

Windows
Phone

Web Services

Interacting with Web Services, OData
Services and Web Content

Networking for Windows Phone

- Support for networking features:
 - Windows Communication Foundation (WCF)
 - HttpWebRequest
 - WebClient
 - No cross-domain policy file required
- Not Supported:
 - Sockets
 - Custom WCF Bindings
 - WCF RIA Services
 - NTLM authentication
 - RSS and Atom feeds (syndication)

Windows Phone Jump Start

Simple Http Operations: WebClient

```
1. WebClient client = new WebClient();
2. public MainPage()
3. {
4.     InitializeComponent();
5.
6.     client.DownloadProgressChanged += client_DownloadProgressChanged;
7.     client.DownloadStringCompleted += client_DownloadStringCompleted;
8.     client.OpenReadCompleted += client_OpenReadCompleted;
9. }
10. private void WebClientButton_Click(object sender, RoutedEventArgs e){
11.     client.DownloadStringAsync(
12.         new Uri("http://localhost/ServicesApplication/rssdump.xml"), "sample rss");
13. }
14.
15. void client_DownloadProgressChanged(object sender, DownloadProgressChangedEventArgs e){
16.     if (e.UserState as string == "sample rss"){
17.         this.DownloadProgress.Value = e.ProgressPercentage;
18.     }
19. }
20.
21. void client_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e){
22.     this.DownloadedText.Text = e.Result;
23. }
```

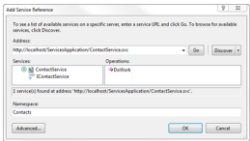
Windows Phone Jump Start

More Control: HttpWebRequest

```
1. private void HttpWebRequestButton_Click(object sender, RoutedEventArgs e){
2.     var req = HttpWebRequest.Create(
3.         new Uri("http://localhost/ServicesApplication/rssdump.xml")) as HttpWebRequest;
4.     req.BeginGetResponse(HttpWebRequestButton_Callback, req);
5. }
6.
7. private void HttpWebRequestButton_Callback(IAsyncResult result){
8.     var req = result.AsyncState as HttpWebRequest;
9.     var resp = req.EndGetResponse(result);
10.    var strm = resp.GetResponseStream();
11.    var reader = new StreamReader(strm);
12.
13.    this.Dispatcher.BeginInvoke(() =>{
14.        this.DownloadedText.Text = reader.ReadToEnd();
15.        this.TextView.Visibility = System.Windows.Visibility.Visible;
16.    });
17. }
```

WCF/ASMX Services

- Can 'Add Reference' from Windows Phone projects to automatically generate proxy classes
 - ASMX should 'just work'
 - WCF requires that you use basicHttpBinding



RESTFUL WEB SERVICES

Building Them

- Instead of building "walled gardens," data should be published in a way that allows it to reach the broadest range of mobile clients
- With Windows Mobile, you built SOAP 1.1 Web Services using ASP.NET or Windows Communication Foundation (WCF) with BasicHttpBinding
- With Windows Phone 7 and Silverlight, we use WCF both on-premise and as a Web Role in Windows Azure to publish our data from local and cloud-based data sources like SQL Azure
- We use lightweight REST + JSON Web Services that are better optimized for high-latency, slow, intermittent wireless data connections

WCF Data Services: OData

- WCF Data Services provides an extensible tool for publishing data using a REST based interface
 - Publishes and consumes data using the OData web protocol (<http://www.odata.org>)
 - Formatted in XML or JSON
- Download OData Client Library for Windows Phone 7 Series CTP from MSDN downloads

```
<?xml version="1.0" encoding="utf-8"
standalone="yes" ?>
<service xmlns:base=
"http://odata.netflix.com/Catalog/" *>
  <workspace>
    <atom:title>Default</atom:title>
    <collection href="Genres">
      <atom:title>Genres</atom:title>
    </collection>
    <collection href="Titles">
      <atom:title>Titles</atom:title>
    </collection>
  ...
</workspace>
</service>
```

Generate Client Proxy

- Open a command prompt as administrator and navigate to %windir%\Microsoft.NET\Framework\v4.0.30128
- Run this command :
DataSvcutil.exe /uri:[http://odata.netflix.com/Catalog/](http://odata.netflix.com/Catalog/DataServiceCollection/Version:2.0/outnetflixClientTypes)
/DataServiceCollection /Version:2.0 /outnetflixClientTypes
- Add generated file to your project

Fetching Data

```
1. public class NetflixWiaModel : INotifyPropertyChanged
2. {
3.     private DataServiceCollection<Title> _titles;
4.     private NetflixCatalog context;
5.
6.     public NetflixWiaModel()
7.     {
8.         context = new NetflixCatalog(new Uri("http://odata.netflix.com/Catalog"));
9.         _titles = new DataServiceCollection<Title>(context);
10.    }
11.
12.    public void GetTitles(int numberOfTitles)
13.    {
14.        _titles.Clear();
15.        _titles.LoadAsync(context.Titles.Take(numberOfTitles));
16.    }
17.    void Titles_LoadCompleted(object sender, LoadCompletedEventArgs e)
18.    {
19.        Titles.Clear();
20.        IEnumerable<Title> titles = e.QueryOperationResponse as IEnumerable<Title>;
21.        foreach (Title title in titles)
22.        {
23.            Titles.Add(title);
24.        }
25.    }
26. }
```

OData Services

Demo

10

Windows Phone Jump Start

NETWORK AWARENESS

Making Decisions based on Data Connections

- Mobile apps shouldn't diminish the user experience by trying to send or receive data in the absence of network connectivity
- Mobile apps should be intelligent about performing heavy data transfers or lightweight remote method calls only when the appropriate data connection is available
- With Windows Phone 7 and Silverlight, we use the **NetworkInterfaceType** object to detect network type and speed and the **NetworkChange** object to fire events when the network state changes

Windows Phone Jump Start

Data Formats

- Simple test case: Download 8 data records
 - Each record just four fields
- Measured bytes to transfer:
 - The OData version from WCF Data Services which weighed-in at 8.5k
 - The ASMX SOAP + DataSet version was 3k
 - The REST + XML version was 1.2k
 - The REST + JSON version was 639 bytes

12

Windows Phone Jump Start

Using JSON serialization

Demo

13

Windows Phone Jump Start

Web Browser Control

- Displays network and local content
- Supports pan, double tap and pinch to zoom
- Supports transforms & projections
- Application can interact with javascript
- Optimized for privacy
 - Script is disabled by default
 - Cookies and cache are isolated


Silverlight to Script

Silverlight

```
string fromScript = wb.InvokeScript("runScriptStuff",  
arg1, arg2);
```

JavaScript

```
function runScriptStuff(arg1, arg2)  
{ return "awesome"; }
```



Script to Silverlight

Silverlight

```
void wb_ScriptNotify(object s, NotifyEventArgs e)
{
    string fromScript = e.Value;
}
```

javascript


```
window.external.Notify(string);
```

WebBrowser control

Demo

17

Windows Phone Jump Start



Windows
Phone

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names and/or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.
