

Developing a BCS External Content Type with Visual Studio 2010

Hands-On Lab

Lab Manual

SPCHOL304 – Developing a BCS External Content Type with Visual Studio 2010 – 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**

[SPCHOL304 - Developing a BCS External Content Type with Visual Studio 2010 2](#_Toc259796760)

[Lab Objective 2](#_Toc259796761)

[Additional Resources 2](#_Toc259796762)

[Getting Started 2](#_Toc259796763)

[Logging in to the Virtual Machine 2](#_Toc259796764)

[Locations 2](#_Toc259796765)

[Lab Pre-requisites 3](#_Toc259796766)

[Copying code samples from Word document 3](#_Toc259796767)

[Code Snippets 4](#_Toc259796768)

[Exercise 1 – Build a BCS External Content Type 5](#_Toc259796769)

[Task 1 – Create a Business Data Catalog Model project 5](#_Toc259796770)

[Task 2 – Update the Schema of the Entity Class 7](#_Toc259796771)

[Task 3 – Update the Service Code 17](#_Toc259796772)

[Task 4 – Build and Deploy 19](#_Toc259796773)

[Exercise 2 – Using the External Content Type from Outlook with a Custom Outlook form 22](#_Toc259796774)

[Task 1 – Configure the External Content Type for offline use 22](#_Toc259796775)

[Task 2 – Taking the list offline 32](#_Toc259796776)

[Task 4 – Remove the external list 34](#_Toc259796777)

[Lab Summary 35](#_Toc259796778)

# SPCHOL304 - Developing a BCS External Content Type with Visual Studio 2010

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

## Lab Objective

This lab demonstrates how to create a Business Connectivity Service External Content Type and then make it available offline in Outlook.

The objective of this lab is to learn how to build BCS External Content Types and make them available offline in Outlook using Visual Studio 2010 and SharePoint Designer.

* Build a BCS External Content Type
* Create a Business Data Catalog Model project
* Configure the External Content Type for offline use
* Open the list using Outlook

## Additional Resources

This lab includes the following additional resources:

|  |  |  |
| --- | --- | --- |
| This Lab Manual | SPCHOL304\_Manual\_CS.docx | This document |
| Source Code | Completed\CS\Ex1 | Completed lab source code. |
| Resources | Resources\CS\Ex1 | Various 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\SPCHOL304\Resources.**

The default working folder for this lab is **C:\SPHOLS\SPCHOL304**.

### Lab Pre-requisites

Browse to base HOL directory ***Supporting Files\SPCHOL304\Resources***and execute the **optimize.ps1** PowerShell script:

1. Right-click on **optimize.ps1** and select **Run with PowerShell**:

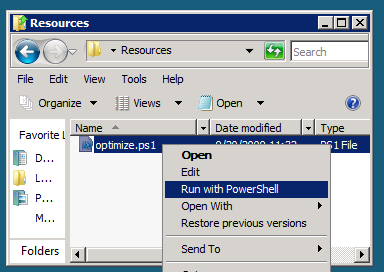


Figure 1 - Execute the PowerShell script

1. 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):

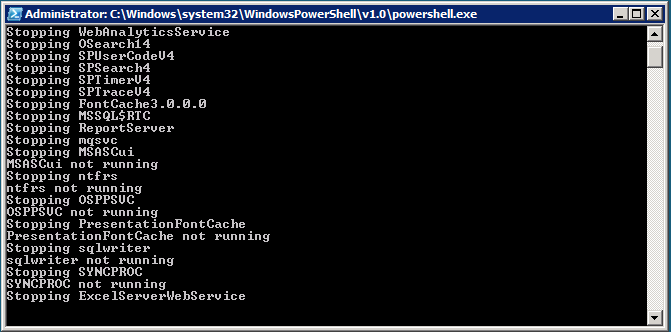


Figure 2 - PowerShell Window executing the script

### Copying code samples from Word document

Copying and pasting code from this Word document to Visual Studio is only safe for the sections of 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:**

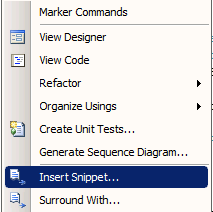


Figure 3 - Visual Studio Code Context Menu.

* Select code snippets from **My Code Snippets** gallery.

## Exercise 1 – Build a BCS External Content Type

This exercise will guide you through creating an external content type using the Visual Studio 2010 BDC Model project template and extend the external content type.

### Task 1 – Create a Business Data Catalog Model project

1. Open **Visual Studio 2010** by going to **Start Menu | All Programs | Microsoft Visual Studio 2010 | Microsoft Visual Studio 2010,** and from the menu select **File | New | Project**.
2. Select the **Visual C# | SharePoint | 2010 |** **Business Data Connectivity Model** project template.
3. Enter **BDCEx1** in the Name textbox.

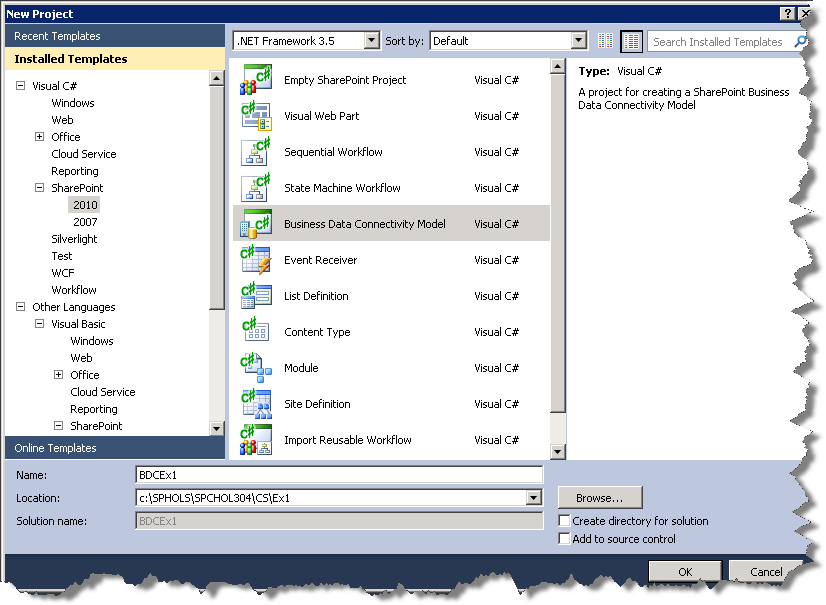


Figure 4 - New Project Dialog

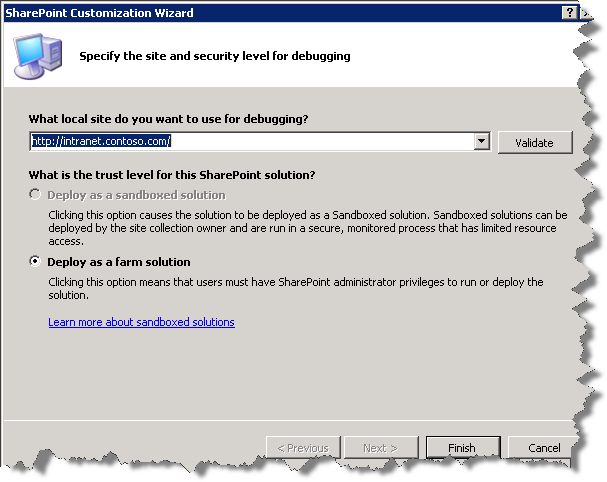
1. Enter **C:\SPHOLS\SPCHOL304\CS\Ex1** in the Location textbox.
2. Click **OK**.
3. Change the URL to [**http://intranet.contoso.com**](http://intranet.contoso.com).
4. **Select** the **Deploy as a farm solution** radio button.  
     
