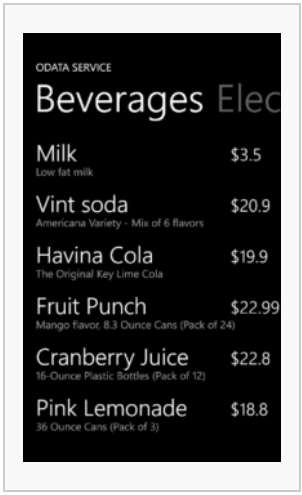
OData client library for Windows Phone 7

This article demonstrates how to consume an Open Data Protocol (OData) feed in a Windows Phone app using the OData client library for Windows Phone 7. Note that it does not work in Windows Phone 8

Introduction

The OData client library for Windows Phone 7 generates HTTP requests to a data service that supports OData and transforms the entries in the response feed into objects on the client. The DataSvcUtil.exe tool creates the client data service classes that are needed to access a data service from a .NET Framework client application. This tool has the same form as the normal SDK for .NET and Silverlight framework has. In this example we will use http://services.odata.org/OData/OData.svc/ OData service to read data feeds.

Note: Microsoft has launched OData Client Tools for Windows Phone which supports OData V3. This release supports the `Add Service Reference` experience with client-side OData support for developing Windows Phone 7 and Windows Phone 8 applications using Windows Phone SDK 8.0 To learn more see OData client for Windows Phone.



Category 1

Category 2

Implementation

Create a standard Windows Phone Project

Let's create a standard Windows Phone Application Project

- Launch **Visual Studio**
- Click on **File**
- Click on **New Project**
- Select **Windows Phone Pivot Application** (Visual C# Template)
- Add **Name** and **Location** of the project
- Click **OK** to create the project.

WCF Data Service Client Utility (DataSvcUtil.exe)

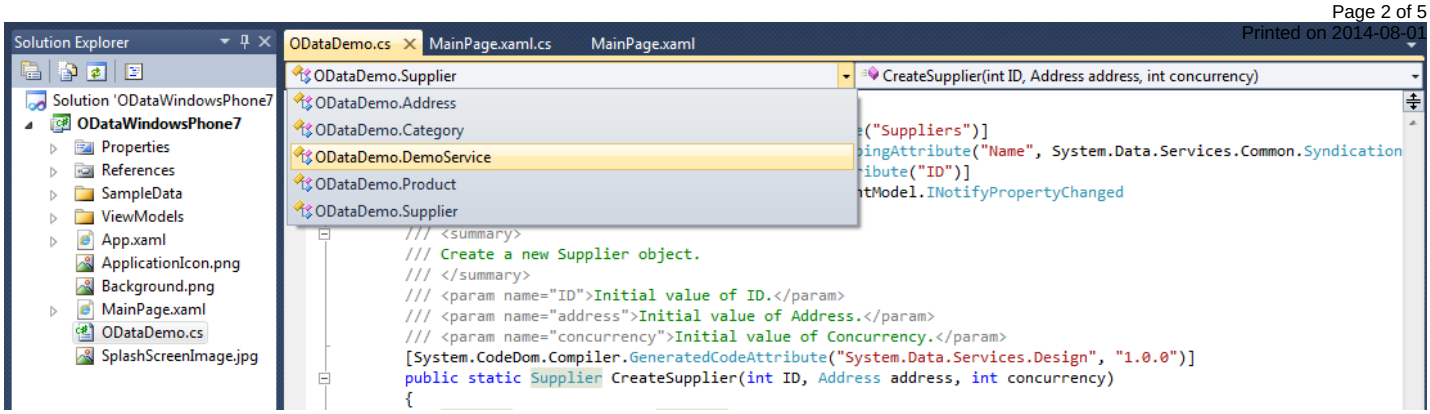
`DataSvcUtil.exe` is a command-line tool provided by WCF Data Services. To create the client data service classes and to consume an Open Data Protocol (OData) feed, let's download the OData client SDK for Windows Phone 7 and extract the files in your project folder. To use the `DataSvcUtil.exe` tool let's launch the command prompt, and navigate to our project directory and run the below command. This will generate a **ODataDemo.cs** file in your project folder.

```
DataSvcUtil.exe /uri:http://services.odata.org/OData/OData.svc /version:2.0
/DataServiceCollection /language:CSharp /out:ODataDemo.cs
```

```
C:\WP7\workspace\ODataWindowsPhone7\ODataWindowsPhone7>DataSvcUtil.exe /uri:http
://services.odata.org/OData/OData.svc /version:2.0 /DataServiceCollection /langu
age:CSharp /out:ODataDemo.cs
```

The `/uri` parameter points to the OData service we have requested for, `/DataServiceCollection` class contains the OData data feed, `/language` defines the supported programming language, `/out` is the output file of the client data service classes. To learn more see MSDN. To compile the project successfully make sure to include the **ODataDemo.cs** file in your project and add the reference **System.Data.Services.Client.dll** that comes with the `DataSvcUtil.exe` tool in OData client SDK for Windows Phone 7. If you open the **ODataDemo.cs** file you will see that it has created a `DemoService` class which we will use to execute query and also other classes like `Address`, `Category`, `Product`, and `Supplier`.

