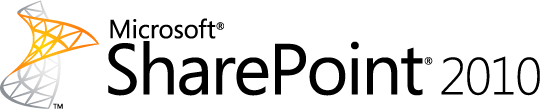
****

A SharePoint Developer Introduction

Hands-On Lab

Lab Manual

SPCHOL303 – Using Client OM and REST from a .NET Application – C#

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft. All rights reserved.

**Contents**

[SPCHOL303 – Using Client Object Model and REST from a .NET Application 2](#_Toc260141539)

[Lab Objective 2](#_Toc260141540)

[Additional Resources 2](#_Toc260141541)

[Getting Started 2](#_Toc260141542)

[Logging in to the Virtual Machine 2](#_Toc260141543)

[Locations 2](#_Toc260141544)

[Lab Pre-requisites 3](#_Toc260141545)

[Copying code samples from Word document 3](#_Toc260141546)

[Code Snippets 4](#_Toc260141547)

[Exercise 1 – Building a WPF Application using Client Object Model 5](#_Toc260141548)

[Task 1 – Create a new WPF .NET Application 5](#_Toc260141549)

[Task 2 – Using the Client Object Model 7](#_Toc260141550)

[Exercise 2 – Building a Windows Forms Application using SharePoint REST Services 10](#_Toc260141551)

[Task 1 – Creating a new Windows Forms Application 10](#_Toc260141552)

[Task 2 – Adding the SharePoint List WCF Data Service Reference 12](#_Toc260141553)

[Task 3 – Accessing SharePoint List Data 17](#_Toc260141554)

[Task 4 – Updating the SharePoint List Data 22](#_Toc260141555)

[Task 5 – Verifying the Updated Data 24](#_Toc260141556)

[Lab Summary 25](#_Toc260141557)

# SPCHOL303 – Using Client Object Model and REST from a .NET Application

Estimated time to complete this lab: **30 minutes**

Visual Studio 2010 and SharePoint Foundation 2010 are required for these exercises. These are installed on the Virtual Machine used in this lab.

## Lab Objective

Client Object Model provides a unified and complete object model for SharePoint to access SharePoint site data from remote clients. By including the Client Object Model library, developers can build SharePoint application as quickly as possible. It provides complete access to SharePoint site data.

The objective of this lab is to learn about:

* How to use the new Client Object Model.
* How to use the new REST API framework in SharePoint 2010.

## Additional Resources

This lab includes the following additional resources:

|  |  |  |
| --- | --- | --- |
| This Lab Manual | SPCHOL303*\_*Manual\_CS*.*docx | This document. |
| Source Code | Completed\CS\Ex1  Completed\CS\Ex2 | Completed lab source code in C#. |
| Resources | *Resources\CS* | Various other resources used throughout this lab. |

## Getting Started

### Logging in to the Virtual Machine

Please log into the virtual machine as the following user:

**Username**: Administrator

**Password**: pass@word1

### Locations

This Hands-On Lab contains a number of additional resources in fixed locations. By default, it is assumed that the base HOL directory is **C:\Content Packs\Packs\SharePoint 2010 Developer Labs 1.0\Supporting Files\SPCHOL303\Resources***.*

The default working folder for this lab is C**:\*SPHOLS\SPCHOL303***

### Lab Pre-requisites

1. Browse to base HOL directory ***Supporting Files\SPCHOL303\Resources***and execute the **optimize.ps1** PowerShell script:
   * Right click on **optimize.ps1** and select **Run with PowerShell:**

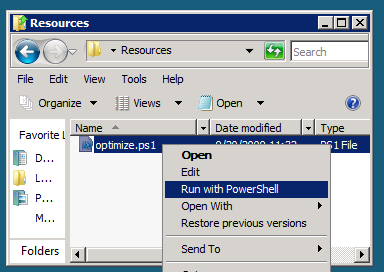


Figure 1 - Execute the PowerShell script

* + This will open a PowerShell window to execute the script. Please wait until the PowerShell script completes executing the script and closes the PowerShell window (this may take a few moments):

