



Silverlight Development

Part2

The Final Answer of Rich Web Technology

株式会社アークウェイ
代表取締役社長 .NET コンサルタント
森屋英治(Hideharu Moriya)
Microsoft MVP Solutions Architect

アークウェイについて



- ビジョン:『開発者たちの幸せ』
- サービス
 - コンサルティングサービス
 - .NET技術を中心としたアーキテクチャ／システム構築のコンサルテーション
 - アーキテクチャ策定サービス
 - 最適なアーキテクチャ策定をご支援します
 - メンタリングサービス
 - 開発現場の一員として参加しながら、.NET技術やアジャイル開発などの技術指導を行います
 - トレーニングサービス
 - .NET技術、アジャイル開発などのトレーニング

アークウェイについて



● 製品

- aiNote (アイノート)
 - マインドマップ機能対応ソフトウェアファクトリ支援ツール
 - **ベータ3 2007/09版 明日リリース!**

● 研究開発

- ソフトウェアファクトリ
- NAgile

● 詳しくはホームページをご覧ください

- URL: <http://www.archway.co.jp>

● モチベーションが高い開発者、メンターを募集しています!

- 連絡先: sales@archway.co.jp

Part1 Agenda

- Silverlightとは
- コンセプト
- DAL
 - 開発スタイル
 - アーキテクチャ
 - XAML
- オブジェクトブラウザは親友
- 基本コンポーネント
- X:Name
- イベント
- 1.0 と 1.1の違い

Part2 Agenda

- SilverLightとは
- Coding Anywhere
- .NETとの連携
 - ASP.NETとの連携
 - ネットワーク
 - XML
 - LINQ
- その他
 - ファイル
 - MACでの開発



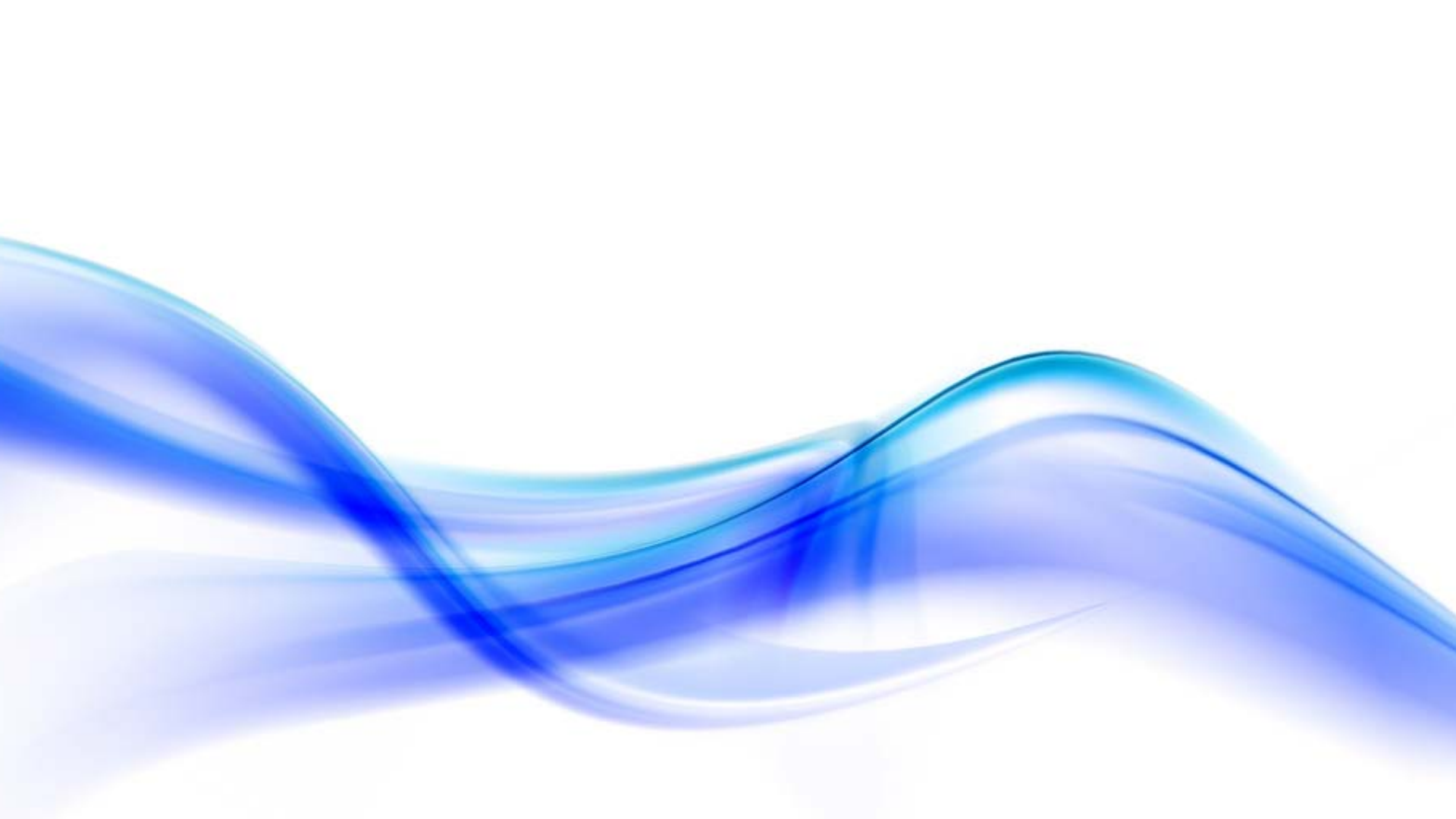
Microsoft®
Silverlight™

Silverlightとは

- Microsoft Silverlight は、クロスブラウザ・クロスプラットフォームに対応する **Web ブラウザのプラグイン**です。**.NET を基盤**とし、Web における次世代のメディア体験と RIA* を実現します。