   

Figure 5 - SharePoint Customization Wizard

1. Click **Finish**.

### Task 2 – Update the Schema of the Entity Class

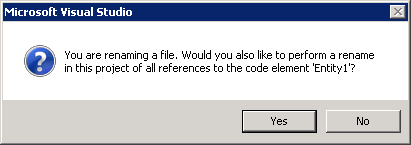
1. From the top menu, select **View | Solution Explorer**. In the **Solution Explorer,** rename Entity1.csto **Customer.cs** andEntity1Service.csto **CustomerService.cs**. Click **Yes** when prompted with the below:  
     
   

Figure 6 - Visual Studio file rename prompt

1. If it’s not already open, double-click on **Customer.cs** in the Solution Explorer. Replacethe code in the body of the **Customer** class with the following code. This is adding two extra properties and changing the type of a third as a mapping for the type descriptors we added and modified in the BDC definition.  
     
   Note when you change the Identifier1 to CustomerId, you can use the glyph that appears in the editor to rename all references to the field Identifier1. This doesn’t help you in this case but is a good practice.

public Int32 CustomerId { get; set; }

public string Message { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

**Code Snippet:** *My Code Snippets | spchol304\_ex1\_customer*

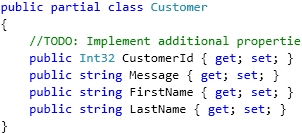


Figure 7 - Customer class after changes to the body

**Task 3 – Extend the BCS Metadata for the External Content Type**

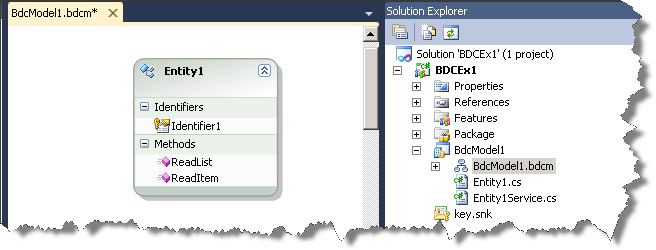
1. In the Solution Explorer, open the **BdcModel1.bdcm** if the BDC designer isn’t already open.  
     
   

Figure 8 - Solution Explorer displaying BDCModel1.bdcm in the BDC Designer.

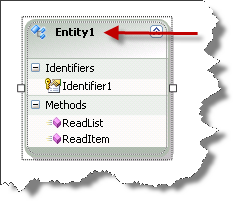
1. In the designer select the **Entity1** entity by clicking on the title.  
     
   

Figure 9 - Entity1 entity

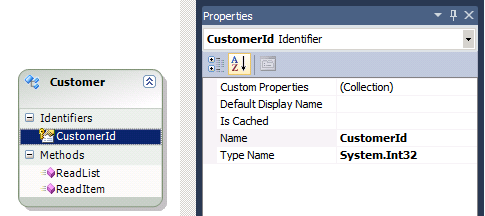
1. Rename the **Entity1** to ***Customer***by clicking on the **Entity1** name in the designer and pressing F2 (alternatively you can change the name in the **Property Grid**).
2. Highlight **Identifier1** in the Entity designer and press **F4** to open the Properties Grid. Use the **Properties Grid** to renameIdentifier1 to **CustomerId** and set itsType Nameto **System.Int32*.***

Figure 10 - Properties panel after the Type Name change

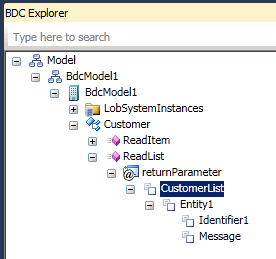
1. In the BDC Explorer in Visual Studio, expand **Model | BDCModel1 | BDCModel1 | Customer | ReadList | returnParameter**. Click Entity1List to open its Property Grid.  
     
     
   

Figure 11 - BDC Explorer highlighting Entity1List

1. Using the Properties grid,rename **Entity1List** to **CustomerList***.*
2. In the BDC Explorer, expand **CustomerList**.

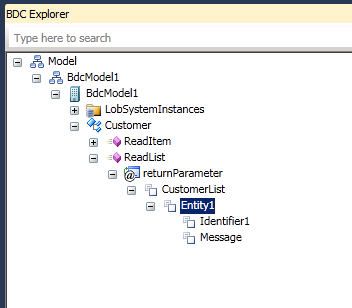


Figure 12 - BDC Explorer highlighting Entity1

1. Using the **Properties** grid, rename **Entity1** to **Customer.**

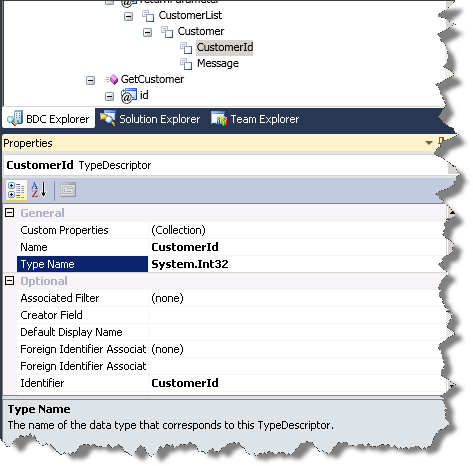
In the **BDC Explorer,** expand **Customer** and using the **Properties** grid renameIdentifier1to **CustomerId,** set theTypeNameto **System.Int32.**

Figure 13 - Properties grid after changes to Identifier1

1. In the **BDC Explorer** right-click **Customer** and select **Add Type Descriptor**.  
     
   

Figure 14 - Customer entity context menu

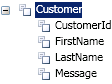
1. Select **TypeDescriptor** and, using the **Properties** grid, change Name to **FirstName**.
2. In the **BDC Explorer** right-click **Customer** and select **Add Type Descriptor**.
3. Select **TypeDescriptor** and, using the **Properties** grid, change Name to **LastName**.  
     
   

Figure 15 - Customer entity in BDC Explorer after the addition of FirstName and LastName type descriptors

1. In the **BDC Explorer** right-click **Customer** and select **Copy**.  
     
   

Figure 16 - Customer entity context menu

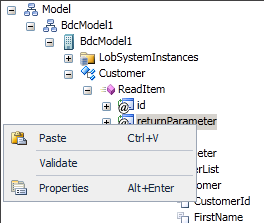
1. In the **BDC Explorer** expand the **ReadItem** method, right-click **returnParameter** and select **Paste**.  
     