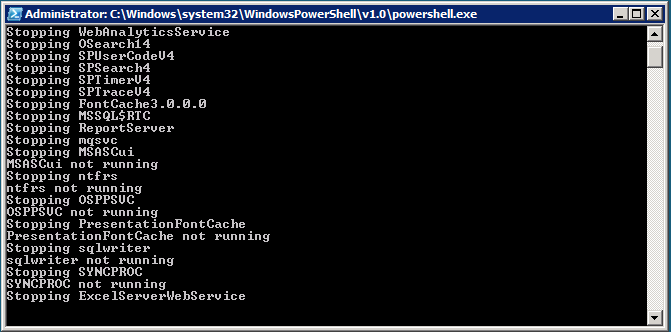


Figure 2 - PowerShell Window executing the script

### Copying code samples from Word document

Copying and pasting code from this Word document to Visual Studio is only possible for sections containing formatted code, e.g.:

Console.WriteLine("This is safe code!");

Code not in these sections may contain Unicode or other invisible characters that are not valid XML or C#/VB code, e.g.:

**Console.WriteLine(“This is NOT safe code !!”);**

### Code Snippets

You can also use Code Snippets to insert the appropriate code in the lab. To use the required code snippet for this lab:

* Right click on the code file where you want to insert the code snippet.
* Select **Insert Snippet:**

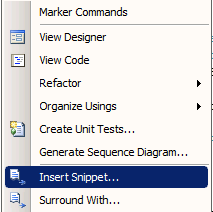
****

Figure 3 - Insert Code Snippet

* Select the required code snippet(s) from **My Code Snippets** gallery.

## Exercise 1 – Building a WPF Application using Client Object Model

In this exercise, you will create a WPF .NET application that uses the new SharePoint Client Object Model to access SharePoint data.

### Task 1 – Create a new WPF .NET Application

In this task you will write create a new WPF .NET Application.

1. Open **Visual Studio 2010** by going to **the Start Menu | All Programs | Microsoft Visual Studio 2010 | Microsoft Visual Studio 2010**.
2. From the menu, select **File | New | Project**.
3. In the **New Project** dialog window, choose **Visual C# | Windows** from the **Installed Templates**.
4. Select the **WPF Application** template from the list.

