# Demo 1: OData Sources

In this Demo, you will create an External List from an oData source.

## Step 1 - Upload BDC Metadata Model

In this Step, you will upload a BDC Metadata Model to the BDC Service Application.

* You will need the file Northwind\_oData.bdcm

1. Open **Central Administration**.
2. Select **Application Management⮚Manage Service Applications**.
3. On the Service Application page, click **Business Data Connectivity Service**.
4. On the Ribbon in the BDC Service Application, click **Import**.
5. On the Import BDC Model page, click **Browse**.
6. In the **Choose a File to Upload** dialog, browse to the **Northwind\_oData.bdcm** file and click **Open**.
7. Click **Import**.
8. After the file is imported, click **OK**.

## Step 2 – Grant Permissions

In this Step, you will grant permissions for users to execute the methods of the BDC Metadata Model.

1. In the list of BDC Models, hover over **ODataWebNorthwindModel** and select **Set Permissions**.
2. In the **Set Object Permissions** dialog, click the **Browse** button.
3. In the **Select People and Groups** dialog, search for your current account.
4. Click **Add** and **OK**.
5. In the **Set Object Permissions** dialog, click **Add**
6. Check the permissions for **Edit**, **Execute, Selectable In Clients**, and **Set Permissions**.
7. Click **OK**.
8. In the Ribbon, select **External Content Types** from the drop-down list in the View group.
9. In the list of External Content Types, hover over **Customer** and select **Set Permissions**.
10. In the **Set Object Permissions** dialog, click the **Browse** button.
11. In the **Select People and Groups** dialog, search for your current account.
12. Click **Add** and **OK**.
13. In the **Set Object Permissions** dialog, click **Add**
14. Check the permissions for **Edit**, **Execute, Selectable In Clients**, and **Set Permissions**.
15. Ensure the **Propagate Permissions** box is checked.
16. Click **OK**.

## Step 3 – Create an External List

In this Step, you will create an External List based on the oData Source.

1. **Navigate** to a SharePoint site where you can create new lists.
2. On the home page of the site, click **Site Contents**.
3. On the Apps page, click **Add an App**.
4. On the Add an App page, click **External List**.
5. In the Adding an External List dialog, enter **Northwind Customers** in the **Name** field.
6. Click **Select External Content Type**.
7. In the External Content Type Picker, select the source **http://services.odata.org/Northwind/Northwind.svc**.
8. Click **OK**.
9. Click **Create**.
10. On the **Site Contents** page, click **Northwind Customers** to view the list.

# Demo 2: App-Level ECTs

In this Demo, you will create an External List in a SharePoint App from an ECT that is deployed with the App.

## Step 1 - Create a SharePoint App

In this Step, you will create a SharePoint App, which will contain the BDC Model and display an External List.

1. Open Microsoft Visual Studio 2012 and create a new SharePoint App project
   1. **Open** Microsoft Visual Studio 2012
   2. Select **File⮚New Project** from the main menu
   3. Click the **Templates⮚Visual C#⮚Office/SharePoint⮚Apps** node and select the **App for SharePoint 2013** project template
   4. Name the new project **AppLevelECT**
   5. Click the **OK** button
2. In the **SharePoint Customization Wizard**, name the new App **App-Level ECT**.
3. Specify the site you will use for this lab.
4. Select **SharePoint-hosted** as the hosting type.
5. Click **Finish**.
6. Generate an External Content Type
   1. In the **Solution Explorer**, right click the project node and select **Add⮚Content Types for an External Data Source** from the context menu.
   2. In the Specify OData Source dialog, enter **http://services.odata.org/Northwind/Northwind.svc** for the OData source Url.
   3. Name the OData source **Northwind OData Source**.
   4. Click **Next**.
   5. In the **Select Data Entities** screen, check **Alphabetical\_list\_of\_products**.
   6. Check the **Create List Instances** box.
   7. Click **Finish**.

## Step 2 - Code the App

In this Step, you will add code to the App to support the display of items in the External List.