   

Figure 17 – returnParameter context menu

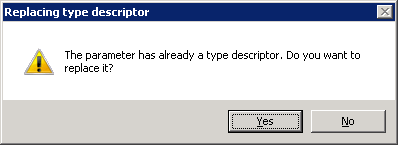
1. Click **Yes**.  
     
   

Figure 18 - Replace Type Descriptor prompt

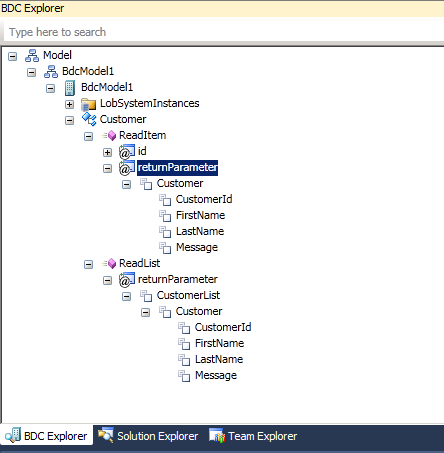
1. Expand **returnParameter | Customer** and confirm *FirstName* and *LastName* type descriptors have successfully copied.  
     
    

Figure 19 - BDC Explorer after Task 1’s, Step 1 to Step 18 changes

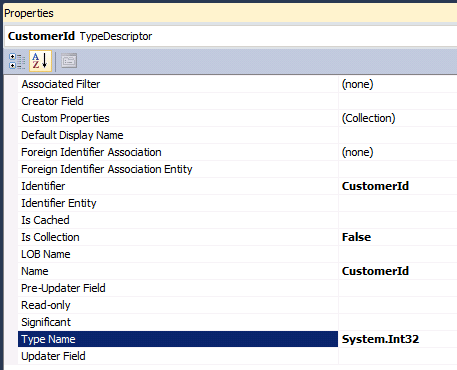
1. In the **BDC Explorer** expand the **ReadItem | id** and select **Identifier1**.
2. Using the **Properties** grid, renameIdentifier1to **CustomerId**,set theTypeNameto **System.Int32**.  
    

Figure 20 - Properties grid after changes to Identifier1

1. Using the BDC Method Details pane add an update method

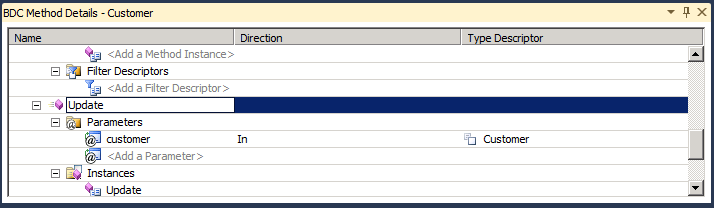


Figure 21 - BDC Method Details adding Update method

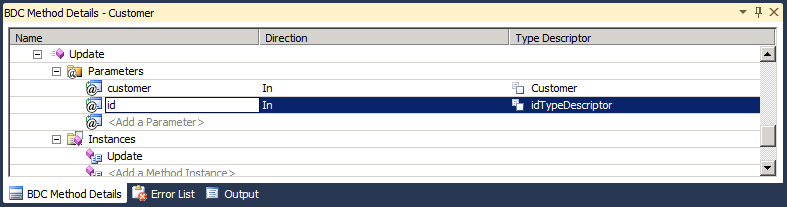
1. Add a parameter and name it id  
     
   

Figure 22 - Adding the id parameter to the Update method

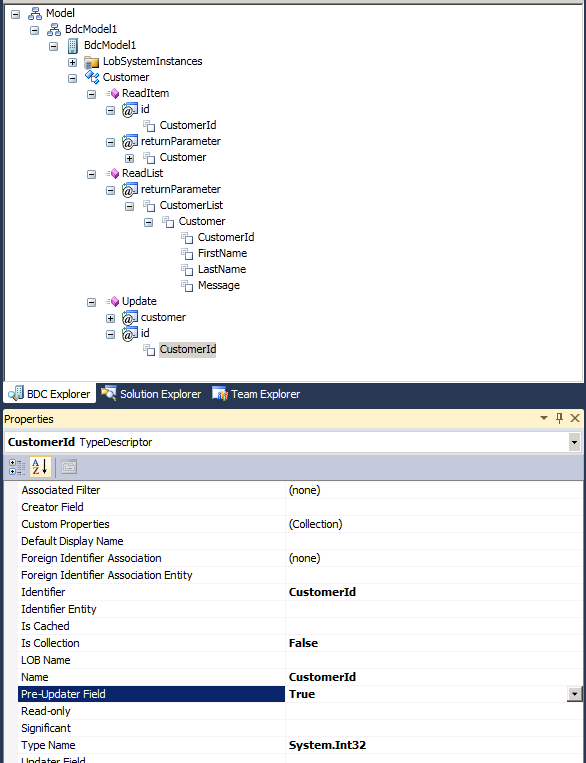
1. Open the new Update method in the **BDC Explorer** window
2. Copy the type descriptor from the **CustomerId** type description from **@id** parameter in the **ReadItem** method to the @id parameter in the **Update** method
3. Change the Pre-Updater property on the id parameter to be **True**.  
     
   

Figure 23 - Updating the Pre-Updater Field to True

## Task 3 – Update the Service Code

1. In the **Solution Explorer** open the **CustomerService.cs**.
2. **Replace** the body of the **CustomerService** class with the following code. This will return a customer based on a given id, and an array of customers if requested.

public static Dictionary<Int32, Customer> d = null;

public static Customer ReadItem(Int32 id)

{

// take a copy for SharePoint

Customer c = new Customer();

Customer e = d[id];

c.CustomerId = e.CustomerId;

c.FirstName = e.FirstName;

c.LastName = e.LastName;

c.Message = e.Message;

return c;

}

public static IEnumerable<Customer> ReadList()

{

// this is usually the first method called so check for null

if (d == null)

{

d = new Dictionary<Int32, Customer>();

for (int i = 0; i < 10; i++)

{

Customer e = new Customer();

e.CustomerId = i;

e.Message = i + " Item Data";

e.FirstName = i + " First Name";

e.LastName = i + " Last Name";

d.Add(i, e);

}

}

return d.Values;

}

public static void Update(Customer customer, Int32 id)

{

d[id].FirstName = customer.FirstName;

d[id].LastName = customer.LastName;

d[id].Message = customer.Message;

}

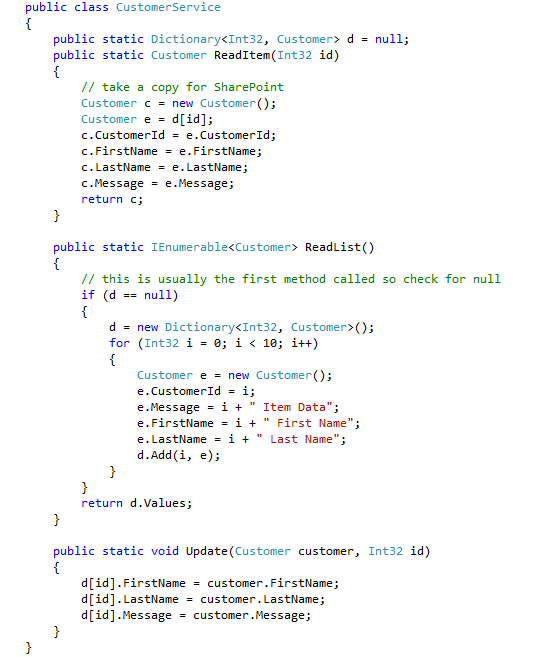
**Code Snippet:** *My Code Snippets | spchol304\_ex1\_customerservice* 

Figure 24 - CustomerService class after the changes

1. In the Solution Explorer, expand the Features folder and double click the Feature1.feature node. This opens the feature designer.
2. Click the **Manifest** tab at the bottom of the feature designer.
3. Click the + glyph next to **Edit Options** to open the template editor.
4. Add a new Property for the Site URL

<Property Key="SiteUrl" Value="http://intranet.contoso.com" />

The feature designer should now look like Figure 25.