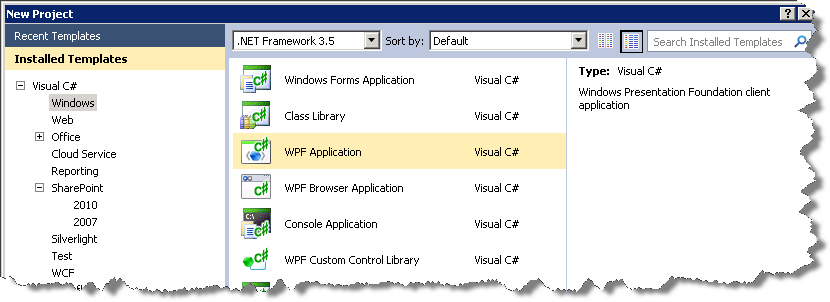


Figure 4 - WPF Application Project

1. In the Name textbox, enter **SPCHOL303-Ex1***.*
2. In the Location textbox, enter **C:\SPHOLS\SPCHOL303\CS\Ex1**.

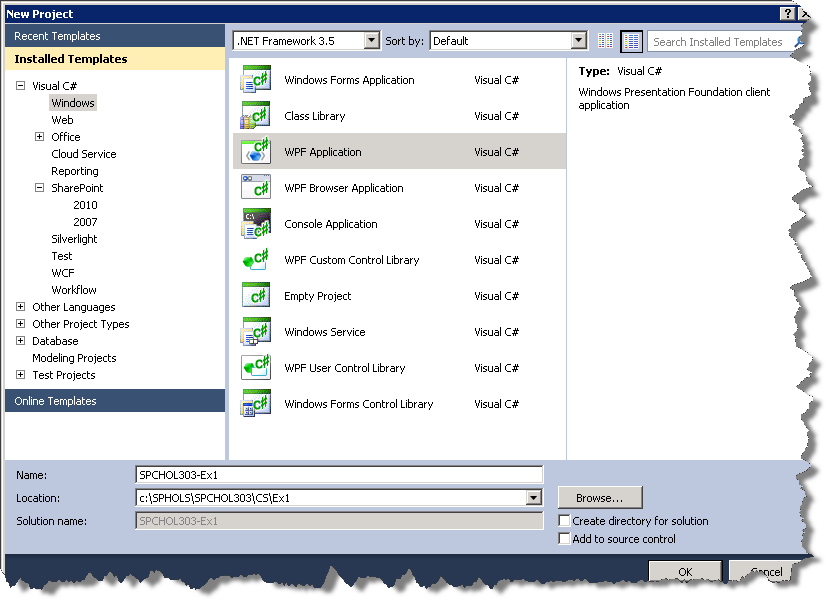


Figure 5 - Visual Studio New Project Dialog Window

1. Uncheck **Create directory for solution**.
2. Click **OK**.
3. Visual Studio will create the new **SPCHOL303-Ex1** project and add the necessary files.

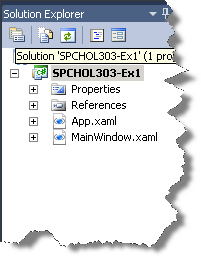


Figure 6 - Solution Explorer

### Task 2 – Using the Client Object Model

1. Right-click on the SPCHOL303-Ex1 in the Solution Explorer and select **Properties**.
2. In the Application tab, change the **Target Framework** property to *.NET Framework 4*.
3. In the Solution Explorer, right-click on **References** and select **Add Reference…**

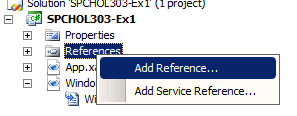


Figure 7 - Add Reference

1. Switch to **Browse** tab.
2. Type the following path in the File Name textbox and press **enter**:

**C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\ISAPI**

1. Select **Microsoft.SharePoint.Client.dll** and **Microsoft.SharePoint.Client.Runtime.dll**, and click **OK.**

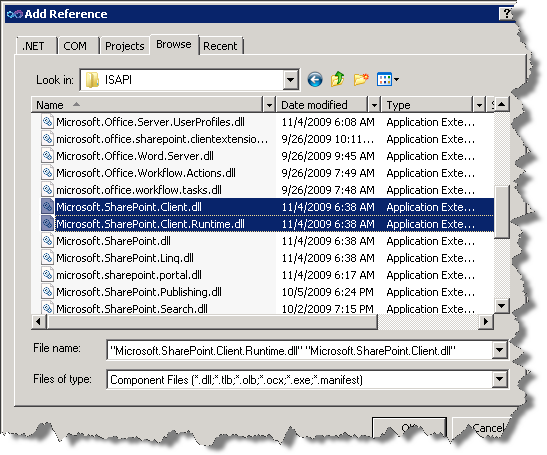


Figure 8 - Add Reference Dialog Window

1. In the Solution Explorer, double--click on Main**Window.xaml** to open the XAML view.
2. Insert the following XAML code inside the **Grid** element in the XAML view.

<ListBox Height="238" Margin="12,11,12,13" Name="ListBox1" Width="254" />

This XAML code creates a new ListBox control.

1. You should see the ListBox added to the designer view.

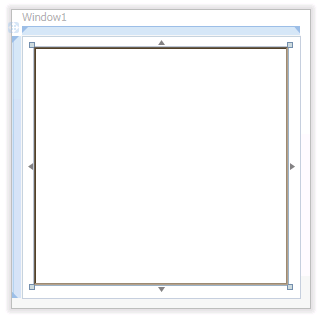


Figure 9 - Window1.xaml Designer View

1. In the Solution Explorer, right-click on Main**Window.xaml** and select **View Code.**



Figure 10 - View Code

1. Add the following **using statements**:

using Microsoft.SharePoint.Client;

using SP = Microsoft.SharePoint.Client;

**Code Snippets**: *My Code Snippets |* spchol303*\_ex1\_namespaces*

1. Insert the following code block in the **Window1** constructor after the **InitializeComponent()** method call:

ClientContext context =

new ClientContext("http://intranet.contoso.com");