* RIA (Rich Interactive Application)

参照: <http://www.microsoft.com/japan/silverlight/default01.aspx>



Coding Anywhere

Coding Anywhere

- Silverlight アーキテクチャ
- JavaScriptからのアクセス
- マネージドコードからのアクセス
- ダイナミックランゲージ(Python)
- ASP.NETとの連携

Browser Host

MS AJAX Library

DOM Integration

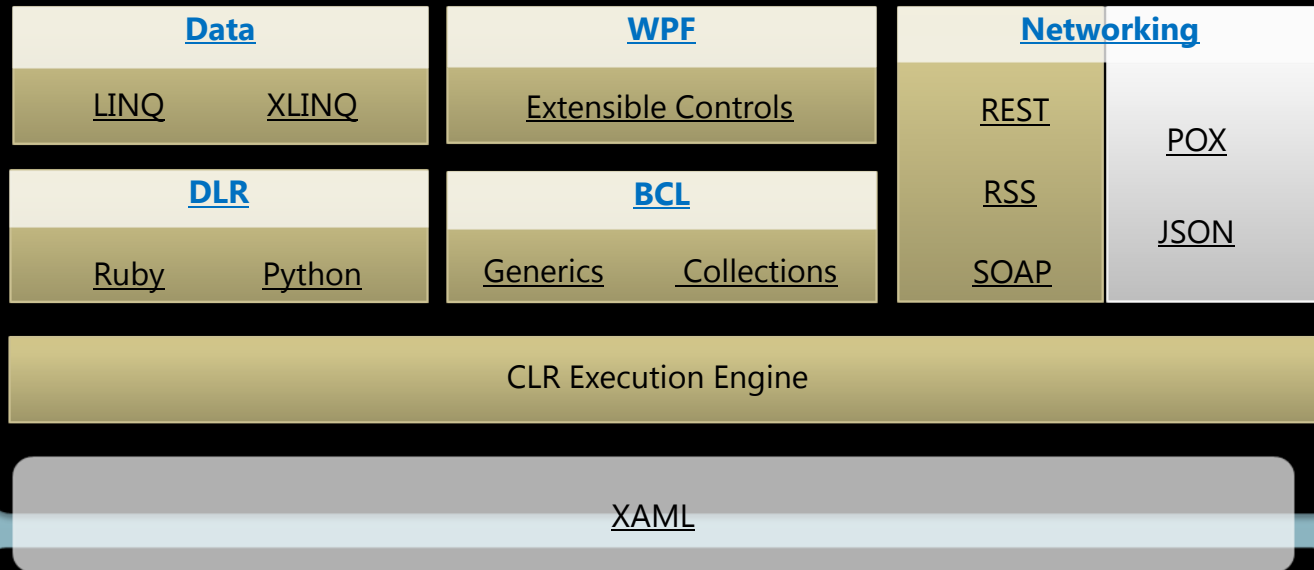
Application Services

Deploy

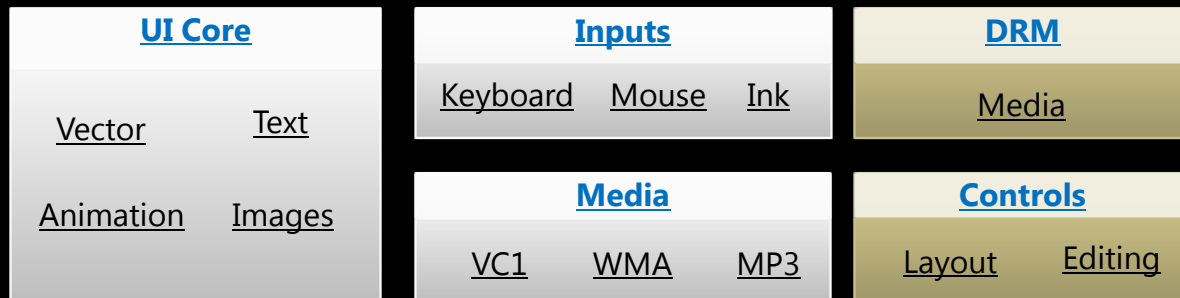
Friction-Free Installer

Auto-Updater

.NET for Silverlight



Presentation Core

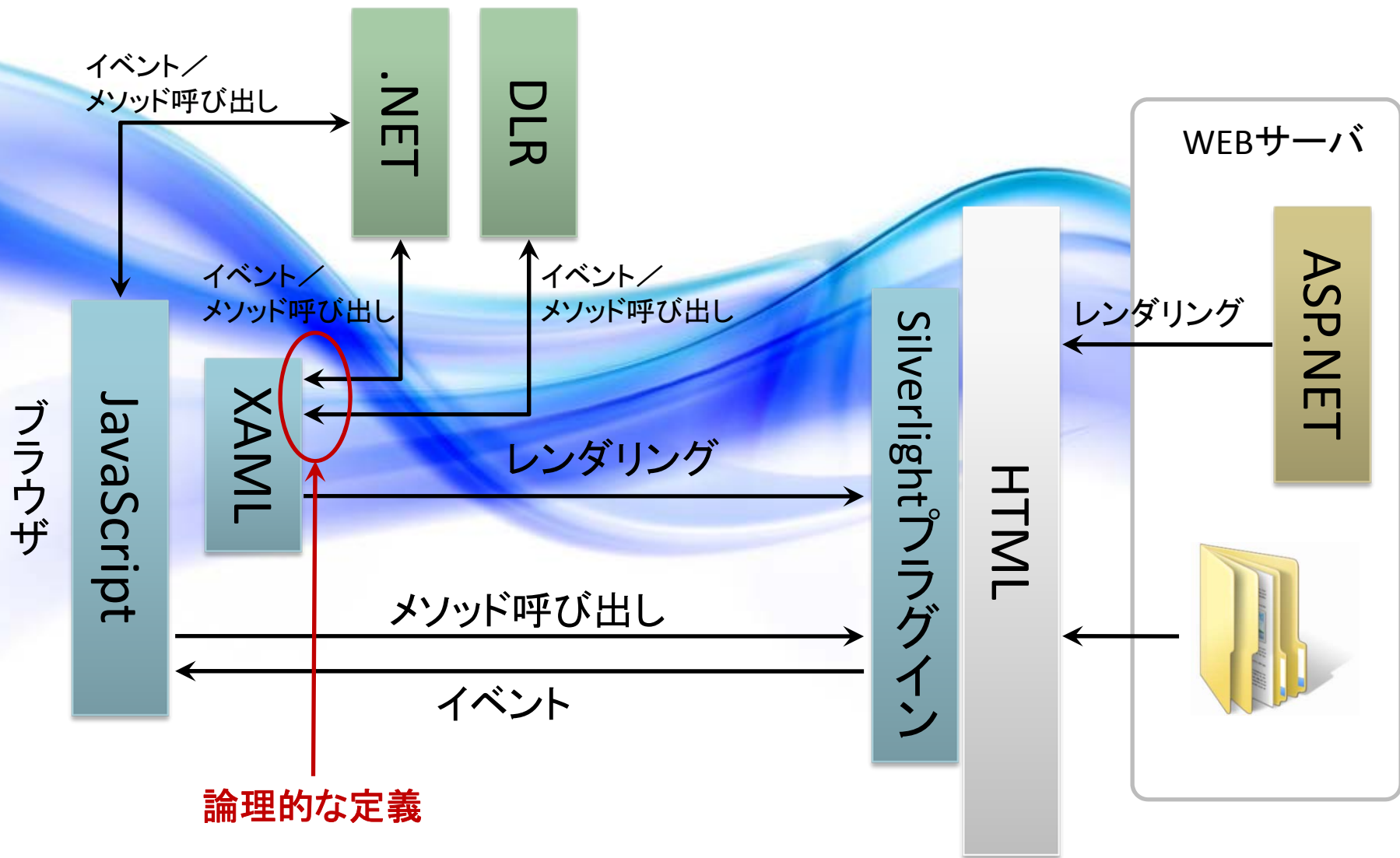


Legend

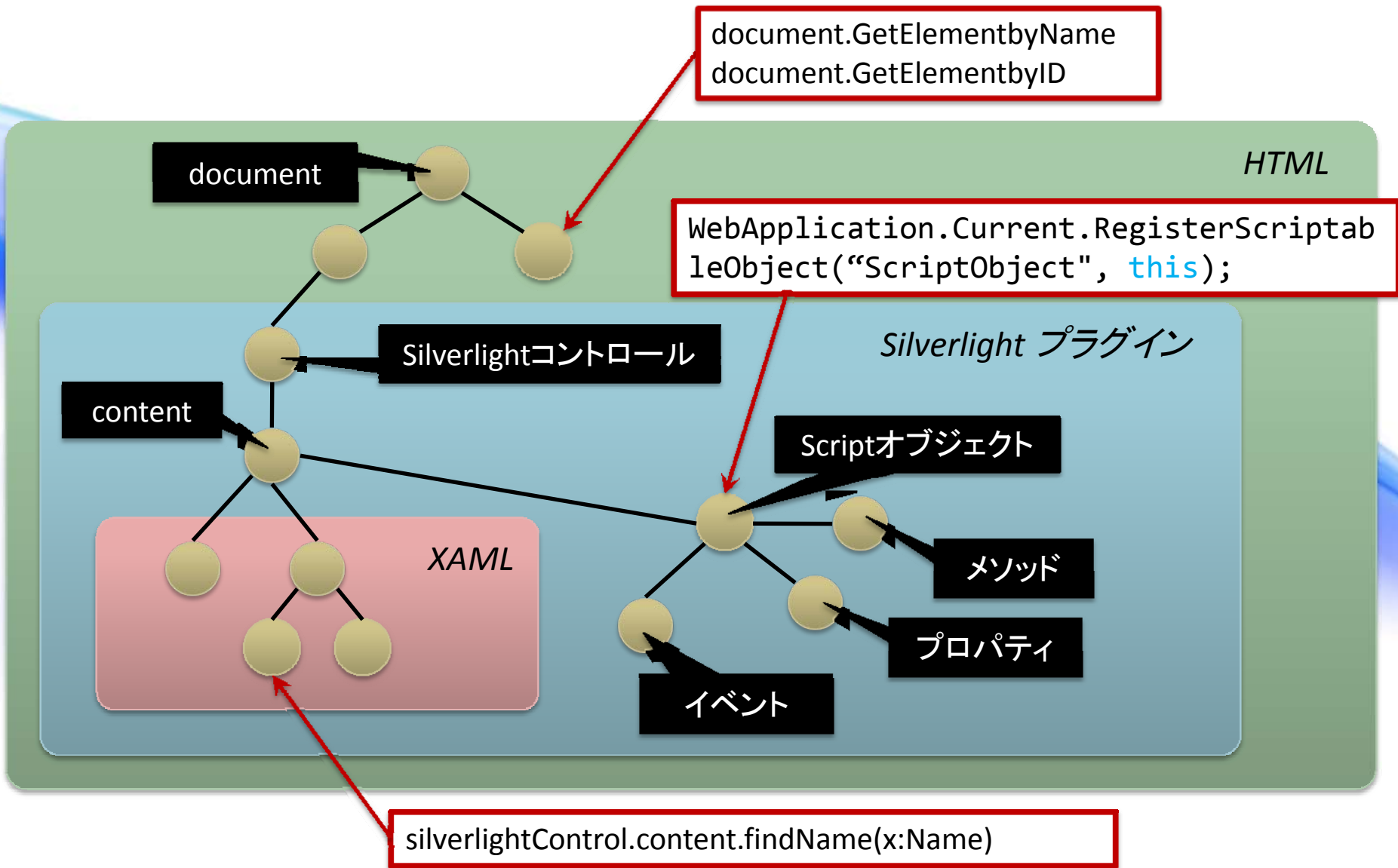
V1.1

Legend

V1.0



JavaScriptからのアクセス

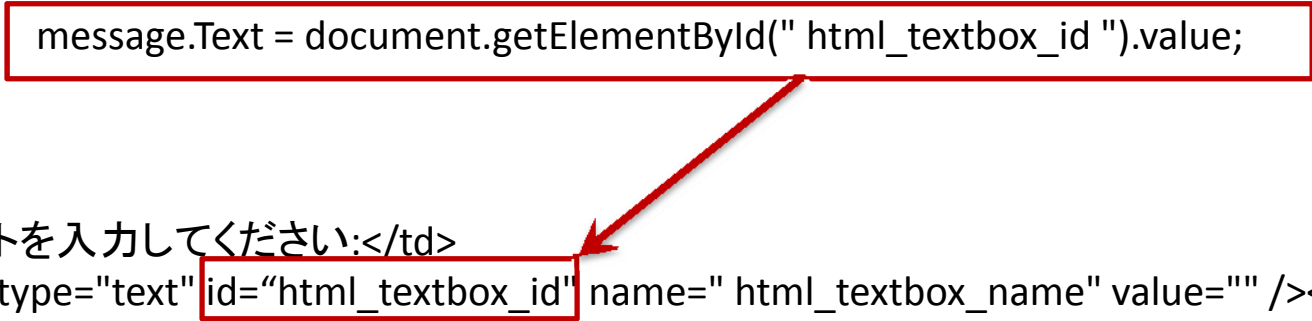


JavaScriptからのアクセス

- JavaScriptからDOM
