

# Silverlight

## Basics

# Silverlight Overview

- Powerful development platform for creating engaging, interactive user experiences for Web, desktop, and mobile applications when online or offline.
  - Offers great flexibility by accessing the object model support provided by ASP.NET.
- Free plug-in powered by the .NET framework that is compatible across multiple browsers, devices and operating systems to bring a new level of interactivity wherever the Web works.
- Rich selection of controls
  - Extensive layout and styling options.
  - Powerful set of communication protocols.
  - Robust data access and strong support for high-definition media.
- Helps create visually rich experiences with backend support for rapid development through the Microsoft Web Platform, Visual Studio and Expression Studio.

# Silverlight features

- Runs the same in all popular Web browsers, including Microsoft Internet Explorer, Mozilla Firefox, and Apple Safari
- Runs on Microsoft Windows and Apple Mac OS X.
- Sreams video and audio.
- Includes graphics that users can manipulate — drag, turn, and zoom — directly in the browser.
- Reads data and updates the display without refreshing the whole page.
- Includes a subset of WPF technology, which greatly extends the elements in the browser for creating UI.
- Lets rich client features such as graphics, animation, media, and others to be created.
- Integrates seamlessly with your existing JavaScript and ASP.NET AJAX code.

# Running Silverlight-Based Applications

- **Small free plug-in application in a browser is needed.**
  - If users do not already have the plug-in, they are automatically prompted to install it.
  - The download and installation take seconds and require no interaction from the user except permission to install.

# Tools to create Silverlight-Based Applications

- **Visual Studio 2008 SP 1 or Visual Web Developer 2008 Express Edition SP 1.**
- **Silverlight 3 Tools for Visual Studio 2008 SP1**
  - <http://www.microsoft.com/downloads/details.aspx?familyid=9442b0f2-7465-417a-88f3-5e7b5409e9dd&displaylang=en>
  - **Visual Basic and C# Project templates**
  - **Intellisense and code generators for XAML**
  - **Debugging of Silverlight applications**
  - **Remote debugging of Silverlight applications for Mac**
  - **Web reference support**
  - **WCF Templates**
  - **Team Build and command line build support**
  - **Support for cached transparent platform extensions**
  - **Support for Silverlight 3 Out-of-Browser applications**
- **Expression Blend 3 and SketchFlow (optional)**

# Silverlight

## User Interface Layout Styles

# Silverlight UI Controls (1)

Layout and element grouping	
<b>Border</b>	Provides a border, background, or both to another control.
<b>Canvas</b>	Provides a surface to display child elements at specific coordinates in the canvas.
<b>ContentControl</b>	Represents a control with a single piece of content.
<b>Grid</b>	Provides a surface composed of rows and columns to display child elements. You define the rows and columns for a grid, then assign objects to a specific row or column in the grid.
<b>GridSplitter</b>	Allows a user to resize the columns or rows in a Grid control.
<b>StackPanel</b>	Provides a surface to display child elements in a line; either horizontally or vertically.
<b>VirtualizingStackPanel</b>	Provides a stack panel control that arranges content that is visible on the screen, creating additional UI items as needed.
<b>ScrollBar</b>	Represents a control that provides a scroll bar that has a sliding Thumb whose position corresponds to a value.
<b>ScrollViewer</b>	Provides a scrollable surface for displaying a child element.
<b>TabControl</b>	Provides a tabbed interface for displaying elements. Child elements are hosted in a TabItem.



# Layout Affecting Properties

<b>HorizontalAlignment</b>	Gets or sets the horizontal alignment characteristics applied to a control when it is composed within a layout parent. Inherited from System.Windows.FrameworkElement.
<b>HorizontalContentAlignment</b>	Gets or sets the horizontal alignment of the control's content. Inherited from System.Windows.Controls.Control.
<b>Margin</b>	Gets or sets the outer margin of a control. Inherited from System.Windows.FrameworkElement.
<b>MaxHeight</b>	Gets or sets the maximum height constraint of a control. Inherited from System.Windows.FrameworkElement.
<b>MaxWidth</b>	Gets or sets the maximum width constraint of a control. Inherited from System.Windows.FrameworkElement.
<b>MinHeight</b>	Gets or sets the minimum height constraint of a control. Inherited from System.Windows.FrameworkElement.
<b>MinWidth</b>	Gets or sets the minimum width constraint of a control. Inherited from System.Windows.FrameworkElement.
<b>Padding</b>	Gets or sets the padding inside a control. Inherited from System.Windows.Controls.Control.
<b>VerticalAlignment</b>	Gets or sets the vertical alignment characteristics applied to a control when it is composed within a layout parent. Inherited from System.Windows.FrameworkElement.
<b>VerticalContentAlignment</b>	Gets or sets the vertical alignment of the control's content. Inherited from System.Windows.Controls.Control.