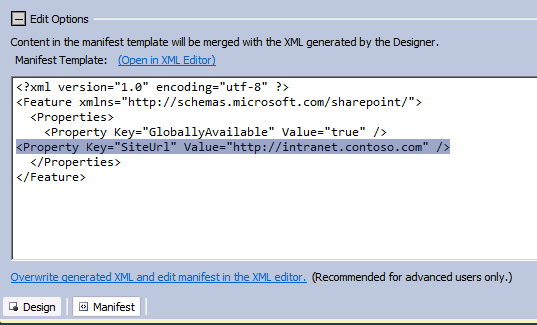


Figure 25 – The feature designer with the SiteUrl property added

### Task 4 – Build and Deploy

1. From the menu, select **Build | Deploy BDCEx1**.
2. Open <http://intranet.contoso.com> in a browser session.
3. Click on **Lists** in the left hand navigation  
     
   

Figure 26 - Lists link

1. Click on **Create**.  
     
   

Figure 27 - Create link

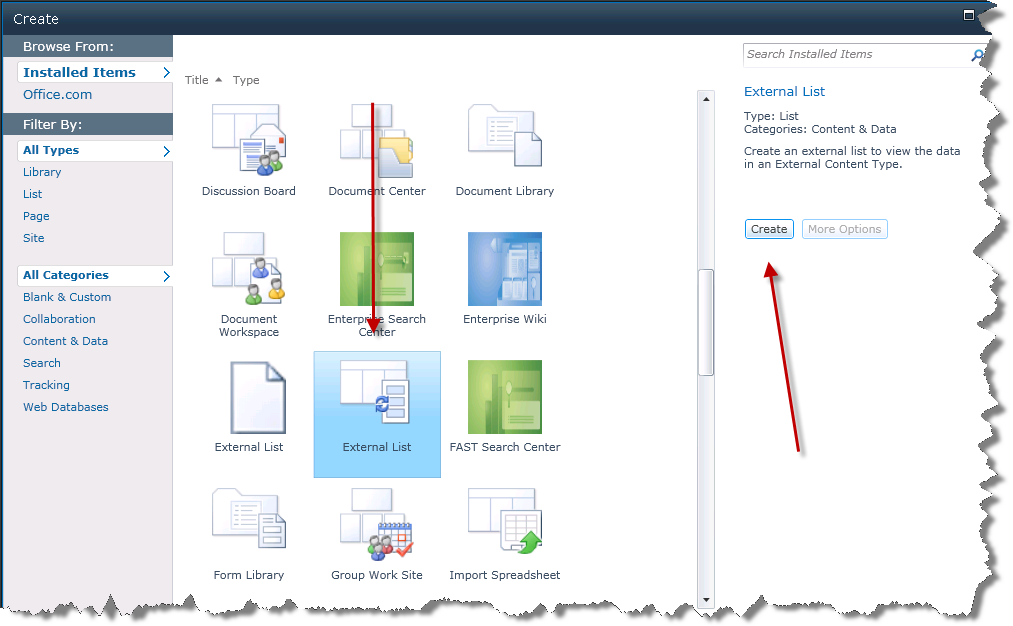
In the Create dialog, select **External List** and select **Create.**  


Figure 31 – Create External List dialog

1. Enter **My Contacts** In the Name textbox.
2. Click on the second button to the right of the External Content Type text box (the one with the tooltip of “Select External Content Type”).