Web site = context.Web;

context.Load(site, osite => osite.Title);

context.ExecuteQuery();

Title = site.Title;

ListCollection lists = site.Lists;

IEnumerable<SP.List> listsCollection =

context.LoadQuery(lists.Include(l => l.Title, l => l.Id));

context.ExecuteQuery();

ListBox1.ItemsSource = listsCollection;

ListBox1.DisplayMemberPath = "Title";

**Code Snippet:** *My Code Snippets |* spchol303*\_ex1\_constructor*

This code block:

* Creates a new client context to the SharePoint site.
* Queries the available SharePoint lists in the site and retrieves the lists along with their properties – **Title** and **Id.**
* Binds the list collection to the WPF ListBox control.

1. Press **F5** to start debugging the application.
2. You should see the Application Window with the ListBox populated with the available SharePoint lists.

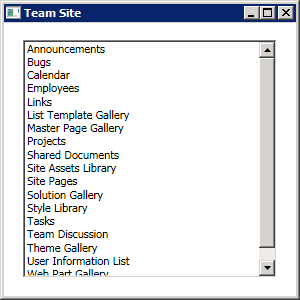


Figure 11 - Application Window

1. Close the Application Window.

In the past few minutes, you have demonstrated how to use the new SharePoint Client Object Model and create a WPF .NET Application to retrieve SharePoint data from a site.

## Exercise 2 – Building a Windows Forms Application using SharePoint REST Services

In this exercise, you will access and update data in a SharePoint list using the SharePoint List WCF Data Service from a Windows Forms Application.

### Task 1 – Creating a new Windows Forms Application

In this task, you will create a new Windows Forms Application.

1. Open **Visual Studio 2010** by going to **the Start Menu | All Programs | Microsoft Visual Studio 2010 | Microsoft Visual Studio 2010**.
2. From the menu, select **File | New | Project**.
3. In the **New Project** dialog window, choose **Visual C# | Windows** from the **Installed Templates**.
4. Select **Windows Forms Application** from the Project Templates.

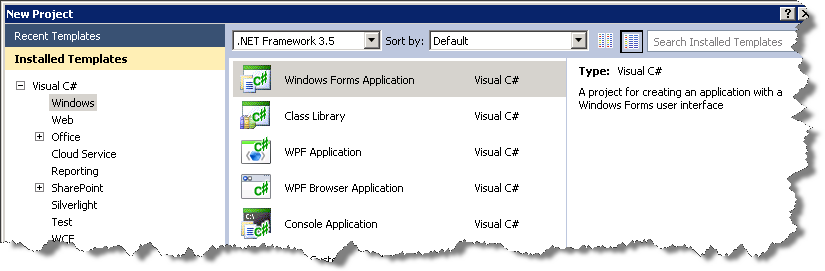


Figure 12 - New Windows Forms Application Project

1. In the **Name** textbox, enter **SPCHOL303-Ex2**.
2. In the **Location** textbox, enter **C:\SPHOLS\SPCHOL303\CS\Ex-2***.*.