# Silverlight UI Controls (2)

## Button/Command controls

<b>Button</b>	Responds to user input from a mouse, keyboard, stylus, or other input device and raises a Click event.
<b>HyperlinkButton</b>	Represents a button control that displays a hyperlink. When clicked, the HyperlinkButton enables users to move to a Web page in the same Web application or a Web page that is external to the current application.
<b>RepeatButton</b>	Represents a button that raises its click event repeatedly from when the button is pressed until it is released.

## Selection controls

<b>CheckBox</b>	Represents a control that a user can select or clear. A check box optionally offers an, indeterminate state.
<b>ComboBox</b>	Displays a drop-down list of items a user can select from.
<b>ListBox</b>	Displays a list of items a user can select by clicking.
<b>RadioButton</b>	Allows a user to select a single option from a list of options. When radio buttons are grouped together they are mutually exclusive.
<b>Slider</b>	Represents a control that lets the user select from a range of values by moving a Thumb control along a track.

# Silverlight UI Controls (3)

## Information display (read-only)

<b>TextBlock</b>	Displays small amounts of read-only text.
<b>ProgressBar</b>	Displays the current progress of an operation to the user.

## Text display and editing

<b>AutoCompleteBox</b>	Represents a control that provides a text box for user input and a drop-down that contains possible matches based on the input in the text box.
<b>PasswordBox</b>	Provides a control that allows the user to enter sensitive data, such as a password.
<b>TextBox</b>	Provides a control for displaying or editing text.

## Navigation

<b>Frame</b>	Supports navigation to Page controls.
<b>Page</b>	Encapsulates content that can be navigated to by a Frame.

## Dialog boxes and windows

<b>OpenFileDialog</b>	Enables the user to select one or more files from the file system.
<b>SaveFileDialog</b>	Enables the user to specify options for saving a file.
<b>ChildWindow</b>	Provides a window that can be displayed over a parent window and blocks interaction with the parent window.
<b>Popup</b>	Overlays content on top of the existing content within the bounds of the Silverlight plug-in.

# DEMO

## Layout

# Silverlight UI Controls (4)

## Date display and selection

<b>Calendar</b>	Allows a user to select a date from a visual calendar display. The calendar can be used on its own or in combination with the DatePicker control..
<b>DatePicker</b>	Allows a user to select a date by typing it in a text field or selecting it from a drop-down calendar control.

## Data display

<b>DataGrid</b>	Displays a collection of data in rows and columns. You can change the type of row or column to fit the needs of your application.
<b>DataPager</b>	Provides a user interface for paging through a collection of data that implements IPagedCollectionView.
<b>TreeView</b>	Displays hierarchical data in a tree structure that has items that can expand and collapse.

# Silverlight UI Controls (5)

## Graphics and video display

<b>Image</b>	Displays an image.
<b>MultiScaleImage</b>	Enables users to open a multi-resolution image which can be scaled or repositioned for detail viewing.
<b>MediaElement</b>	Hosts audio or video content. Provides a rectangular region that can display video on its surface, or play audio if no video is present.
<b>InkPresenter</b>	Provides a drawing surface to support Tablet PC features.

## User Help

<b>DescriptionViewer</b>	Displays a description and tracks error state for an associated control.
<b>Label</b>	Displays a caption, required field indicator, and validation error indicator for an associated control.
<b>ToolTip</b>	Provides the user with information about an element in the UI using a popup window.
<b>ValidationSummary</b>	Displays a summary of the validation errors on a form.

# Attached Properties

## List of some attached properties in Silverlight (not complete)

<b>Canvas.Top</b>	Define the distance for a control from the top edge of its container
<b>Canvas.Left</b>	Define the distance for a control from the left edge of its container
<b>Canvas.ZIndex</b>	Z Index of the control
<b>Grid.Row</b>	Define the row index for a control placed in a Grid container
<b>Grid.Column</b>	Define the row index for a control placed in a Grid container
<b>ScrollViewer.HorizontalScrollBarVisibility</b>	Define the visibility of the horizontal scrollbar
<b>ScrollViewer.VerticalScrollBarVisibility</b>	Define the visibility of the vertical scrollbar
<b>ToolTipService.ToolTip</b>	Define the tool tip associated with a control

# Silverlight Styles

- Created as resources either on control, page, application, or inter-application level
- Include Setters with Property and Value attributes