(C) Copyright Nokia 2014. All rights reserved.



Now we are ready to write some code to consume OData feed.

Adding UI controls on application page

The OData service that we have used has products of three different categories. To display the products according to their categories we added three PivotItem each of which has a ListBox in it. In the project, double-click the **MainPage.xaml** file and add the below code in root element.

```
    <!--LayoutRoot is the root grid where all page content is placed-->
     <Grid x:Name="LayoutRoot" Background="Transparent">
         <!--Pivot Control-->
         <controls:Pivot Title="ODATA SERVICE">
             <!--Pivot item one-->
             <controls:PivotItem Name="HeaderCat1" Header="Food">
                 <ListBox x:Name="ListBoxCat1" Margin="0,0,-12,0"
ItemsSource="{Binding}">
                     <ListBox.ItemTemplate>
                         <DataTemplate>
                             <StackPanel Margin="0,0,0,17" Width="480">
                                 <StackPanel Orientation="Horizontal"  Margin="0,0,0,0">
                                 <TextBlock Text="{Binding Path=Name}"
TextWrapping="NoWrap" Width="350"
                                        Style="{StaticResource
PhoneTextExtraLargeStyle}"/>
                                 <TextBlock Text="$" TextWrapping="NoWrap"
Margin="0,10,0,0"
                                        Style="{StaticResource PhoneTextLargeStyle}"/>
                                 <TextBlock Text="{Binding Path=Price}"
TextWrapping="NoWrap"  Margin="0,10,0,0"
                                        Style="{StaticResource PhoneTextLargeStyle}"/>
                                 </StackPanel>
                                 <TextBlock Text="{Binding Path=Description}"
TextWrapping="NoWrap"
                                        Margin="12,-6,0,0" Style="{StaticResource
PhoneTextSubtleStyle}"/>
                                 </StackPanel>
                         </DataTemplate>
                     </ListBox.ItemTemplate>
                 </ListBox>
             </controls:PivotItem>

             <!--Pivot item two-->
             <controls:PivotItem Header="Beverages">
                 <!--Triple line list no text wrapping-->
                 <ListBox x:Name="ListBoxCat2" Margin="0,0,-12,0"
ItemsSource="{Binding}">
                     <ListBox.ItemTemplate>
                         <DataTemplate>
                             <StackPanel Margin="0,0,0,17" Width="480">
                                 <StackPanel Orientation="Horizontal"  Margin="0,0,0,0">
                                 <TextBlock Text="{Binding Path=Name}"
TextWrapping="NoWrap" Width="350"
                                        Style="{StaticResource
PhoneTextExtraLargeStyle}"/>
                                 <TextBlock Text="$" TextWrapping="NoWrap"
```


(C) Copyright Nokia 2014. All rights reserved.

```
Margin="0,10,0,0"
                                        Style="{StaticResource PhoneTextLargeStyle}"/>
                                    <TextBlock Text="{Binding Path=Price}"
TextWrapping="NoWrap"  Margin="0,10,0,0"
                                        Style="{StaticResource PhoneTextLargeStyle}"/>
                                </StackPanel>
                                <TextBlock Text="{Binding Path=Description}"
TextWrapping="NoWrap"
                                        Margin="12,-6,0,0" Style="{StaticResource
PhoneTextSubtleStyle}"/>
                            </StackPanel>
                        </DataTemplate>
                    </ListBox.ItemTemplate>
                </ListBox>
            </controls:PivotItem>
            <controls:PivotItem Header="Electronics">
                <!--Triple line list no text wrapping-->
                <ListBox x:Name="ListBoxCat3" Margin="0,0,-12,0"
ItemsSource="{Binding}">
                    <ListBox.ItemTemplate>
                        <DataTemplate>
                            <StackPanel Margin="0,0,0,17" Width="480">
                                <StackPanel Orientation="Horizontal"  Margin="0,0,0,0">
                                    <TextBlock Text="{Binding Path=Name}"
TextWrapping="NoWrap" Width="350"
                                        Style="{StaticResource
PhoneTextExtraLargeStyle}"/>
                                    <TextBlock Text="$" TextWrapping="NoWrap"
Margin="0,10,0,0"
                                        Style="{StaticResource PhoneTextLargeStyle}"/>
                                    <TextBlock Text="{Binding Path=Price}"
TextWrapping="NoWrap"  Margin="0,10,0,0"
                                        Style="{StaticResource PhoneTextLargeStyle}"/>
                                </StackPanel>
                                <TextBlock Text="{Binding Path=Description}"
TextWrapping="NoWrap"
                                        Margin="12,-6,0,0" Style="{StaticResource
PhoneTextSubtleStyle}"/>
                            </StackPanel>
                        </DataTemplate>
                    </ListBox.ItemTemplate>
                </ListBox>
            </controls:PivotItem>
        </controls:Pivot>
    </Grid>
```

