

Silverlight For Windows Embedded in **Windows Embedded Compact 7**

Douglas Boling
Boling Consulting Inc.

About Douglas Boling

- Independent consultant specializing in Windows Mobile and Windows Embedded Compact (Windows CE)
 - On-Site Instruction
 - Consulting and Development
- Author
 - *Programming Embedded Windows CE*
 - *Fourth Edition*

Agenda

- Basic Silverlight for Embedded
- Fielding Events
- Creating Storyboards
- Bringing it together

Disclaimer...

- I'm not a designer!
 - I'm a geek.
- This geek is artistically challenged
 - Don't expect too much

User Interface Evolution

- In the beginning...
 - There was the command line
 - Perhaps a 10 key keyboard with LED display
- GUI Operating Systems
 - Windows/Mac/Linux w/ a front end
- Browser-based U/I
 - It's like the Internet, it must be great!

Next Wave of U/I

- Microsoft Silverlight
- “Web-like” descriptive U/I language
 - XAML → Machine and human (mostly) readable
- Real code backing for functionality
- Separates Designers from Developers

Designers vs. Developers

- Designers are great at UI
 - Also great at designing a UI that's impossible to build!
- Developers Spend countless hours implementing a design
 - When they should be implementing business logic

Silverlight

- Original impression: Powerful platform for delivering web-based content
- Actually, a powerful platform for developing platform independent applications
- Combines the powerful U/I constructs of WPF with the light weight of a much smaller .NET runtime
- Still, it's .NET-based runtime not perfect for embedded systems...

Silverlight for Embedded

- XAML-based interface description
 - Silverlight 3 for Windows Embedded Compact
 - Silverlight 2 for Windows CE 6 Release 3
- Backed by native C++ code
 - High performance
 - Full Win32 API, and hardware access
 - No ‘garbage collect pause’
- Support for Hardware-based graphics acceleration

Silverlight vs. Silverlight for Embedded

Silverlight

- XAML based U/I
- Managed code behind
 - C#
 - BCL framework
- Runs in sandbox
- Tools
 - Expression Blend
 - Visual Studio

Silverlight For Embedded

- XAML based U/I
- Native code behind
 - C++
 - Win32 API
- Runs natively on OS
- Tools
 - Expression Blend
 - Visual Studio

Silverlight For Embedded Versions

- Windows Embedded CE 6 Release 3
 - Supports
 - Silverlight 2
 - Manual porting of XAML to C++ application
- Windows Embedded Compact 7
 - Supports
 - Silverlight 3
 - Automated porting of XAML to C++ application

Tools in Windows Embedded Compact 7

- Expression Blend 3
 - Select Silverlight 3 application
- Silverlight for Embedded SDK
 - Provides a project template for Silverlight for Embedded
- Platform Builder 7
 - Contains Migration tool for converting Silverlight project to native C++ project

Silverlight Development Machine Preparation

- Expression Blend 3 provided as part of Platform Builder 7
 - Evaluation version expires after 30 days
- Be sure to install Expression Blend 3 before PB
 - This allows installer to find Blend and add SWE App templates
 - If you forget, you will need to reinstall Silverlight SDK after Blend installed

Basic Development Process

- Create a project with Expression Blend 3
 - Create basic U/I
 - Name any elements you wish to access
 - Add any event handlers
- In PB, use Tools | Windows Embedded Silverlight Tools...
 - Create Platform Builder Project
 - Add created project as subproject using “Add Existing...” menu item
- Add C++ code behind
- Launch application and debug...