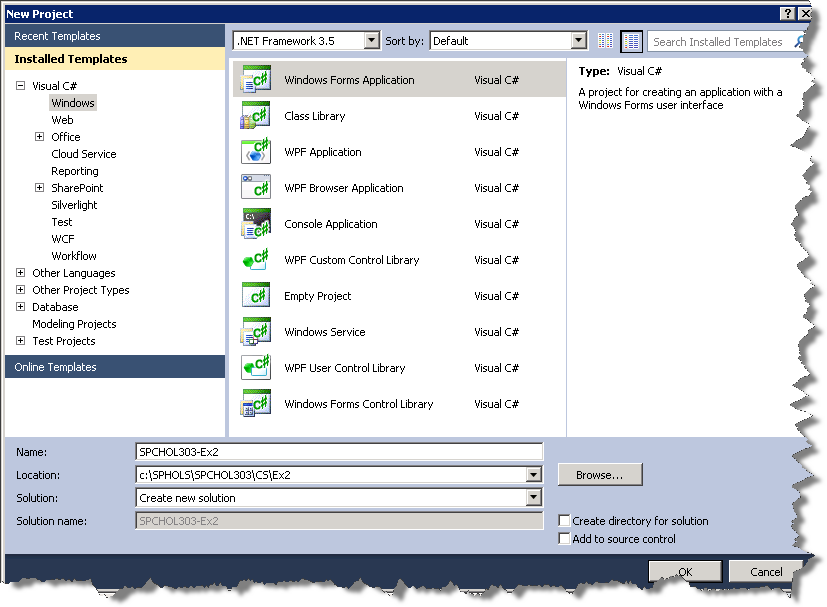


Figure 13 - Visual Studio New Project Dialog Window

1. Uncheck **Create directory for solution**.
2. Click **OK**.
3. Visual Studio will create the new **SPCHOL303-Ex2** project and add the necessary files.

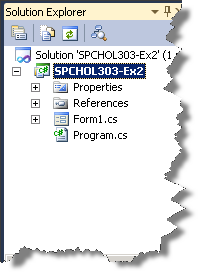


Figure 14 - SPCHOL303-Ex2 Solution Explorer

### Task 2 – Adding the SharePoint List WCF Data Service Reference

In this task, you will add a reference to the new SharePoint ADO.NET List Data Services to access the Employees List data from the local SharePoint site.

1. From the Visual Studio top menu bar, select **Data | Add New Data Source…**

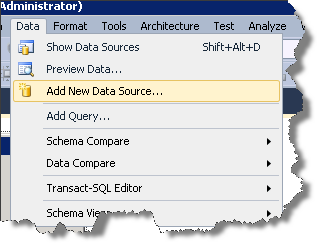


Figure 15 - Add New Data Source

1. In the **Data Source Configuration Wizard** select **SharePoint** and click **Next >** button.

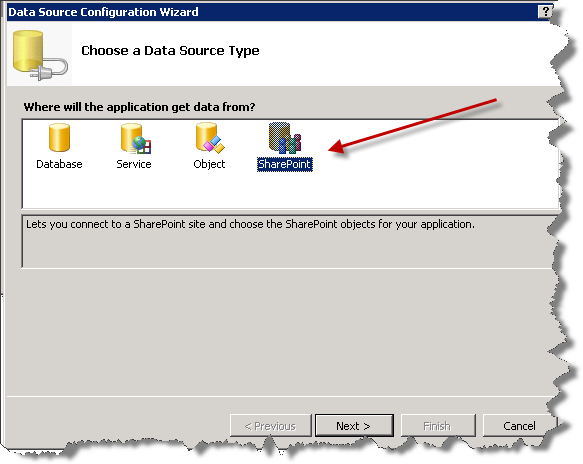


Figure 16 - Data Source Configuration Wizard

1. In the Add Service Reference dialog window, type the following URL in the Addresstextbox and click **Go**:

**http://intranet.contoso.com/\_vti\_bin/ListData.svc**

1. You should see a service added in the left-hand **Services** paneof the dialog window (name may vary), and the **Namespace** textbox should display ServiceReference1, as in this figure:

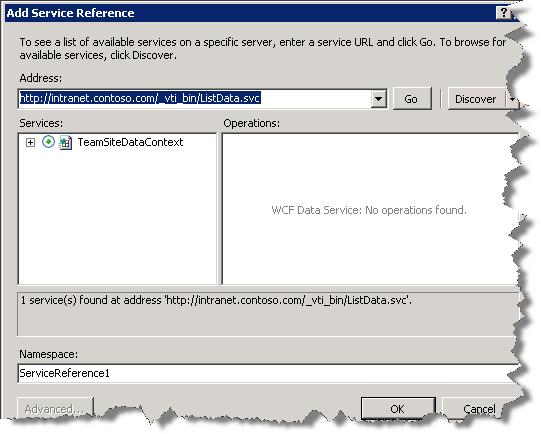


Figure 17 - List Data Service

SharePoint web services reside in the **\_vti\_bin** virtual directory. The **ListData.svc** is the SharePoint List Data service that uses the REST API. Using this web service, you can work with the SharePoint list data via the REST interface.

