuse xaml/logic in multiple places
- Platform Support
  - <UserControl x:Class="MyControl"> … </>
  - public partial class MyControl : UserControl {}
  - XAML is optional

UserControl Trivia

- A Silverlight “page” is a UserControl
  - Visual element container
  - Databinding
  - Mouse & keyboard events
  - … and so on

Customizing Existing Controls

- Scenario
  - Tweek minor visual characteristics
- Platform Support
  - Manually setting visual properties on control
  - <Style>



The <Style> and <Setter> tags

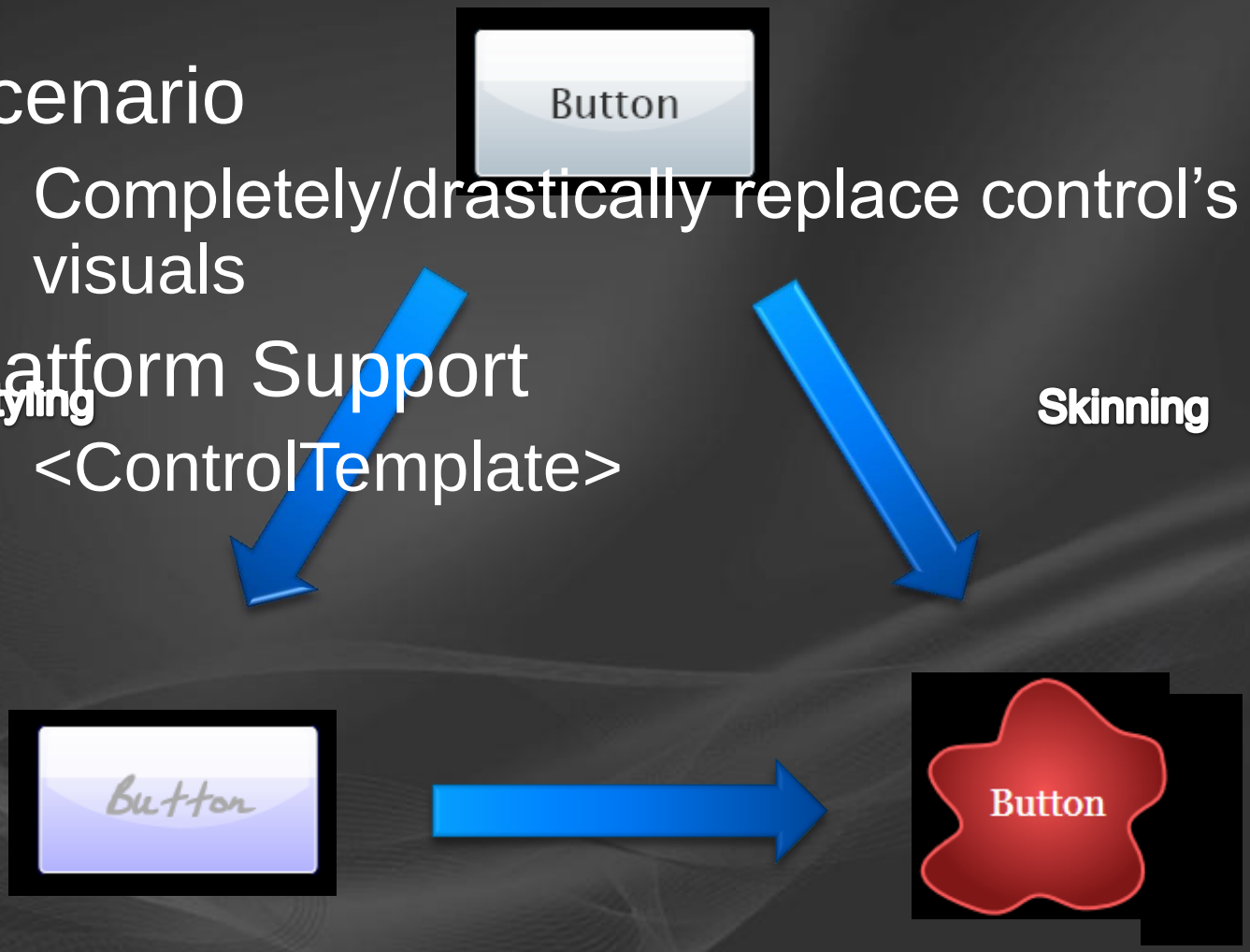
- <Style> sets properties through a sequence of child <Setter> tags
- Each <Setter> has two important attributes:
  - Property – indicates the name of the property to set
  - Value – the value to give to the property.