Initial XAML

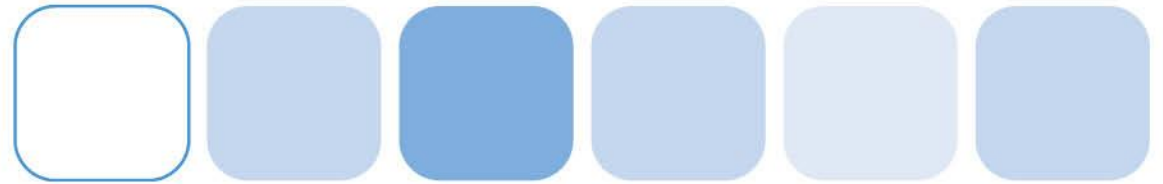
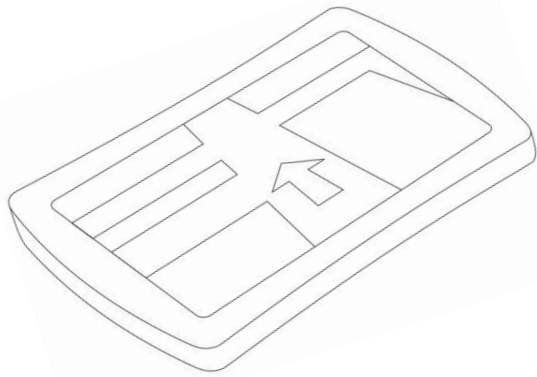
```
<UserControl>  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    x:Class="WindowsEmbeddedSilverlightApplication1.MainPage"  
    Width="640" Height="480">  
  
    <Grid x:Name="LayoutRoot" Background="White"/>  
</UserControl>
```

Native Code Framework

- PB tool creates a basic framework
- App.cpp Startup/Exit callbacks, Framework init
- App.h Implementation of basic app functionality
- MainPage.cpp Where event callbacks are placed
- MainPage.h Support for MainPage.cpp
- Other .cpp and .h files as needed for additional forms

Initial MainPage.cpp

```
// =====  
// OnLoaded  
// Description: Calls InitializeComponent to bind member variables to named  
//              elements, and attach event handlers specified in XAML  
//  
// Parameters:  pRoot - The root dependency object.  
// =====  
HRESULT MainPage::OnLoaded(__in IXRDependencyObject* pRoot)  
{  
    UNREFERENCED_PARAMETER(pRoot);  
    HRESULT hr = InitializeComponent();  
  
    // Add calls to FindName or Add__EventHandler() methods after this comment.  
    return hr;  
} // OnLoaded
```



Demo: Basic Development Process

Fielding Events

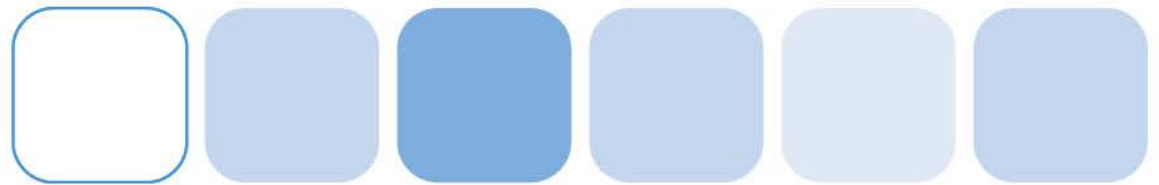
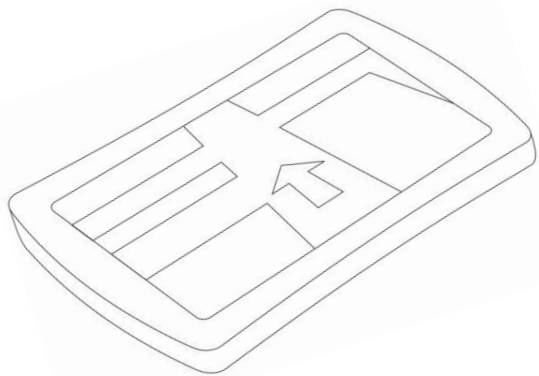
- Add controls in XAML that generate events
- Add x:Name attribute to identify
- Add dummy event handler in Expression
- Run PB Silverlight Migration tool to update native project
- Update native project to act on event

