



WPF(.NET3.5)

vs Silverlight 2.0 Beta1

中 博俊@わんくま同盟



アジェンダ

- Silverlight2.0までの
おさらい
- 細かい違いを見よう
- まとめ

Silverlight2.0までのおさらい



Silverlightとは

- 1.0が2007/09/04リリース
 - <http://www.microsoft.com/presspass/press/2007/sep07/09-04SilverlightPR.msp>
- Flashキラー？
- XAML(サブセット)で見た目を作成する
- OSがWindowsとMacintosh
- ブラウザがIE, Safari, Firefox, Mozilla

Silverlightとは

- Silverlight Mobile (Nokia) が発表されました。
 - <http://www.nokia.com/A4136001?newsid=1197788>
 - 日本上陸は？

2.0の特徴

- MiniCLRが搭載されている
- Xamlのデータモデルが結構変更されている。
- 日本語対応

2.0って？

- 2.0は今年中リリース予定？

- <http://blogs.msdn.com/ashish/archive/2008/04/03/silverlight-roadmap-questions.aspx>

- We are targeting late Summer
- Here is a rough timeline:
 - Silverlight 2.0 Beta 1 (Q1CY08 with limited (non commercial) Go-Live)
 - Silverlight 2.0 Beta 2 (Q2CY08 with Go-Live)
 - Silverlight 2.0 RTM (Summer 2008) – Exact timing TBD
- Silverlight v.next – We are working on a v.Next plan and have nothing to announce at this time
- Silverlight for mobile – No date available

2.0 Betaを試す環境

- Silverlight 2.0 SDK Beta1
 - <http://www.microsoft.com/downloads/details.aspx?FamilyId=4E03409A-77F3-413F-B108-1243C243C4FE&displaylang=en>
- Microsoft® Silverlight™ 2 Software Development Kit Beta 1 Documentation
 - <http://www.microsoft.com/downloads/details.aspx?FamilyId=1840CAB5-196C-4264-B55D-562242A72625&displaylang=en>
- Tools Beta1
 - <http://www.microsoft.com/downloads/details.aspx?displaylang=ja&FamilyID=e0bae58e-9c0b-4090-a1db-f134d9f095fd>
- Blend 2.5 March 2008 Preview
 - <http://www.microsoft.com/downloads/details.aspx?FamilyId=32A3E916-E681-4955-BC9F-CFBA49273C7C&displaylang=en>

細かい違いをみてみよう



起動ルーチン

- WPFのデフォルトはStartupUri

```
<Application x:Class="WpfApplication"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  StartupUri="Window1.xaml">
  <Application.Resources>
```

WPF

- ほかにApplication.Startイベントを利用する。

起動ルーチン

- SilverlightのデフォルトはStartupイベント

```
public App()  
{  
    this.Startup += this.Application_Startup;  
    this.Exit += this.Application_Exit;  
    this.UnhandledException += this.Application_UnhandledException;  
  
    InitializeComponent();  
}  
  
private void Application_Startup(object sender, StartupEventArgs e)  
{  
    // メインコントロールを読み込みます。  
    this.RootVisual = new Page();  
}
```

- RootVisualにPageを入れています。

Silverlight



Application

The screenshot displays the 'Application Class' hierarchy in Visual Studio. It is divided into three main sections:

- Application Class (Left):** A tree view showing the class structure. The 'Application Methods' folder is expanded to show 'GetResourceStream Method', 'LoadComponent Method', and 'Application Properties'. The 'Application Properties' folder is also expanded to show 'Current Property', 'Host Property', 'Resources Property', and 'RootVisual Property'. The 'Application Events' folder is expanded to show 'Exit Event', 'Startup Event', and 'UnhandledException Event'.
- Application メソッド (Middle):** A list of methods. The 'GetContentStream メソッド' is highlighted in blue. Other methods include FindResource, GetCookie, GetRemoteStream, GetResourceStream, LoadComponent, OnActivated, OnDeactivated, OnExit, OnFragmentNavigation, OnLoadCompleted, OnNavigated, OnNavigating, OnNavigationFailed, OnNavigationProgress, OnNavigationStopped, OnSessionEnding, OnStartup, Run, SetCookie, Shutdown, and TryFindResource.
- Application プロパティ (Right):** A list of properties. The 'Application プロパティ' folder is expanded to show 'Current プロパティ', 'MainWindow プロパティ', 'Properties プロパティ', 'ResourceAssembly プロパティ', 'Resources プロパティ', 'ShutdownMode プロパティ', 'StartupUri プロパティ', and 'Windows プロパティ'. The 'Application イベント' folder is expanded to show 'Activated イベント', 'Deactivated イベント', 'DispatcherUnhandledException イベント', 'Exit イベント', 'FragmentNavigation イベント', 'LoadCompleted イベント', 'Navigated イベント', 'Navigating イベント', 'NavigationFailed イベント', 'NavigationProgress イベント', 'NavigationStopped イベント', 'SessionEnding イベント', and 'Startup イベント'.