## Code Example

```
<UserControl..... >
  <UserControl.Resources>
    <Style x:Key="TextBoxStyle" TargetType="TextBox">
      <Setter Property="Width" Value="150"/>
      <Setter Property="Margin" Value="20,0,0,0"/>
    </Style>
  </UserControl.Resources>
  <Grid x:Name="LayoutRoot" Background="Beige">
    <TextBox Style="{StaticResource TextBoxStyle}" />
  </Grid>
</UserControl>
```



# DEMO

## Styles

# Silverlight

## Animation Media

# Animation types

Property type	Corresponding basic (From/To/By) animation	Corresponding key-frame animation	Usage example
Color	ColorAnimation	ColorAnimationUsingKeyFrames	Animate the Color of a SolidColorBrush or a GradientStop.
Double	DoubleAnimation	DoubleAnimationUsingKeyFrames	Animate the Width of a Rectangle or the Height of an Ellipse (or any FrameworkElement)
Point	PointAnimation	PointAnimationUsingKeyFrames	Animate the Center position of an EllipseGeometry.
Object	None	ObjectAnimationUsingKeyFrames	Animate the Fill property from one GradientBrush to another.

# Animation types

## Code Example

### XAML

```
<UserControl x:Class="Animation.SilverlightControl1" ... >
  <UserControl.Resources>
    <Storyboard x:Name="sb" Storyboard.TargetName="rec" >
      <DoubleAnimation Storyboard.TargetProperty="(Canvas.Top) "
        From="0" To="100" Duration="0:0:2" />
      <ColorAnimation Storyboard.TargetProperty="(Rectangle.Fill).(SolidColor.Color) "
        From="Blue" To="Yellow" Duration="0:0:2" />
    </Storyboard>
  </UserControl.Resources>
  <Canvas>
    <Rectangle x:Name="rec" Width="100" Height="100" Fill="Blue" />
  </Canvas>
</UserControl>
```

### CodeBehind

```
namespace Animation {
  public partial class SilverlightControl1 : UserControl {
    public SilverlightControl1() {
      InitializeComponent();
      sb.Begin();
    }
  }
}
```

# Animation Easing Functions

<b>BackEase</b>	Retracts the motion of an animation slightly before it begins to animate in the path indicated.
<b>BounceEase</b>	Creates a bouncing effect.
<b>CircleEase</b>	Animation accelerates and/or decelerates using a circular function.
<b>CubicEase</b>	Animation accelerates and/or decelerates using the formula $f(t) = t^3$ .
<b>ElasticEase</b>	Animation resembles a spring oscillating back and forth until it comes to rest.
<b>ExponentialEase</b>	Animation accelerates and/or decelerates using an exponential formula.
<b>PowerEase</b>	Animation accelerates and/or decelerates using the formula $f(t) = t^p$ where $p$ is equal to the Power property.
<b>QuadraticEase</b>	Animation accelerates and/or decelerates using the formula $f(t) = t^2$ .
<b>QuarticEase</b>	Animation accelerates and/or decelerates using the formula $f(t) = t^4$ .
<b>QuinticEase</b>	Animation accelerates and/or decelerates using the formula $f(t) = t^5$ .
<b>SineEase</b>	Animation accelerates and/or decelerates using a sine formula

[Easing Functions Sample](#)

# DEMO

## Animation

# MediaElement

- Represents an object that contains audio, video, or both.
- Rectangular region that can display video on its surface or play audio.
- Supports input operations such as mouse and keyboard events, and can capture focus.
- Whether or not the media plays immediately after the MediaElement object has loaded is defined AutoPlay property (bool) value
- Height and width of the video display surface can specified
  - Better to let the media display at its natural size



# DEMO

## Media

# Silverlight

## Navigation

# Navigation scopes

## Navigation within application

- Implemented by using the `Frame` and `Page` objects.
- `Page` objects represent discrete sections of content.
- `Frame` acts as a container for `Page` objects.
- `UriMapper` can be used.

## Navigation within solution

- Implemented by using `System.Windows.Browser.HtmlPage.Window.Navigate` method and relative URI.

## External navigation

- Implemented by using `System.Windows.Browser.HtmlPage.Window.Navigate` method and absolute URI.