Fielding Events XAML

```
<UserControl>
    ...
    <Grid x:Name="LayoutRoot" Background="White">
        <Button x:Name="MyButton" Content="Click Me!"
            Margin="250 200" Click="MyButton_Click"/>
    </Grid>
</UserControl>
```

Fielding Events Code

```
// =====  
// MyButton_Click  
// =====  
HRESULT MainPage::MyButton_Click (IXRDependencyObject* pSender,  
                                   XRMouseButtonEventArgs* pArgs)  
{  
    HRESULT hr = E_NOTIMPL;  
  
    if ((NULL == pSender) || (NULL == pArgs))  
    {  
        hr = E_INVALIDARG;  
    }  
    return hr;  
}
```



Demo: Adding Events

Creating Storyboards

- Storyboards are animations of objects
 - Done by defining object properties at specific times
- Easily created with Expression Blend
 - Not so easy to do manually
- Storyboards are objects
 - Call FindName to get pointer to object
 - IXRStoryboard
 - Call Begin() method to play

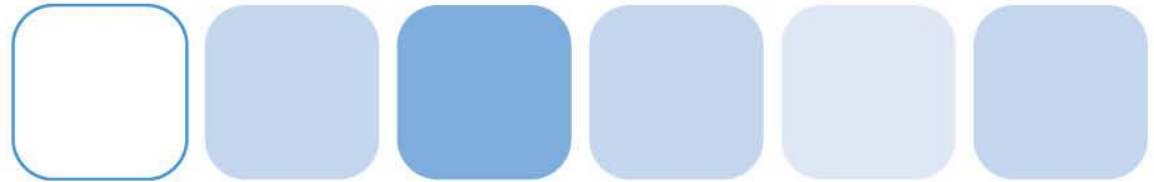
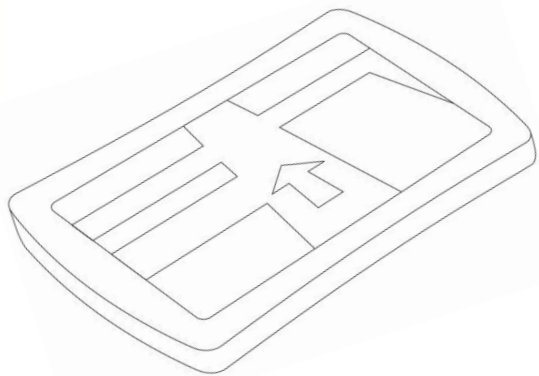
Playing A Storyboard (XAML)

```
<Storyboard x:Name= Storyboard1">  
  <DoubleAnimationUsingKeyFrames BeginTime= 00:00:00"  
    Storyboard.TargetName= btnLeft  
    Storyboard.TargetProperty="(UIElement.Opacity)">  
  
    <SplineDoubleKeyFrame KeyTime= 00:00:00.5000000  
      Value="0"/>  
  </DoubleAnimationUsingKeyFrames>  
</Storyboard>
```


Playing A Storyboard (Code)

```
//  
// Play a storyboard  
//  
IXRStoryboard *pSB;  
pRoot->FindName (TEXT("Storyboard1"),  
                (IXRDependencyObject**)&pSB);  
  
if (pSB != NULL)  
{  
    pSB->Stop();  
    pSB->Begin();  
  
    pSB->Release();  
}
```

- Demo



Playing A Storyboard

Bringing It Together

- All the following techniques enable a great user interface
- The key is that you can still use Win32 APIs directly
- This makes Silverlight for Embedded a great tool for embedded applications

Summary

- Silverlight for Embedded changes everything
- U/I now done by designer without creating implementation pain for the developer
- Powerful flexibility makes up for somewhat tedious hookup

Questions...

Doug Boling

Boling Consulting Inc.

www.bolingconsulting.com

[dboling @ bolingconsulting.com](mailto:dboling@bolingconsulting.com)

Microsoft[®]