1. Open **Default.aspx** for editing in Visual Studio.
2. Add the following HTML inside of the PlaceHolderMain control.

<div id="displayDiv"></div>

1. Open **Alphabetical\_list\_of\_products.ect** in Visual Studio.
2. Examine the names of the fields defined in the External List instance. These names will be used in code to retrieve values from the list.
3. Open **App.js** for editing in Visual Studio.
4. Add the following code to read the external List and display items in the home page of the App.

$(document).ready(function () {

//Namespace

window.AppLevelECT = window.AppLevelECT || {};

//Constructor

AppLevelECT.Grid = function (hostElement, surlWeb) {

this.hostElement = hostElement;

if (surlWeb.length > 0 &&

surlWeb.substring(surlWeb.length - 1, surlWeb.length) != "/")

surlWeb += "/";

this.surlWeb = surlWeb;

}

//Prototype

AppLevelECT.Grid.prototype = {

init: function () {

$.ajax({

url: this.surlWeb +

"\_api/lists/getbytitle('Alphabetical\_list\_of\_products')/items?" +

"$select=BdcIdentity,ProductID,ProductName",

headers: {

"accept": "application/json;odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

},

success: this.showItems

});

},

error: function (data) {

alert(data);

$("#displayDiv").html(data);

},

showItems: function (data) {

var items = [];

items.push("<table>");

items.push("<tr><td>Product ID</td>" +

"<td>Product Name</td></tr>");

$.each(data.d.results, function (key, val) {

items.push('<tr id="' + val.BdcIdentity + '"><td>' +

val.ProductID + '</td><td>' +

val.ProductName + '</td></tr>');

});

items.push("</table>");

$("#displayDiv").html(items.join(''));

}

}

ExecuteOrDelayUntilScriptLoaded(getProducts, "sp.js");

});

function getProducts() {

var grid = new AppLevelECT.Grid($("#displayDiv"),

\_spPageContextInfo.webServerRelativeUrl);

grid.init();

}

## Step 4 - Test the SharePoint App

In this Step, you will run the App and verify that the data from the External List is displayed.

1. Select **Debug⮚Start Debugging** in Visual Studio 2012
2. Verify that the App starts and displays items from the External List.

# Demo 3: External List Alerts

In this Demo, you will create an External List from a SQL Server database source and set up an alert.

## Step 1 - Install the Sample Database

In this Step, you will install the sample database for use with the External List.

* You will need the file MiniCRM.bak and administrator access to SQL Server.

1. Open **SQL Server Management Studio.**
2. In the **Connect to Server** dialog, enter the name of the server where SQL Server is running and click **Connect**.
3. In the **Object Explorer**, right click the **Databases** folder and select **Restore Database** from the context menu.
4. In the Restore Database dialog, enter MiniCRM in the Database field.
5. Select the Device option.
6. Click the ellipsis.
7. In the Specify Backup dialog, click Add.
8. In the Locate File dialog, navigate to MiniCRM.bak.
9. Click OK.
10. Click OK again.
11. Check the Restore box next to the backup set.
12. Click OK.

## Step 2 - Upload BDC Metadata Model

In this Step, you will upload a BDC Metadata Model to the BDC Service Application.

* You will need the file MiniCRMModel.bdcm

1. Open **Central Administration**.
2. Select **Application Management⮚Manage Service Applications**.
3. On the Service Application page, click **Business Data Connectivity Service**.
4. On the Ribbon in the BDC Service Application, click **Import**.
5. On the Import BDC Model page, click Browse.
6. In the Choose a File to Upload dialog, browse to the MiniCRMModel.bdcm file and click Open.
7. Click Import.
8. After the file is imported, click OK.

## Step 3 – Grant Permissions

In this Step, you will grant permissions for users to execute methods of the BDC Metadata Model.