1. Click and expand on **TeamSiteDataContext** to explore the various service elements:

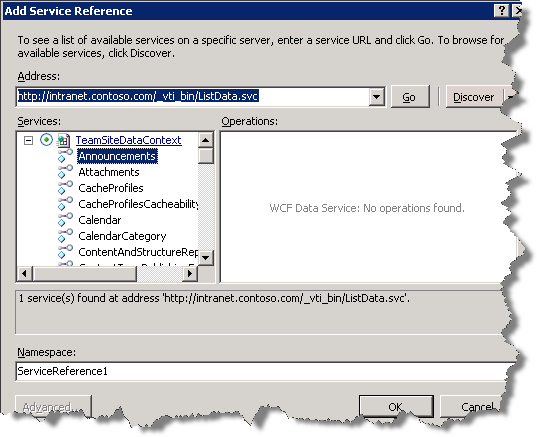


Figure 18 - Service Elements

1. Click **OK**.
2. Click the **Finish** button in the **Data Source Configuration Wizard.**

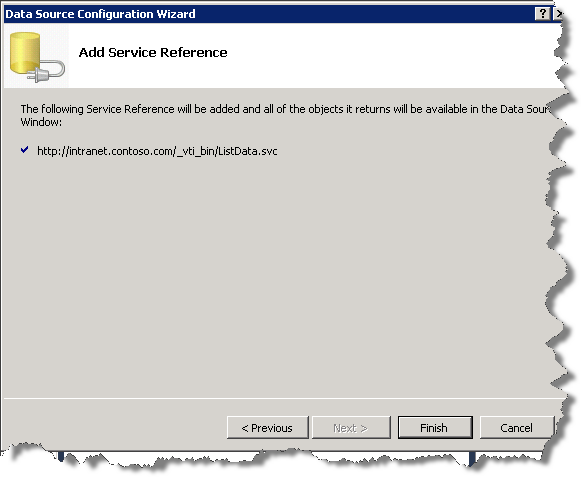


Figure 19 - Service Reference added

1. This will add the service reference to the solution:

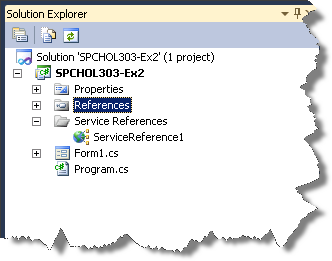


Figure 20 - Service Reference

### Task 3 – Accessing SharePoint List Data

1. In this task, you will access and retrieve the Employees list data using the SharePoint List WCF Data Service.
2. Right click on **References** and select **Add Reference...**
3. In the Add Reference dialog window, switch to the **.NET**  tab and select **WindowsBase**
4. Click **OK.**
5. Double-click on **Form1.cs** in the Solution Explorer. This will bring up the Form Designer.
6. From the Visual Studio 2010 menu, click on **Data | Show Data Sources**

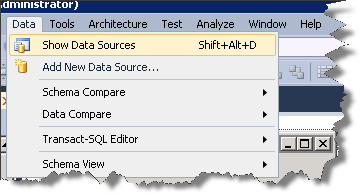


Figure 21 - Show Data Sources

1. This will open the Data Sources window showing the service reference we added in **Task 2** along with available SharePoint Lists in the site. The data source consists of the SharePoint lists and document libraries in the form of entity data model. Click on the pin icon to lock the list in an open position.

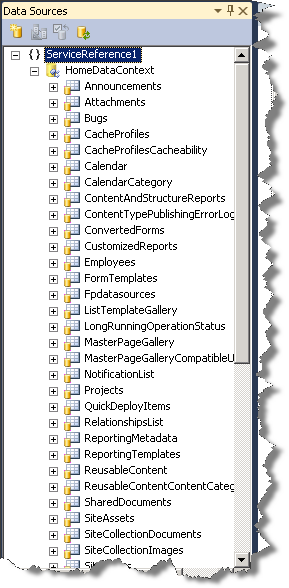


Figure 22 - Data Sources Window

1. Drag and drop **Employees** from the **Data Sources** window to the **Form** window. This will insert the **Employees DataGrid** into the Form. Adjust the Form and DataGrid to suit the width and height respectively:

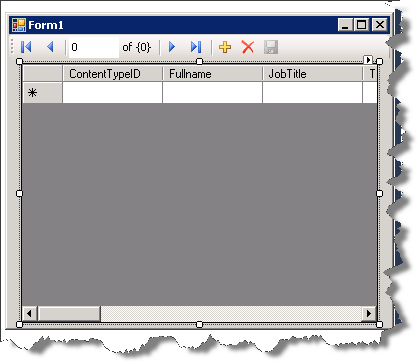


Figure 23 – Windows Form Window and Employees DataGrid

1. Right-click anywhere on the empty **Employees DataGrid** and select **Edit Columns.**

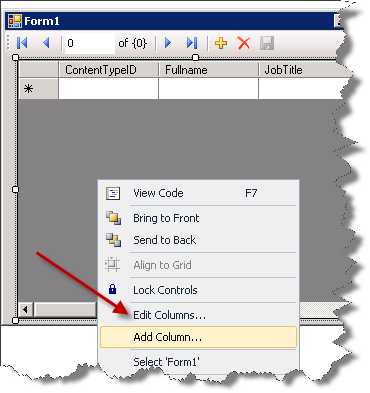


Figure 24 - Edit Columns

1. In the Edit Columns window, remove every column except the following three. To remove a column, select a column from the **Selected Columns** list and click the **Remove** button:
   * Fullname
   * Job Title
   * Team

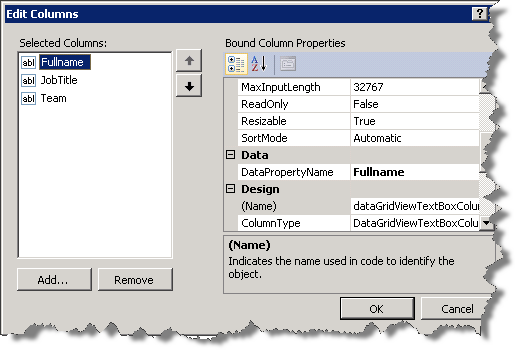


Figure 25 - Edit Columns Window

1. Click **OK.**
2. You should see the DataGrid updated with the changes:

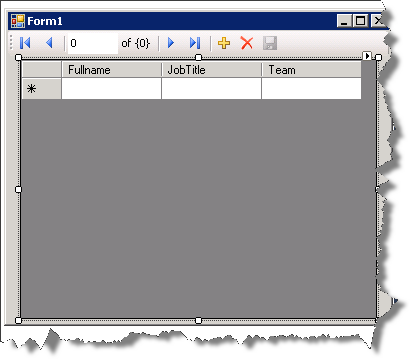


Figure 26 - Updated DataGrid

1. **Double-click** on the title bar in the Designer (where it says Form1) to add the **Form1\_Load** method (this will open the Code view):

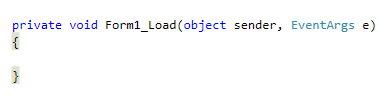


Figure 27 - Form1\_Load Method

1. Add the following using statement in the code behind:

using SPCHOL303\_Ex2.ServiceReference1;

using System.Net;

**Code Snippet**: *My Code Snippets |* spchol303*\_ex2\_namespaces*

1. Insert the following code just after the **Form1** class declaration:

TeamSiteDataContext context =

new TeamSiteDataContext(new Uri("http://intranet.contoso.com/\_vti\_bin/ListData.svc"));

**Code Snippet**: *My Code Snippets |* spchol303*\_ex2\_context*



Figure – TeamSiteDataContext

1. Insert the following code in the **Form1\_Load** method:

context.Credentials = CredentialCache.DefaultCredentials;

employeesBindingSource.DataSource = context.Employees;

**Code Snippet**: *My Code Snippets |* spchol303*\_ex2\_form\_load*

1. Press **F5** to start debugging the application.
2. You should see the application window with the DataGrid populated with Employees list:

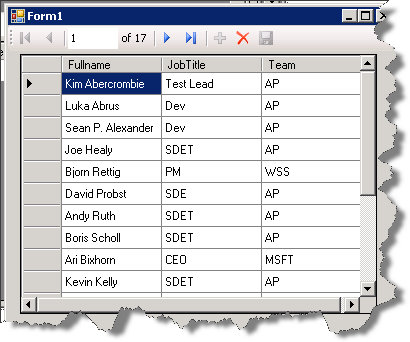


Figure 29 - Application Window

1. **Close** the application window.