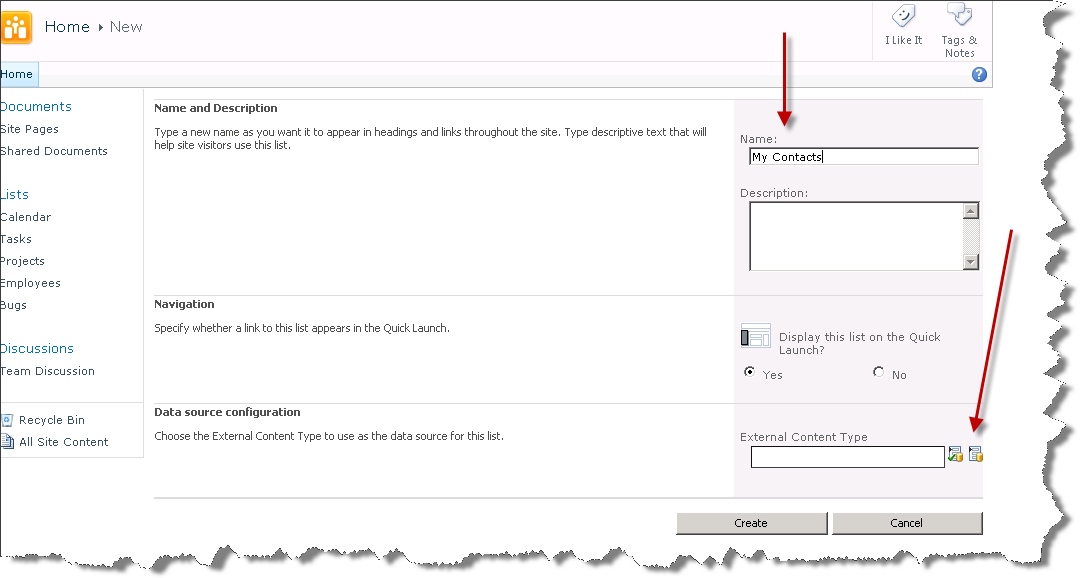


Figure 32 – Create new external list page

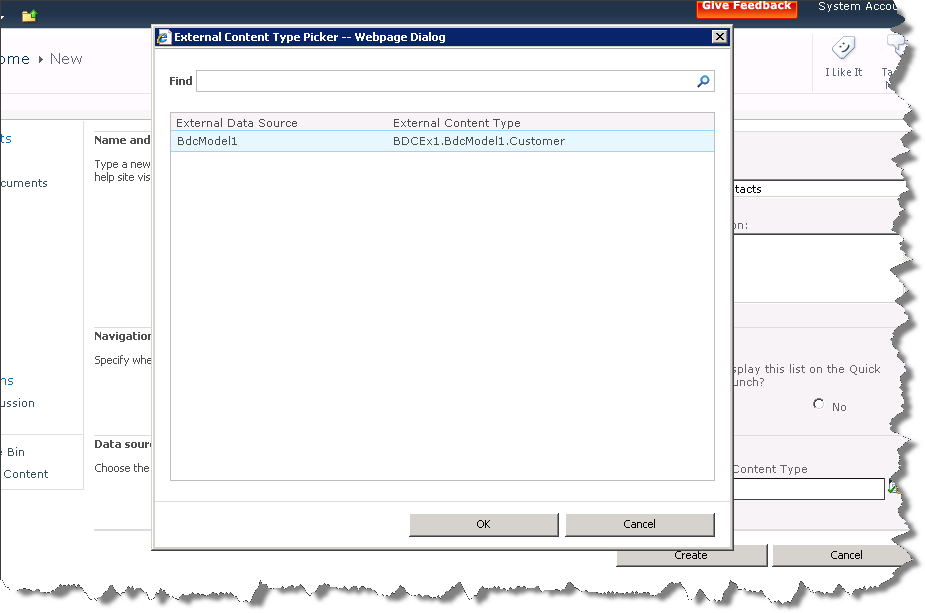
1. Select **BDCEx1.BDCModel1.Customer** and click **OK**.  
     
    

Figure 33 - Business Data Type Picker -- Webpage Dialog

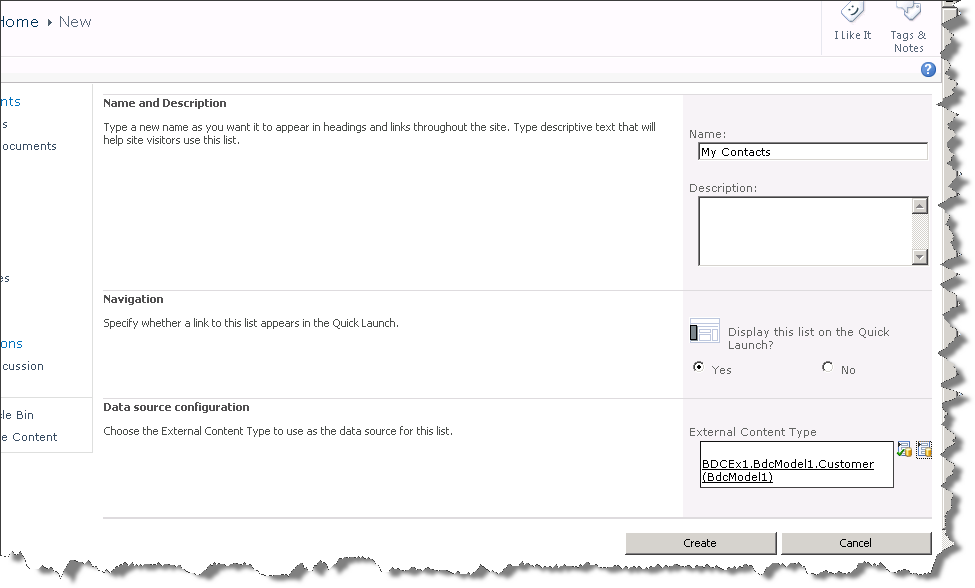


Figure 28 - New external list page

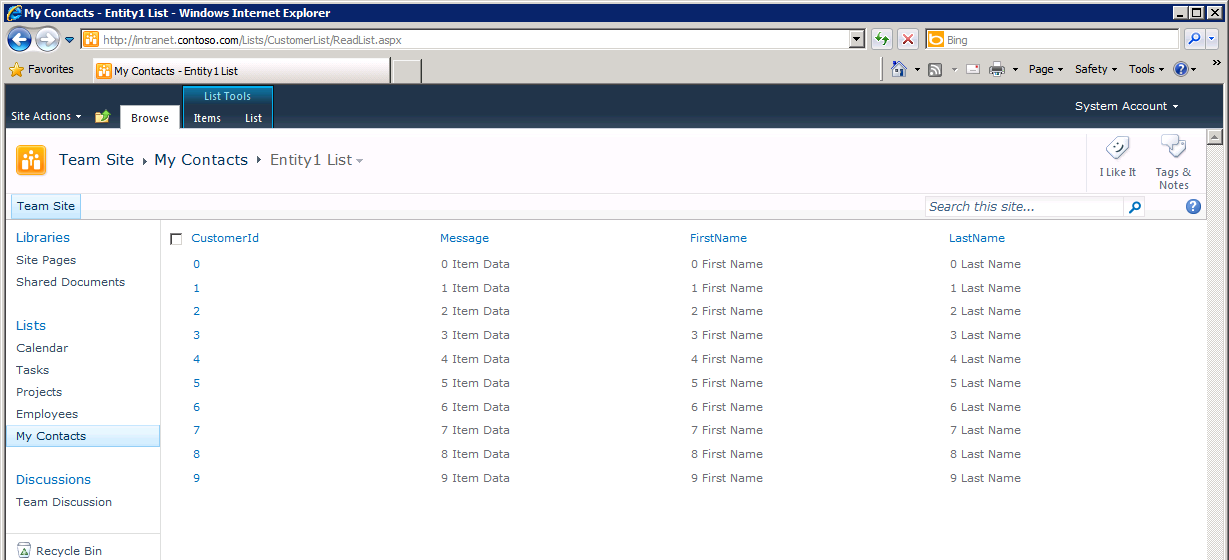
1. Click **Create**.
2. You have just created an external content type and a list to host that content type.  
     
   

Figure 29 - External Content Type (BDC model) displayed in list

## Exercise 2 – Using the External Content Type from Outlook with a Custom Outlook form

This task will guide you through making the external content type created in Exercise 1 available in Outlook using SharePoint Designer 2010.

### Task 1 – Configure the External Content Type for offline use

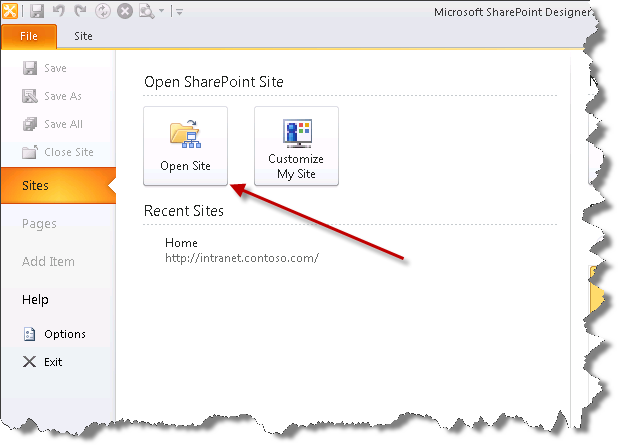
1. Open **SharePoint Designer 2010** by going to **Start | All Programs | SharePoint | Microsoft SharePoint Designer 2010**
2. Click on **Sites** and then **Open Site**.  
     