WPF

Silverlight

- 必要最小限に絞り込まれています



Application

- 終了させる方法は？

```
Application.Current.MainWindow.Close();
```

WPF

- たぶんSilverlightはない

Silverlight

- Flashっぽいアプリケーションなら頭から再生を用意しよう。
- 業務系ならあっさり<TextBlock Text=“終了しました”>でいいかもしれない。
- Application.Exitイベントが走るのはブラウザが閉じられるか、別ページに行ったとき
- Webのお約束で、実行を期待しないこと。

最初の画面、画面遷移

- WPFはWindow作成

```
<Window x:Class="WpfApplication1.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="300">
  <Grid>

  </Grid>
</Window>
```

WPF

- SilverlightはすべてUserControl

```
<UserControl x:Class="SilverlightApplication1.Page"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Width="400" Height="300">
  <Grid x:Name="LayoutRoot" Background="White">

  </Grid>
</UserControl>
```

Silverlight



最初の画面、画面遷移

- 画面遷移などは？

```
private void clk(object sender, RoutedEventArgs e)
{
    Hide();
    new Window2().ShowDialog();
    Show();
}
```

WPF

- 今までのWindowsFormと特に変わらない

最初の画面、画面遷移

- Windowがない
- 画面遷移などは？

```
private void clk(object sender, RoutedEventArgs e)
{
    var visual = (Page)Application.Current.RootVisual;
    visual.LayoutRoot.Children.Clear();
    visual.LayoutRoot.Children.Add(new Page2());
}
```

```
<UserControl x:Class="SilverlightApplication1.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="400" Height="300">
    <Grid x:Name="LayoutRoot" Background="White">
        <Button Click="clk" />
    </Grid>
</UserControl>
```

Silverlight

最初の画面、画面遷移

DEMO1



わんくま同盟 大阪勉強会#19

メッセージボックス

- いつものメッセージボックス **WPF**

```
if ( MessageBox.Show("登録しますか?", "確認",  
    MessageBoxButton.YesNo, MessageBoxImage.Question )  
    == MessageBoxResult.Yes )  
{  
    //処理  
}
```

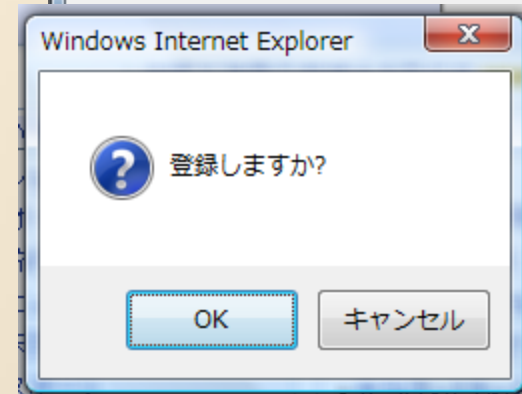
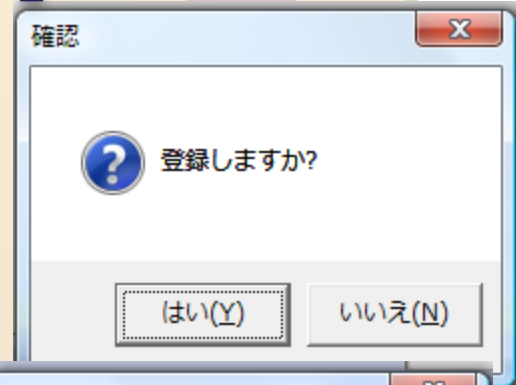
- Javascript



```
javascript:confirm('登録しますか?')
```

- Silverlightはホストページのconfirmをつかう

```
private void clk(object sender, RoutedEventArgs e)  
{  
    HtmlPage.Window.Confirm("登録しますか?");  
}
```



Silverlight












メッセージボックス

DEMO2



わんくま同盟 大阪勉強会#19

System.Windows.Browser 名前空間

	Class
	BrowserInformation
	HtmlDocument
	HtmlElement
	HtmlElementCollection
	HtmlEventArgs
	HtmlObject
	HtmlPage
	HtmlWindow
	HttpUtility
	ScriptableMemberAttribute
	ScriptableTypeAttribute
	ScriptObject