Property Bag “Styling”

- <Style>
  - A set of properties that can be applied to controls, text, shapes, …
- Silverlight 3
  - “Write once” behavior
  - Application theme styles not supported
  - BasedOn styles supported

"Skinning" Existing Controls

- Scenario
  - Completely/drastically replace control's visuals
- Platform Support
  - <ControlTemplate>



Microsoft Silverlight

Skinning

- <ControlTemplate>
  - A set of elements that make up the visual structure of a control.
- {TemplateBinding PropertyName}
  - A link between the template visuals and the control’s visual properties

Styling and Skinning

demo

Using <ControlTemplate>

Application Model

- Threading
- Security
- Error Handling
- HTML/Javascript Interop

Threading

- One UI Thread
- <many> background threads
- UI elements can only be accessed from UI Thread
- Background threads access UI thread via Dispatcher object (every UI element has one)

```
void OnBackgroundCallBack( … )
{
    ObjectDelegate cb = delegate( object )  {}
    myUIElement.Dispatcher.BeginInvoke( cb , (object) args.Result );
}
```

Microsoft Silverlight

Security

- Secure/Extended sandbox
  - Cross-domain
  - Isolated Storage
  - OpenFileDialog
- One level of trust
  - No elevation
- Multiple plugin instances on a page
  - Communication using HTML Browser interop

HTTP Post/Get

```
public void GetAndPost()
{
    string uploadData = "hello world";
    WebClient client = new WebClient();
    client.UploadStringCompleted += OnUploadFinished;
    client.UploadStringAsync(new Uri("uploadUrl"), "POST", uploadData);

    client.DownloadStringCompleted += OnDownloadFinished;
    client.DownloadStringAsync(new Uri("downloadUrl"));
}
```

- Files used for cross domain access:
  - clientaccesspolicy.xml
  - crossdomain.xml

HTML/JavaScript Interop

- Getting Current Window and Document
  - System.Windows.Browser.HtmlPage.Window
  - System.Windows.Browser.HtmlPage.Document
- Traversing DOM
  - On Document, call GetElementById(string)
  - Or walk DOM by calling Children on elements inside Document.Body
- Calling JavaScript
  - Call Invoke on ScriptObject

Error handling

Exception is thrown

ImageFailed is fired

Application_UnhandledException

ImageFailed Handler

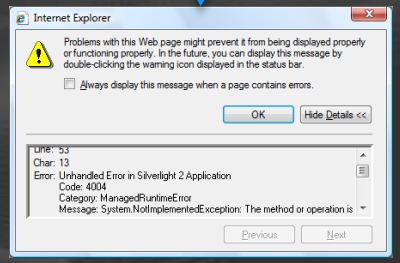
e.Handled = true;

e.Handled = false;

e.Handled = true;

e.Handled = false;

Plug-in's OnError



Microsoft Silverlight

Getting the tools

1. Go to www.dreamspark.com and sign up.
   1. Download Visual Studio 2008 Professional
   2. Download Expression Blend 3
2. Silverlight Tools for VS2008
   - http://www.microsoft.com/downloads/details.aspx?familyid=9442B0F2-7465-417A-88F3-5E7B5409E9DD&displaylang=en
3. Download Silverlight Toolkit
   - http://www.codeplex.com/Silverlight

Microsoft Silverlight

Useful Links

- Silverlight Home Page
  - http://www.silverlight.net
- Silverlight Page at MSDN
  - http://msdn.microsoft.com/en-us/library/cc838158(VS.95).aspx
- Silverlight Community Blogs
  - http://blogs.silverlight.net/blogs/community/
- Jesse Liberty's Blog
  - http://blogs.silverlight.net/blogs/jesseliberty/

Go-Go Gadget Competition

- Build miniature Windows 7 Gadgets
- Starts today (One week long)
- Lots of cool prizes ($36,000 worth)
- Website: http://win7.mit.edu
- All you need to know is HTML, CSS, Javascript

Questions?

Microsoft®

Your potential. Our passion.™

Silverlight