   

Figure 30 - Sites menu button in SharePoint Designer 2010

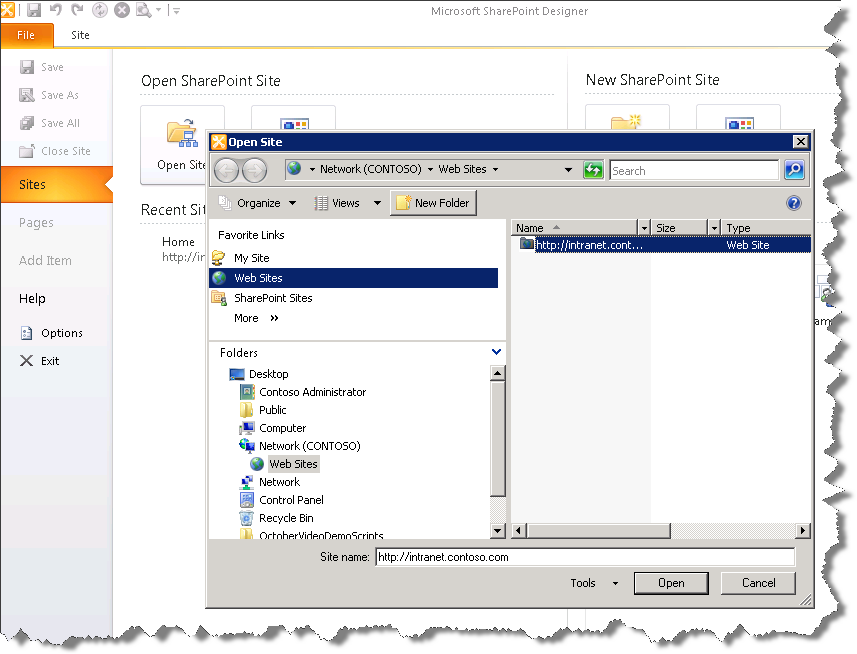
Enter **http://intranet.contoso.com** in theSitename textbox, and click **Open.** This will open the SharePoint Designer.  
  
  


Figure 31 - Open Site dialog

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

**Username**: administrator

**Password**: pass@word1

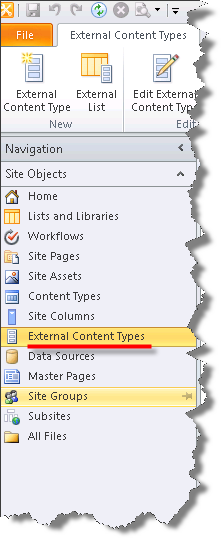
1. Click on **External Content Types** in the left hand navigation.  
     
   

Figure 32 - External Content Types menu item

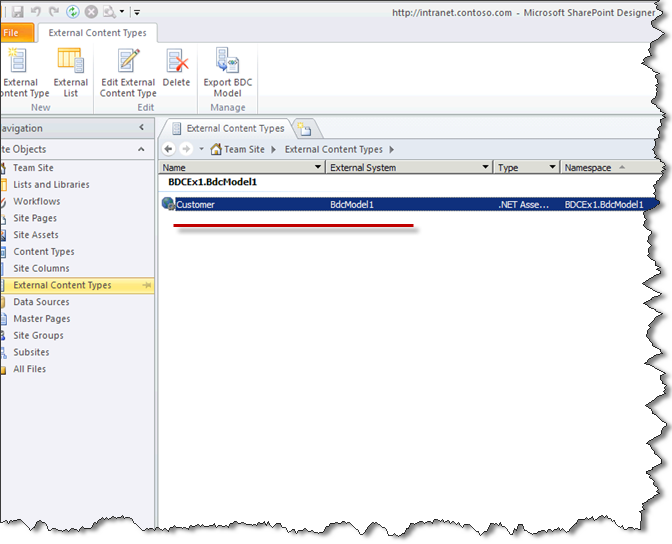
1. Double-click on the **Customer** entity.  
     
     
   

Figure 33 – External Content Types tab with available entities

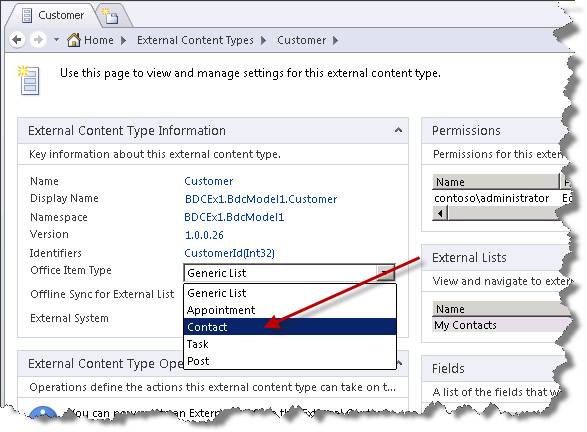
1. From the Office Item Type drop-down, select **Contact**.  
     
   

Figure 34 - External Content Type Information panel

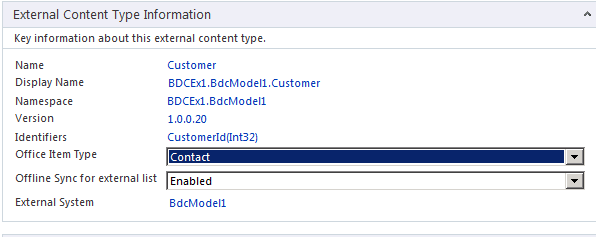
1. Note the warning in the **External Content Type Operations** pane.  
     
   

Figure 35 - External Content Type Operation panel

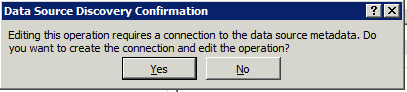
1. To edit the property mappings **double**-click **ReadItem**.
2. Click **Yes** in the **Data Source Discovery Confirmation** prompt.  
     
     
   

Figure 36 - Data Source Discovery Confirmation prompt

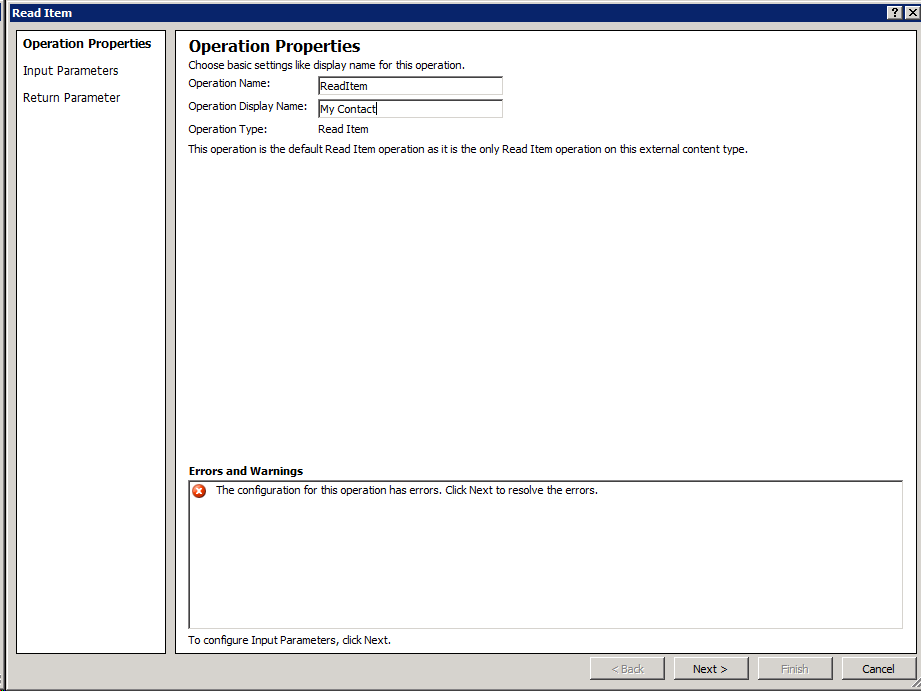
1. Enter **My Contact** in the Operation Display Name textbox. Ignore the configuration error for now.  
     