# DEMO

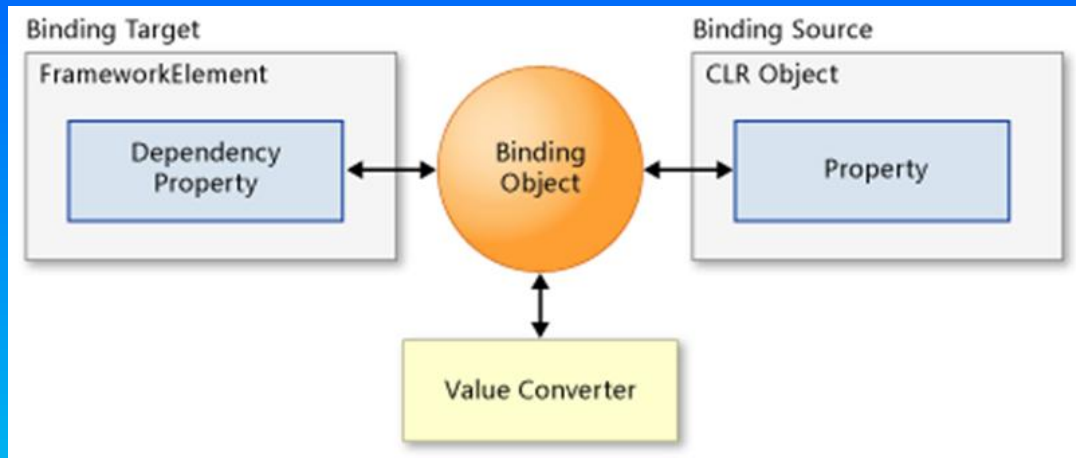
## Navigation

# Silverlight

## Data Binding

# Connecting UI Elements with Data

## Binding Concept



- ☉ Target UI property displays and possibly allows user change the data.
- ☉ Target can be any DependencyProperty of a FrameworkElement.
- ☉ Source can be any CLR object.
- ☉ Optional value converter applies to the data as it is passed.
  - ☐ Class that implements IValueConverter.

- ☉ Direction of the data flow is specified by setting the Mode property on the Binding object.

- ☐ OneTime: Target is updated when binding is created.
- ☐ OneWay (default): Target is updated when binding is created and anytime the data changes.
- ☐ TwoWay: Target and source are updated when either changes. Source updates either automatically or at times of choosing.

- ☉ In order for automatic target updates to occur, the source object must implement the INotifyPropertyChanged interface.

# DEMO

## Data Binding



# Silverlight

## Asynchronous ASMX Web Service Calls

# Asynchronous ASMX Web Service Calls (1)

When service reference is added Visual Studio creates binding configuration automatically and saves it in ServiceReferences.ClientConfig file

```
<configuration>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="Service1Soap"
          maxBufferSize="2147483647"
          maxReceivedMessageSize="2147483647">
          <security mode="None">
            <transport>
              <extendedProtectionPolicy policyEnforcement="Never" />
            </transport>
          </security>
        </binding>
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint address="http://localhost:49367/Service1.asmx"
        binding="basicHttpBinding"
        bindingConfiguration="Service1Soap"
        contract="ProductService.Service1Soap"
        name="Service1Soap" />
    </client>
  </system.serviceModel>
</configuration>
```

The diagram illustrates the relationship between the binding configuration and the endpoint configuration in the ServiceReferences.ClientConfig file. A box labeled "Static endpoint address" points to the `address` attribute of the `<endpoint>` element. Two curved arrows originate from the `<binding name="Service1Soap">` element and point to the `binding` and `bindingConfiguration` attributes of the `<endpoint>` element, indicating that the endpoint uses the specified binding configuration.

# Asynchronous ASMX Web Service Calls (2)

ASMX web service is called asynchronously by Silverlight client

```
Uri endpointaddr = new Uri(Application.Current.Host.Source, "../Service1.asmx");
public MainPage() {
    InitializeComponent();
    ProductService.Service1SoapClient proxy = new ProductService.Service1SoapClient(
        "Service1Soap", endpointaddr.AbsoluteUri);
    proxy.GetAllProductsCompleted += new EventHandler<ProductService.
        GetAllProductsCompletedEventArgs>(proxy_GetAllProductsCompleted);
    proxy.GetAllProductsAsync();
}
void proxy_GetAllProductsCompleted(object sender, ProductService.
    GetAllProductsCompletedEventArgs e) {
    AllProducts.DataContext = e.Result;
}
```

Dynamic endpoint address

Service proxy

Binding name

Asynchronous Service call

Completed event handler declaration

Returned object can be consumed in completed event handler

```
[WebMethod]
public List<Product> GetAllProducts() {
    ProductbaseDataContext pbdc = new ProductbaseDataContext();
    return pbdc.Products.Select(p => p).OrderBy(p => p.Code).ToList();
}
```

# DEMO

## Asynchronous ASMX Web Service Call

# Links

[Microsoft Official Silverlight Site](#)

[Microsoft Silverlight Learning Videos](#)

[Silverlight Animation Easing Functions](#)