 - `document.GetElementbyName`
 - `document.GetElementbyID`
- JavaScriptからXAML
 - `silverlightControl.content.findName(x:Name)`
- JavaScriptからマネージドコード
 - [Scriptable]属性
 - `WebApplication.Current.RegisterScriptableObject("basic", this)`
 - `control.Content.basic.Search()`

JavaScriptからDOMへのアクセス

```
<div id="SilverlightControlHost" class="silverlightHost" >
  <script type="text/javascript">
    createSilverlight();
    function HideForm() {
      var silverlightControl = document.getElementById('SilverlightControl');
      if (silverlightControl) {
        var message = silverlightControl.content.findName("messageText");
        message.Text = document.getElementById("html_textbox_id").value;
      }
    }
  </script>
  <table><tr>
    <td>テキストを入力してください:</td>
    <td><input type="text" id="html_textbox_id" name="html_textbox_name" value="" /></td>
  </tr>
  <tr><td>&nbsp;</td>
    <td><input type="button" id="m_Submit" name="m_Submit" value="クリック"
onclick="HideForm();" /></td></tr></table>
</div>
```



JavaScriptからXAMLへのアクセス

```
<script type="text/javascript">
  createSilverlight();
  function HideForm() {
    var silverlightControl = document.getElementById('SilverlightControl');
    if (silverlightControl) {
      var storyboard = silverlightControl.content.findName("testStory");
      storyboard.Begin();
    }
  }
</script>
```