Consuming OData

- Open the **MainPage.xaml.cs** file and add the following `using` statements:

```
using ODataDemo;
using System.Data.Services.Client;
```

- Add the following declarations to the `MainPage` class:

```
        private DemoService context;
        private Uri demoserviceUri = new
Uri("http://services.odata.org/OData/OData.svc/");
        private DataServiceCollection<Product> productList1;
        private DataServiceCollection<Product> productList2;
        private DataServiceCollection<Product> productList3;
```

- Add the following `MainPage_Loaded` method to the `MainPage` class:

```
private void MainPage_Loaded(object sender, RoutedEventArgs e)
```


(C) Copyright Nokia 2014. All rights reserved.

```
        {
            context = new DemoService(demoserviceUri);
            productList1 = new DataServiceCollection<Product>(context);
            productList2 = new DataServiceCollection<Product>(context);
            productList3 = new DataServiceCollection<Product>(context);

            var query1 = "/Categories(0)/Products";
            var query2 = "/Categories(1)/Products";
            var query3 = "/Categories(2)/Products";

            productList1.LoadCompleted += new
EventHandler<LoadCompletedEventArgs>(productList1_LoadCompleted);
            productList2.LoadCompleted += new
EventHandler<LoadCompletedEventArgs>(productList2_LoadCompleted);
            productList3.LoadCompleted += new
EventHandler<LoadCompletedEventArgs>(productList3_LoadCompleted);

            productList1.LoadAsync(new Uri(query1, UriKind.RelativeOrAbsolute));
            productList2.LoadAsync(new Uri(query2, UriKind.RelativeOrAbsolute));
            productList3.LoadAsync(new Uri(query3, UriKind.RelativeOrAbsolute));
        }
```

`DemoService` creates the context object and points to the data service. `DataServiceCollection` object stores collection of `Products`. The `query` returns all the products of the specific category. `LoadCompleted` event is called when the data service request is completed. `LoadAsync()` execute the query asynchronously. To know more about query parameters see URI Conventions.

- Add the following `productList<x>_LoadCompleted` method to the `MainPage` class:

```
void productList3_LoadCompleted(object sender, LoadCompletedEventArgs e)
        {
            if (e.Error == null)
            {
                if (productList3.Continuation != null)
                {
                    productList3.LoadNextPartialSetAsync();
                }
                else
                {
                      this.ListBoxCat3.ItemsSource = productList3;
                }
            }
            else
            {
                MessageBox.Show(e.Error.Message);
            }
        }

void productList2_LoadCompleted(object sender, LoadCompletedEventArgs e)
        {
            if (e.Error == null)
            {
                if (productList2.Continuation != null)
                {
                    productList2.LoadNextPartialSetAsync();
                }
                else
                {
                    this.ListBoxCat2.ItemsSource = productList2;
                }
            }
            else
            {
                MessageBox.Show( e.Error.Message);
            }
        }

void productList1_LoadCompleted(object sender, LoadCompletedEventArgs e)
```


(C) Copyright Nokia 2014. All rights reserved.

```
        {
            if (e.Error == null)
            {
                if (productList1.Continuation != null)
                {
                    productList1.LoadNextPartialSetAsync();
                }
                else
                {
                    this.ListBoxCat1.ItemsSource = productList1;
                }
            }
            else
            {
                MessageBox.Show(e.Error.Message);
            }
        }
```

When `LoadCompleted` event is called `Continuation()` property of the `DataServiceCollection` object is checked and `LoadNextPartialSetAsync()` method of the `DataServiceCollection` object is called to load the next page feeds. The collection of `productList<x>` is finally bind to the `ListBox` item.

Summary

This application shows how to consume OData feeds using `DataSvcUtil.exe` tool in Windows Phone 7. This example might not work on Windows Phone 8. For Windows Phone 8 you can try OData Client Tools for Windows Phone Apps.

Source Code

- The full source code of the example is available here: File:ODataWindowsPhone7.zip

(C) Copyright Nokia 2014. All rights reserved.