   

Figure 37 - Operational Properties dialog

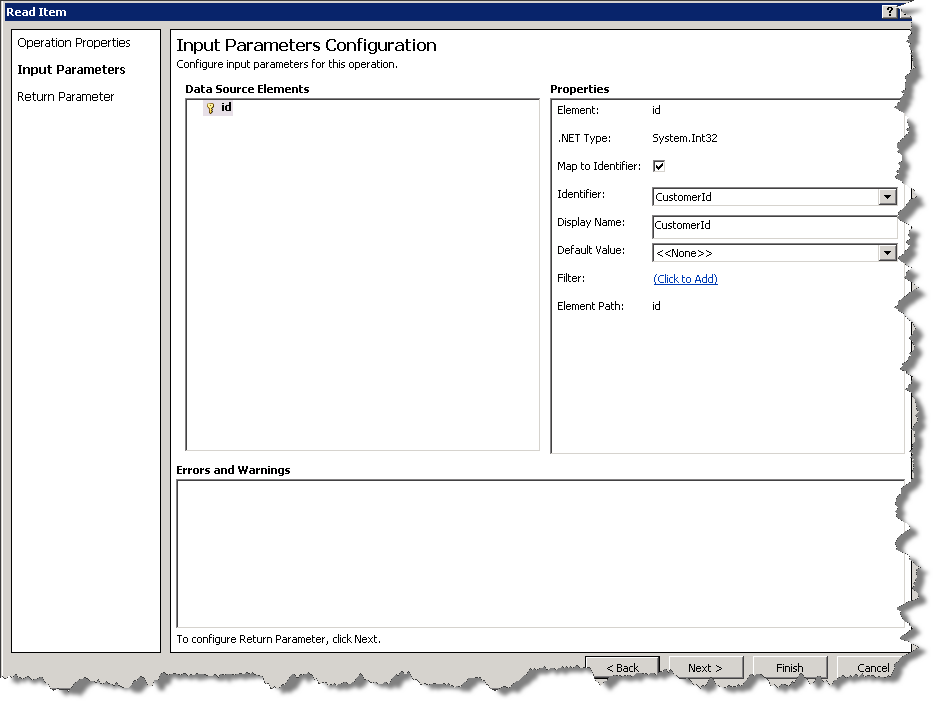
1. Click **Next**.  
     
   

Figure 38 - Input Parameters dialog

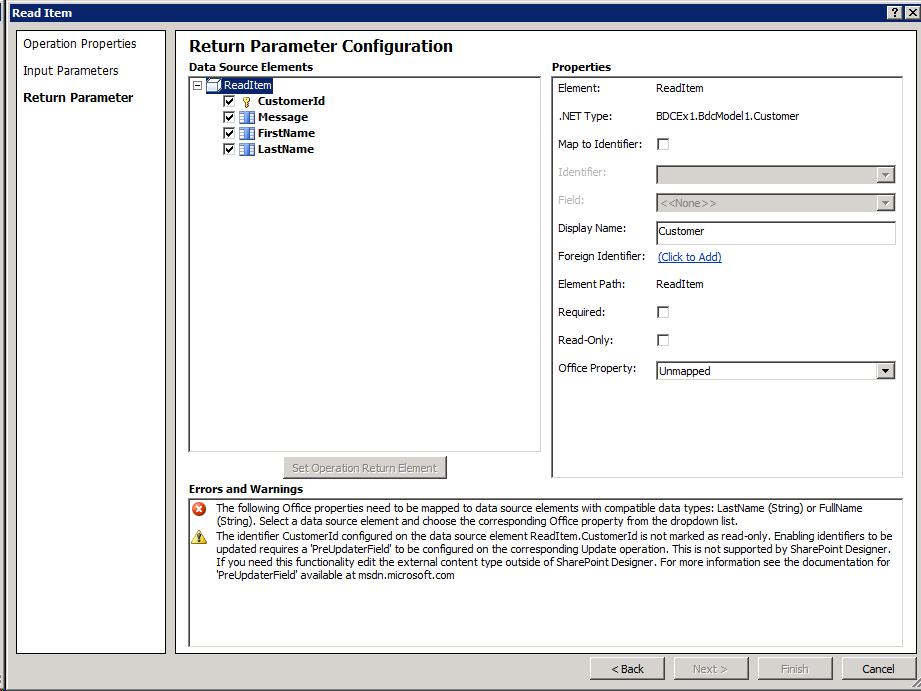
1. Click **Next** to open the Return Parameter Configuration screen.  
     
   

Figure 39 - The element configuration pane

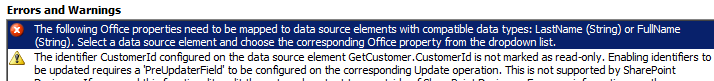
1. Note the **Error Messages** displayed at the bottom of the Return Parameter Configuration.  
     
     
   

Figure 40 - Error Message

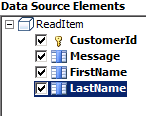
1. To map the **LastName (string),** select **LastName** in Parameter Data Type Hierarchy panel.  
     
   

Figure 41 - Parameter Data Type Hierarchy

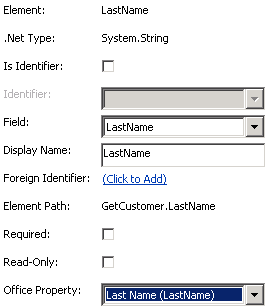
1. Select **Last Name (LastName)** from the **Office Property** drop-down. **Note** the error message clears as soon as Last Name is selected.  
     
   

Figure 42 - Properties for the LastName parameter

1. Select **FirstName** and set the **Office Property** to *First Name (FirstName)*.
2. Select **Message** and set the **Office Property** to *Body (Body)*.
3. Click **Finish**.
4. Click the **Save** icon at the top left of the screen.  
     
   

Figure 43 - Save button

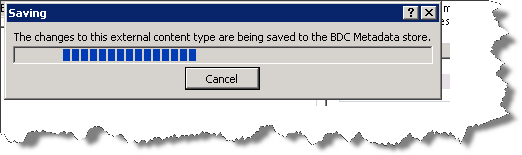
1. You will see a saving progress bar.  
     
   

Figure 44 - Saving progress dialog

### Task 2 – Taking the list offline

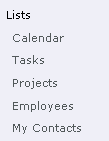
1. Open <http://intranet.contoso.com> in a browser session.
2. Click on **My Contacts** in the left hand navigation.  
     