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  x:Name="parentCanvas" Loaded="Page_Loaded"
  x:Class="AttachFromJavaScript.Page;assembly=ClientBin/AttachFromJavaScript.dll"
  Width="640" Height="480" Background="Black">
  <Canvas.Resources>
    <Storyboard x:Name="testStory">
      <DoubleAnimationUsingKeyFrames BeginTime="00:00:00" Storyboard.TargetName="messageText"
      Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[3].(TranslateTransform.X)"/>
    </Storyboard>
  </Canvas.Resources>
</Canvas>
```

...(以下略)...

JavaScriptからマネージドコード

プロパティやメソッドまたはイベントにScriptable属性を設定

```
[Scriptable]  
public void Search(string Name) { ... }
```

Scriptable オブジェクトを登録する

```
WebApplication.Current.RegisterScriptableObject("basic", this);
```

スクリプトからマネージドコードへアクセスする

```
var control = document.getElementById("silverlightcontrol");  
control.Content.basic.Search(input.value);
```


マネージドコードからのアクセス

- マネージドコードからHTML DOM
 - `HtmlPage.Document.GetElementById`
- マネージドコードからXAML

マネージドコードからHTML DOM

HTML DOM へのアクセスを可能とする名前空間の追加

```
using System.Windows.Browser;
```

イベントとの連結やメソッドコール、プロパティへのアクセス

```
HtmlElement myButton =  
HtmlPage.Document.GetElementById("myButtonID");  
myButton.AttachEvent("onclick", new  
EventHandler(this.myButtonClicked));  
  
private void myButtonClicked(object sender, EventArgs e) { ...  
}
```

マネージドコードからXAML

```
<Rectangle x:Name="rectangle"/>
```

```
void OnMouseEnter(object sender, MouseEventArgs e) {  
    rectangle.Height = 75;  
    rectangle.Opacity = 0;  
}
```

DLR(Python)

Page.xaml

```
<Canvas x:Name="parentCanvas"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Loaded="OnLoad"
  Width="640"
  Height="480"
  Background="Black"
  >
  <x:Code Source="Code.py" Type="text/ironpython" />
  <TextBlock x:Name="Message" Foreground="White" Text="Hello" FontSize="100"
  MouseLeftButtonDown="OnClick" />
</Canvas>
```

x:Classが必要**ない**

Code.py