一通りのDOMコントロールはできるようです。

Panel要素

System.Windows.Controls.Panel

[System.Windows.Controls.Canvas](#)
[System.Windows.Controls.DockPanel](#)
[System.Windows.Controls.Grid](#)
[System.Windows.Controls.StackPanel](#)
[System.Windows.Controls.VirtualizingPanel](#)
[System.Windows.Controls.WrapPanel](#)
[System.Windows.Controls.Primitives.TabPanel](#)
[System.Windows.Controls.Primitives.ToolBarOverflowPanel](#)
[System.Windows.Controls.Primitives.UniformGrid](#)

[System.Windows.Controls.Panel](#)

System.Windows.Controls.Panel

[System.Windows.Controls.Canvas](#)
[System.Windows.Controls.Grid](#)
[System.Windows.Controls.StackPanel](#)

WPF

Silverlight

- Grid, StackPanelが2.0で増えた。
- WrapPanelがないのがちょっと残念
 - (Gridで代用)



WPF

Binding

Silverlight

- Binding プロパティ
 - AsyncState プロパティ
 - BindsDirectlyToSource プロパティ
 - Converter プロパティ
 - ConverterCulture プロパティ
 - ConverterParameter プロパティ
 - ElementName プロパティ
 - IsAsync プロパティ
 - Mode プロパティ
 - NotifyOnSourceUpdated プロパティ
 - NotifyOnTargetUpdated プロパティ
 - NotifyOnValidationError プロパティ
 - Path プロパティ
 - RelativeSource プロパティ
 - Source プロパティ
 - UpdateSourceExceptionFilter プロパティ
 - UpdateSourceTrigger プロパティ
 - ValidatesOnDataErrors プロパティ
 - ValidatesOnExceptions プロパティ
 - ValidationRules プロパティ
 - XPath プロパティ

- system.windows.data namespace
 - Binding Class
 - Binding Members
 - Binding Constructor
 - Binding Methods
 - Binding Properties
 - Converter Property
 - ConverterCulture Property
 - ConverterParameter Propert
 - Mode Property
 - Source Property

```
<TextBox Text="{Binding text1}" />  
<TextBox Text="{Binding text2}" Grid />  
<TextBox Text="{Binding text3}" Grid />  
<Button Click="clk" Grid.Row="2" />
```

```
<TextBox Text="{Binding Path=text1}" />  
<TextBox Text="{Binding Path=text2}" Grid />  
<TextBox Text="{Binding Path=text3}" Grid />
```

• Path=が書けない

Binding

Binding プロパティ

- AsyncState プロパティ
- BindsDirectlyToSource プロパティ
- Converter プロパティ
- ConverterCulture プロパティ
- ConverterParameter プロパティ
- ElementName プロパティ
- IsAsync プロパティ
- Mode プロパティ
- NotifyOnSourceUpdated プロパティ
- NotifyOnTargetUpdated プロパティ
- NotifyOnValidationError プロパティ
- Path プロパティ
- RelativeSource プロパティ
- Source プロパティ
- UpdateSourceExceptionFilter プロパティ
- UpdateSourceTrigger プロパティ
- ValidatesOnDataErrors プロパティ
- ValidatesOnExceptions プロパティ
- ValidationRules プロパティ
- XPath プロパティ

- ポイントになりそうな欠け
 - ElementName
 - 別の要素の値へのバインド
 - RelativeSource
 - VisualTreeをたどってのバインド
 - XPath
 - XMLへの直接のバインド

Binding

DEMO3



わんくま同盟 大阪勉強会#19

Binding

For **OneWay** or **TwoWay** bindings, dynamic changes to the source will not automatically propagate to the target. You must implement the [INotifyPropertyChanged](#) interface on the source object.

For **TwoWay** bindings, changes to the target will automatically be propagated to the source, except if the binding target is the [Text](#) property. In that case, the update will only happen when the [TextBox](#) loses focus.

For **OneTime** and **OneWay** bindings, calls to [SetValue](#) will automatically change the target value and delete the binding.

とりあえず

{Binding text1, Mode=TwoWay}

と書いておけば安心

UIElement.IsEnabledと、プロパティの継承

- <http://d.hatena.ne.jp/Yamaki/20080409/1207721518>
- Silverlight2では依存プロパティ値を継承させる方法がない
 - バインドにもない
- UIElement.IsEnabledがない
- →パネルの範囲を無効にするとかができない。

まとめ



- WPF知ってればSilverlightは全く怖くない
- はまるポイント(できること、できないことをきっちりとは)は確実に抑えよう
- そのためにはSilverlightアプリもWPFアプリも作っておくことが必要。
- RTWがめっちゃめっちゃたのしみです。

Enjoy WPF & Silverlight