In the past few minutes, you have demonstrated how to access the SharePoint List Data using the new SharePoint List WCF Data Service.

### Task 4 – Updating the SharePoint List Data

In this task, you will update an employee’s Fullname and save the changes back to the SharePoint site.

1. In the Solution Explorer, right-click on **Form1.cs** and select **View Designer.**
2. In the Employees DataGrid, click on the **BindingNavigator.**



Figure 30 - BindingNavigator

1. In the BindingNavigator, right-click on the Save button () and select **Enabled.**

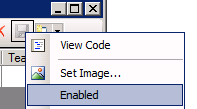


Figure 31 - Enable Save Button

1. Double-click on the **Save** button in the **BindingNavigator** to generate the Save method:

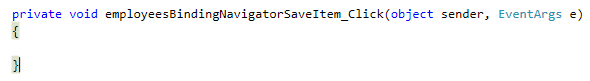


Figure 32 - Save Method

1. Insert the following code in the **employeesBindingNavigatorSaveItem\_Click** method:

context.SaveChanges();

**Code Snippet**: *My Code Snippets |* spchol303*\_ex2\_save\_method*

1. In the Solution Explorer, right-click on **Form1.cs** and select **View Designer.**
2. At the bottom of the designer window, right-click on **employeesBindingSource** and select **Properties:**

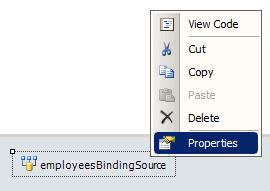


Figure 33 - employeesBindingSource Properties

1. In the **Properties** window, click on **Events** icon (  )
2. Double-click on **CurrentItemChanged** event. Visual Studio will generate the item changed event for the **employeesBindingSource:**

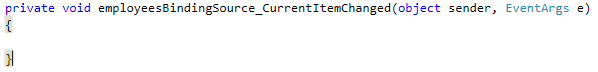


Figure 34 - employeesBindingSource\_CurrentItemChanged Method

1. Insert the following code in the **employeesBindingSource\_CurrentItemChanged** method:

context.UpdateObject(employeesBindingSource.Current);

**Code Snippet**: *My Code Snippets |* spchol303*\_ex2\_item\_changed*

1. Press **F5** to start debugging the application.
2. You should see the application window with the list of Employees.
3. Click on the first employee and change the **Fullname** from **Kim Abercrombie** to **Kim:**

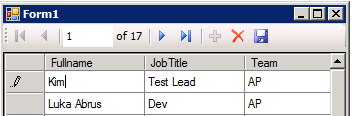
****

Figure 35 - Change the Fullname

1. Press **Enter** key and click the **Save** button ()
2. The changes to the selected employee’s **Fullname** should now be saved back to the SharePoint site.
3. Close the application window.

### Task 5 – Verifying the Updated Data

In this task, you will verify the updated employee’s data performed in **Task 4.**

1. Open Internet Explorer and browse to the following SharePoint site:

<http://intranet.contoso.com>

1. If prompted for authentication, enter the following details:

**Username**: Administrator

**Password**: pass@word1

1. In the Left Navigation, click on **Employees** list to open the Employees List:

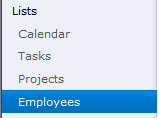


Figure 36 - Left Navigation

1. You should see the updated employee **Kim:**



Figure 37 - Updated Employee List Item

In the past few minutes you demonstrated how to update a List Item from a client application and push back the changes to the SharePoint site.

## Lab Summary

In this lab you performed the following exercises.

Building a WPF Application using the Client Object Model.

Building a Windows Forms Application using the SharePoint REST Services.

Updating a List Item from the Windows Forms Application and saving the changes back to the SharePoint site.

In this lab, you learned how to write Windows Client Applications that access SharePoint list data using the new Client Object Model and the SharePoint List WCF Data Service.