```
def OnClick( sender, e ):
    Message.Text = "Clicked!!"
    Message.FontSize = Message.FontSize * 1.1
```

DEMO

The background of the slide features a series of overlapping, wavy, translucent blue lines that create a sense of motion and depth. The lines vary in opacity and color, ranging from light blue to a deeper, more saturated blue. They flow across the frame, with some lines curving upwards and others downwards, creating a dynamic and fluid visual effect.

.NETとの連携

Rich Web Technology



.NET



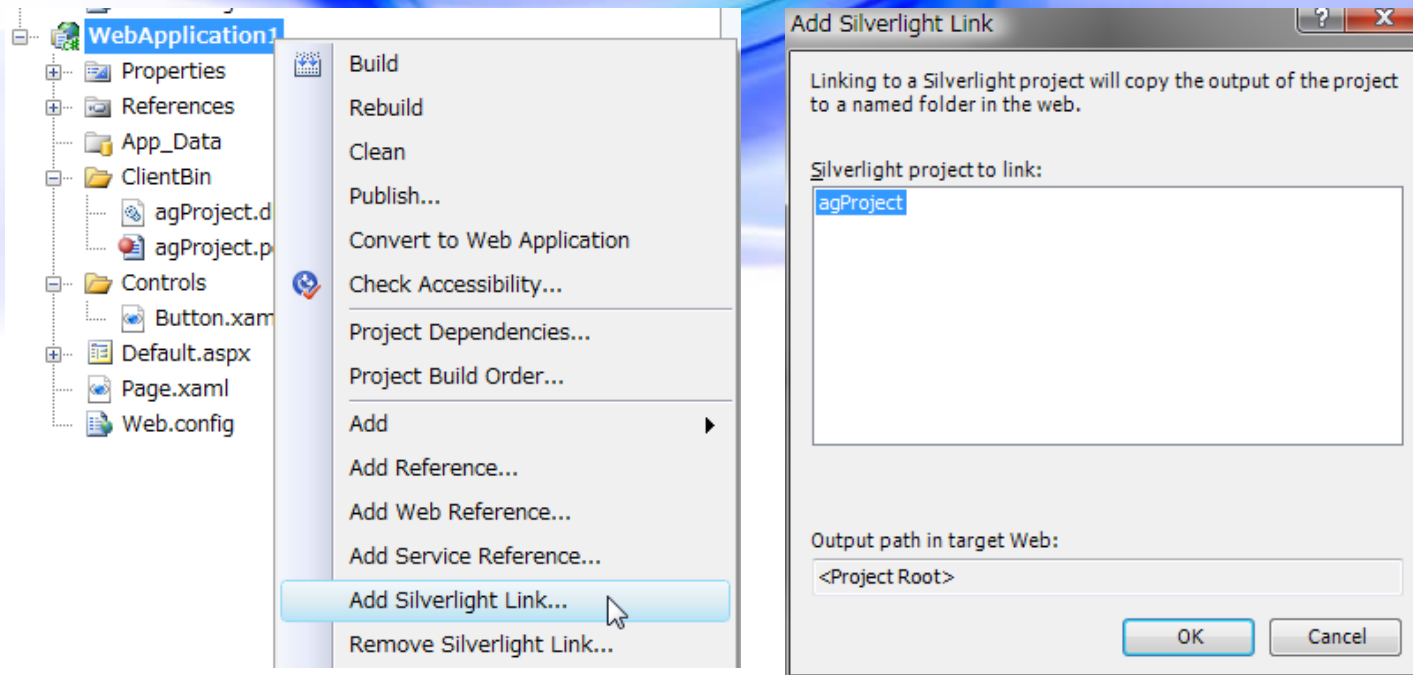
Final Answer!

.NETとの連携

- ASP.NETとの連携(ASP.NET Futures)
- ネットワーク
- XML
- LINQ

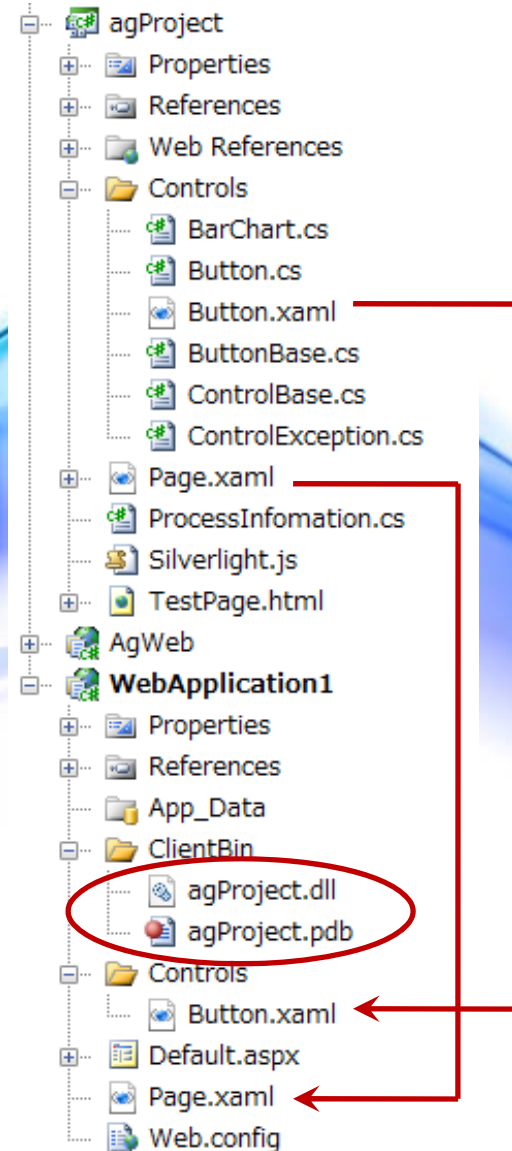
ASP.NETとの連携(ASP.NET Futures)

- ASP.NET Features をインストール
 - 入手先:
<http://www.microsoft.com/downloads/details.aspx?FamilyID=9323777e-fe78-430c-ad92-d5be5b5ead98&DisplayLang=en>
- ASP.NET Web Applicationプロジェクトを右クリックし、[Add Silverlight Link...]をクリック
- 関連付けるSilverlightプロジェクトを選択



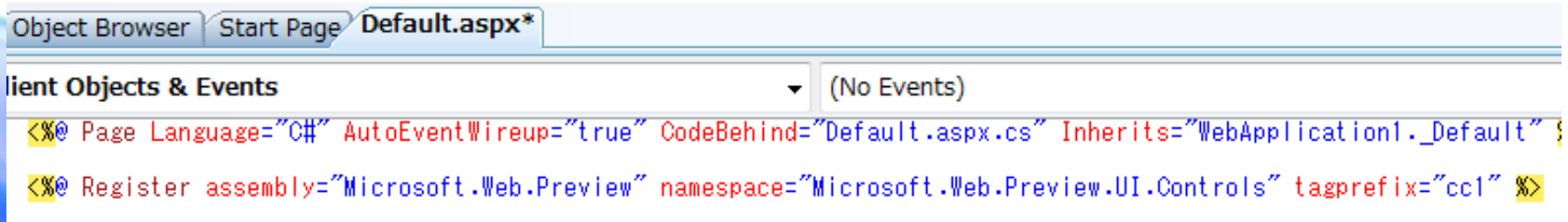
ASP.NETとの連携(ASP.NET Futures)

- XAMLファイルがコピーされる
- コンパイルされたものは ClientBinの下に配置される



ASP.NETとの連携(ASP.NET Futures)

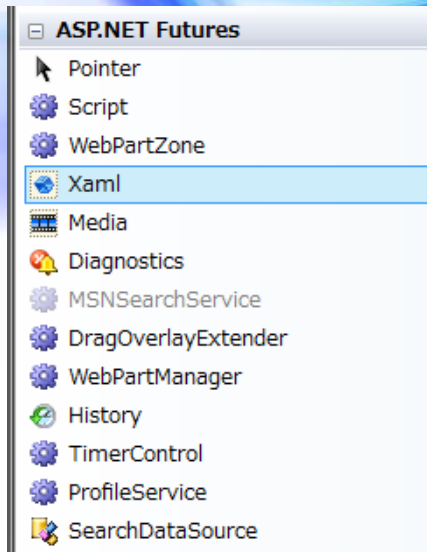
- Microsoft.Web.Preview名前空間が追加される



The screenshot shows the Visual Studio IDE with the 'Object Browser' and 'Start Page' tabs. The 'Default.aspx*' tab is active, displaying the 'Client Objects & Events' section. The code in the background includes the following lines:

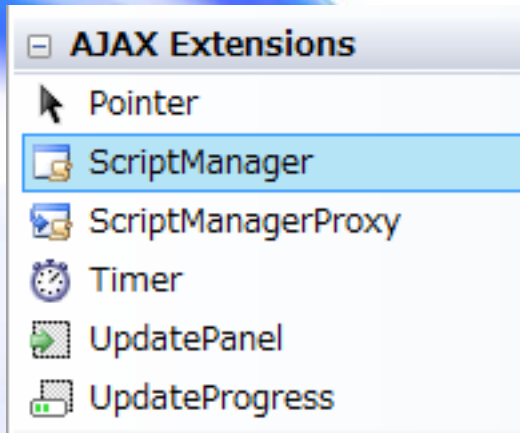
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebApplication1._Default" %>  
<%@ Register assembly="Microsoft.Web.Preview" namespace="Microsoft.Web.Preview.UI.Controls" tagprefix="cc1" %>
```