   

Figure 45 - Lists section of the left hand navigation menu

1. Click **List**.  
     
   

Figure 46 - List Tools tab options

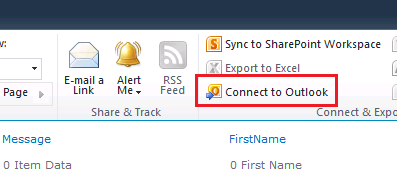
1. Click **Connect to Outlook**  
     
   

Figure 47 - Take list offline button

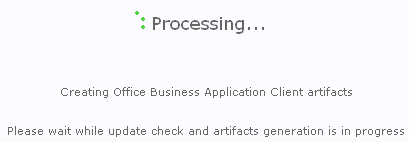
1. This will create a click-once installer to make the list available Offline in Outlook. If you get an Customization Installer prompt dialog as seen below, click **Install**. (This may take several minutes.)  
     
   

Figure 48 - Processing progress dialog

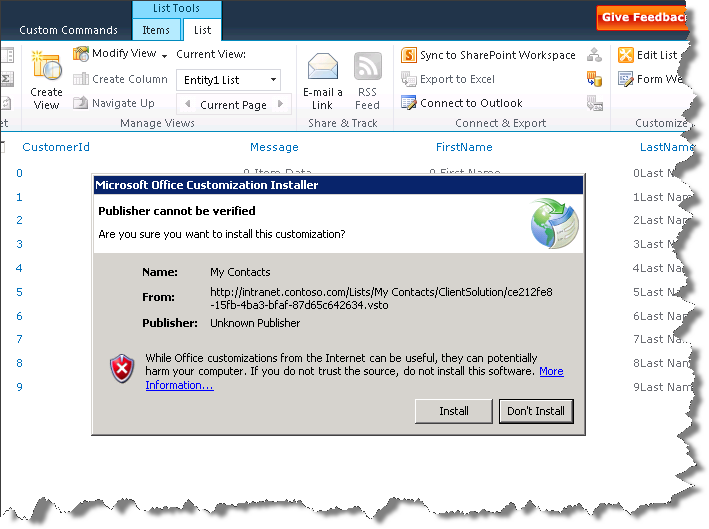


Figure 49 - Microsoft Office Customization Installer prompt

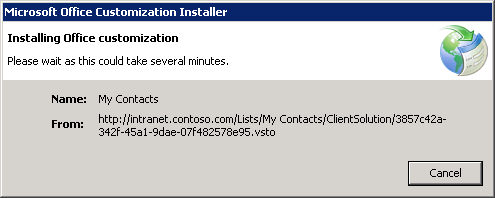
This may take several minutes.  
  


Figure 50 - Microsoft Office Customization Installer progress dialog

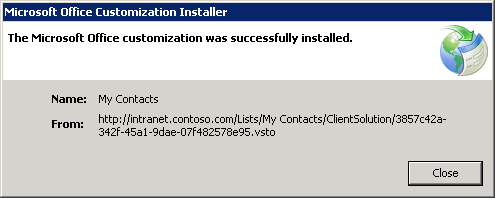
Once the installer has finished click **Close**.  
  


Figure 51 - Microsoft Office Customization Installer complted propmpt

1. Microsoft Outlook will open the external list and display the entities as Contacts

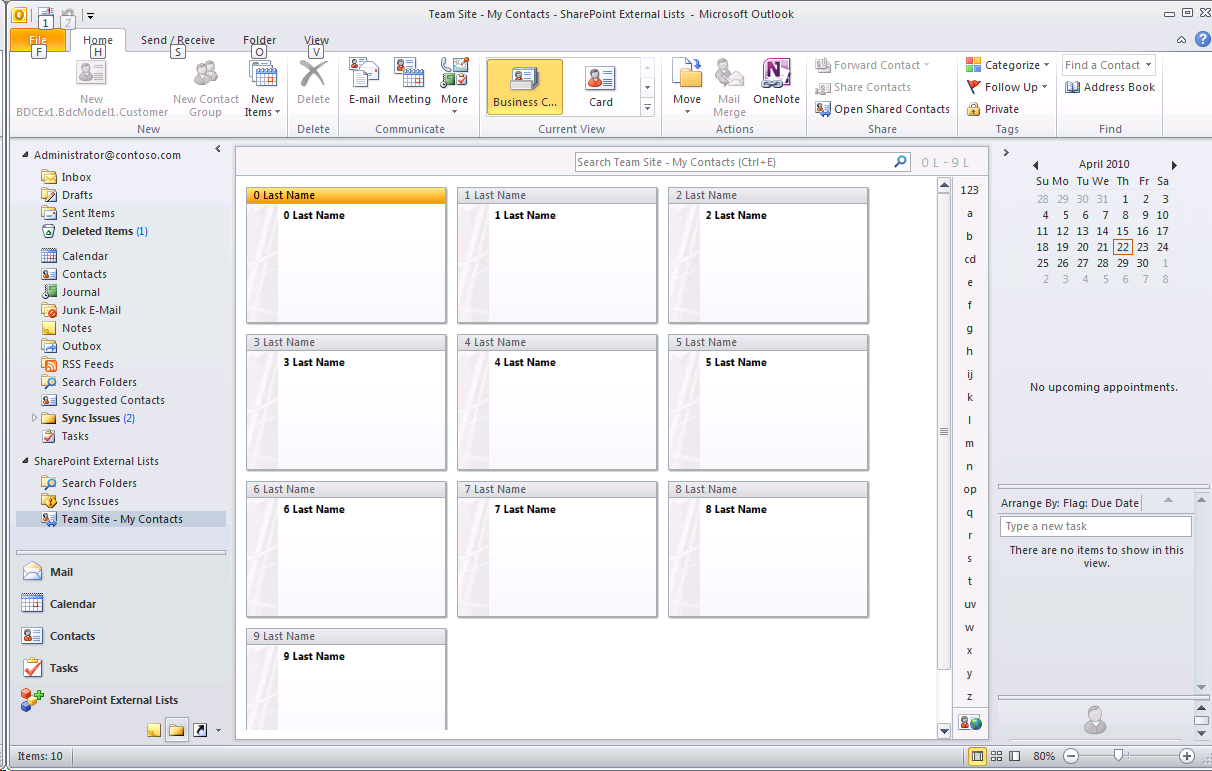


Figure 53 - Contacts displayed in Outlook

### Task 4 – Remove the external list

This is only necessary if you wish to work on other labs. The presence of the external list will break the SPMetal command used with any of the Link to SharePoint features in this Technical Preview build.

1. Open <http://intranet.contoso.com> in a browser session.
2. Click on **My Contacts** in the left hand navigation.
3. Click on **List** in the **List Tools** panel.
4. Click on **List Settings**.
5. Click on **Delete this list.**

## Lab Summary

In this lab you performed the following exercises.

1. Build a BCS External Content Type.
2. Create a Business Data Catalog Model project.
3. Extend the External Content Type.
4. Configure the External Content Type for offline use.
5. Taking the list offline.
6. Open the list using Outlook.