- ツールボックスのASP.NET FuturesタブからXamlを追加



ASP.NETとの連携(ASP.NET Futures)

- XamlUrlプロパティに使用するXAMLファイルを指定
- Xamlだけではエラーとなるため、AJAX ExtensionsタブのScriptManagerを追加する必要がある



```
<cc1:Xaml ID="Xaml1" runat="server" Height="329px" Width="422px" XamlUrl="Page.xaml">  
</cc1:Xaml>  
<asp:ScriptManager ID="ScriptManager1" runat="server">  
</asp:ScriptManager>
```


ネットワーク

- Downloader
- Web Service
- WCF Coming soon...

Downloader

```
private void DownloaderInit()  
{  
    Downloader downloader = new Downloader();  
    downloader.Open("GET", new Uri("/data/logos.zip", UriKind.Relative));  
    downloader.Completed += OnDownloadComplete;  
    downloader.DownloadFailed += OnDownloadFailed;  
    downloader.Send();  
}
```

```
public void OnDownloadComplete(object sender, EventArgs e)  
{  
    ImageControl.SetSource(sender, "Silverlight_logo.png");  
}
```

Logos.zipの中の
ファイル名を指定

Webサービス

- JSONのみサポート(現時点)
- ローカルサービスのみサポート(現時点)
- Silverlight用のプロキシジェネレーターもサポート
 - Typed データアクセス可能
- 同期と非同期モデル

Web Services (同期)

Web Service側の設定

```
[System.Web.Script.Services.ScriptService]
public class ProcessData : System.Web.Services.WebService
{
    [WebMethod]
    [ScriptMethod(ResponseFormat = ResponseFormat.Json)]
    public String GetInfomation()
    {
    }
}
```

クライアントコード

```
private ProcessData proxy = new ProcessData();
string result = proxy.GetInfomation();
```

Web Services (非同期)

Webサービスの非同期呼び出し

```
_proxy.BeginGetInfomation(new AsyncCallback(OnWSCallComplete), null);
```

非同期のコールバック

```
public void OnWSCallComplete(IAsyncResult asyncResult)
{
    _xmlData = _proxy.EndGetInfomation(asyncResult);
}
```

XmlReader & XmlWriter

- AlphaバージョンにおけるCore XML ライブラリのXMLのリードとライト機能
- LINQ to XML サポート予定

Readerの初期化

```
XmlReader xr = XmlReader.Create(new StringReader(RawResponse));
```

ノードの検索と値の読み込み

```
xr.ReadToFollowing("ProcessData");  
string playerNodeText = xr.Value;  
string playerNameAttribute = xr.GetAttribute("Name");
```

LINQ

- LINQ = Language **I**Ntegrated **Q**uery
- C#3.0,VB9.0,..開発言語のクエリー機能拡張
- オブジェクト、リレーショナルDB、XMLをクエリーにより統合可能
- すべてのIEnumerable<T> ベースのソースで動作可能

```
var query = from p in _data
            orderby p.WorkingSet descending
            where p.WorkingSet > 8*1024*1020
            select new ProcessInformation
            {
                Name = p.Name,
                WorkingSet = p.WorkingSet
            };
```

LINQ

C#

VB9.0

その他の言語

Silverlight
サポート

.NET Language Integrated Query

Silverlight
サポート予定

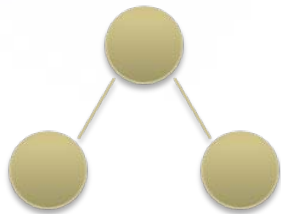
LINQ to
Objects

LINQ to
DataSets

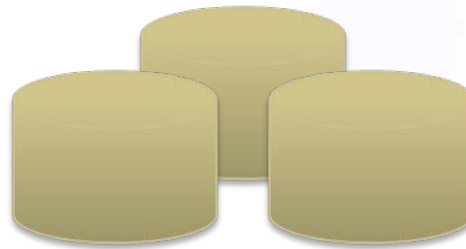
LINQ to
SQL

LINQ to
Entities

LINQ to
XML



オブジェクト



リレーショナル

```
<book>  
  <title/>  
  <author/>  
  <year/>  
  <price/>  
</book>
```

XML

WS + LINQ

[WebMethod]

[ScriptMethod(ResponseFormat = ResponseFormat.Json)]

```
public String GetInfomation()
```

```
{
```

```
    return (new XElement("ProcessInfo",
```

```
        from p in Process.GetProcesses()
```

```
        select new XElement("ProcessData",
```

```
            new XElement("Name", p.ProcessName),
```

```
            new XElement("WorkingSet",
```

```
p.WorkingSet64))))).ToString();
```

```
}
```



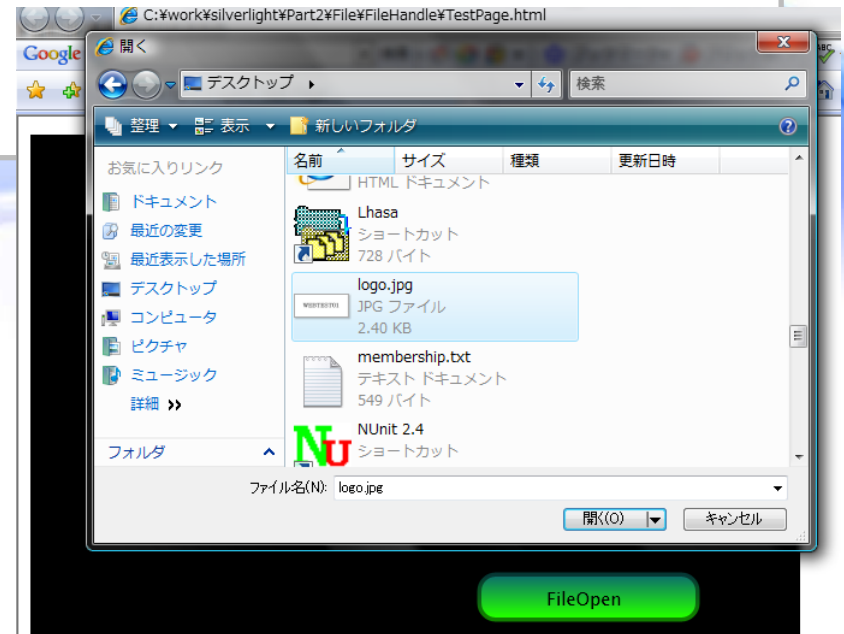
その他

その他

- ファイル
 - File Dialog
 - IsolatedStorage
- MACでの開発

File Dialog

```
public void OnClicked(object o, EventArgs e)
{
    OpenFileDialog fileDialog = new OpenFileDialog();
    if (DialogResult.OK == fileDialog.ShowDialog())
    {
        Message.Text = "Opened" +
            fileDialog.SelectedFile.Name;
    }
}
```



IsolatedStorage

- 分離ストレージは、仮想ファイルシステムを抽象化したもので、仮想ファイルシステムのルートは、物理的なファイルシステムにおいて、ユーザごとに配置される隠しフォルダとなる
- 安全なユーザ単位のファイルを扱うことができる
- スマートクライアントのシナリオでは必須
- Cookieよりもはるかに多くのデータを処理できる

IsolatedStorage File への書き込み

```
using (IsolatedStorageFile isoStore
    = IsolatedStorageFile.GetUserStoreForApplication())
{
    using (IsolatedStorageFileStream isoStream =
        new IsolatedStorageFileStream("ProcessInfo.xml",
            FileMode.Create, isoStore))
    {
        using (StreamWriter writer = new
            StreamWriter(isoStream))
        {
            writer.Write(_xmlData);
        }
    }
}
```

MACでの開発

- Webサーバとブラウザ
 - サーバ: Apachその他あらゆるサーバ
 - ブラウザ: Firefox、Safari(インテルMACのみ)

MACでの開発(デバック)

- MACで Silverlight code のデバックが可能
- MACにプロキシクライアントのインストールが必要
- プロキシとセットアップ手順
 - C:\Program Files\Microsoft Visual Studio 9.0\Silverlight\MacIntel\
 - プロキシはブラウザアクティベーションの前に実行されている必要あり

DEMO

まとめ

- SilverLightとは
- 開発スタイル
- Coding Anywhere
- .NETとの連携
 - ASP.NETとの連携
 - ネットワーク
 - XML
 - LINQ
- その他
 - ファイル
 - MACでの開発

Enjoy
“Silverlight”

ご静聴
ありがとう
ございました

Live Search

検索欄: アークウェイ

検索ボタン: 

検索タイプ: [ウェブ](#) [画像](#) [ニュース](#) [地図\(Beta\)](#)

ページナビゲーション

```
HtmlPage.Navigate("http://www.archway.co.jp");  
String server = HtmlPage.DocumentUri.Host;
```

リソース

- <http://silverlight.net/>
- <http://msdn2.microsoft.com/en-us/library/bb428859.aspx>