1. In the list of BDC Models, hover over **MiniCRMModel** and select **Set Permissions**.
2. In the Set Object Permissions dialog, click the Browse button.
3. In the Select People and Groups dialog, search for your current account.
4. Click Add and OK.
5. In the Set Object Permissions dialog, click Add
6. Check the permissions for Edit, Execute, Selectable In Clients, and Set Permissions.
7. Click OK.
8. In the Ribbon, select **External Content Types** from the drop-down list in the View group.
9. In the list of External Content Types, hover over the Customer ECT associated with the MiniCRM External System and select Set Permissions.
10. In the Set Object Permissions dialog, click the Browse button.
11. In the Select People and Groups dialog, search for your current account.
12. Click Add and OK.
13. In the Set Object Permissions dialog, click Add
14. Check the permissions for Edit, Execute, Selectable In Clients, and Set Permissions.
15. Ensure the Propagate Permissions box is checked.
16. Click OK.
17. In the Ribbon, select **External Systems** from the drop-down list in the View group
18. Click **MiniCRM**.
19. Click **MiniCRM** again.
20. Edit the **Database Server Name** for the External System to refer to the SQL Server in your training environment.
21. Click **OK**.

## Step 3 – Create a Secure Store Application

In this Step, you will create a Secure Store Application that will provide credentials for accessing the MiniCRM database through an External List.

1. Open **Central Administration**.
2. Select **Application Management⮚Manage Service Applications**.
3. On the Service Application page, click **Secure Store Service**.
4. On the Ribbon in the Service Application, click **New**.
5. In the **Create New Target Application Page**, enter **SQLServer** in the **Target Application ID** field.
6. Enter **SQL Server** in the **Display Name** field.
7. Enter **administrator@contoso.com** in the **Contact E-Mail** field.
8. Select **Group** in the **Target Application Type** drop-down list.
9. Click **Next**.
10. Click **Next** again.
11. Enter your Windows account in the **Target Application Administrators** box and click **Check Names**.
12. Enter **All Authenticated Users** in the **Members** box and click **Check Names**.
13. Click **OK**.

|  |
| --- |
| Description: note_dd**Note** |
| *If you can’t get “All Authenticated Users” to resolve, you can use any other group that contains your account. For example, you may be able to use “Domain Users” as the group in your environment.* |

1. Hover over the **SQLServer** Application and select **Set Credentials**.
2. In the **Windows User Name** field, enter the service account that is running the SharePoint Web Application.
3. In the **Windows Password** field, enter the appropriate password.
4. In the **Confirm Windows Password** field, re-enter the password.
5. Click **OK**.

## Step 4 – Create an External List

In this Step, you will create an External List based on the MiniCRM database.

1. **Navigate** to a SharePoint site where you can create new lists.
2. On the home page of the site, click **Site Contents**.
3. On the Apps page, click **Add an App**.
4. On the Add an App page, click **External List**.
5. In the Adding an External List dialog, enter **Mini CRM Customers** in the **Name** field.
6. Click **Select External Content Type**.
7. In the External Content Type Picker, select the source **MiniCRM**.
8. Click **OK**.
9. Click **Create**.
10. On the Apps page, click **Mini CRM Customers** to view the list.
11. Click the **List** tab in the Ribbon.
12. Click the **List Settings** button.
13. On the List Settings page, click the **Customer Read List** view link.
14. In the **Sort** option, change the sort to be **ID** in **Descending Order**.
15. Click **OK**.

## Step 5 – Set up an Alert

In this Step, you will set up an alert for the External List

1. **Navigate** to the home page of the SharePoint site that is hosting the External List.
2. Select **Site**⮚**Site Settings**.
3. On the **Site Settings** page, click **Manage Site Features**.
4. **Activate** the **External System Events** feature.
5. Click the **Site Contents** link.
6. On the Site Contents page, click **Mini CRM Customers** to view the list.
7. Click the **List** tab in the Ribbon.
8. Click **Alert Me⮚Set Alert on this List**.

|  |
| --- |
| Description: note_dd**Note** |
| *When you set up an alert, a notification endpoint is created. You will use this endpoint in the next exercise. If you do not see the Alert Me button in the ribbon, go to Central Administration / System Settings / Configure outgoing e-mail settings and configure an SMTP server.* |

# Demo 4: External System Notifications

In this Demo, you will create a service that receives notifications from External Systems when data changes occur. These notifications will be recorded in a SharePoint list.

## Step 1 - Deploy the Notification Service

In this Step, you will deploy the notification service to SharePoint.

* You will need the Notifications solution, which is a complete Visual Studio project. The amount of code required to create the notification service and client is significant. Therefore, you will start with a completed project and deploy the solution to SharePoint.

1. Start **Visual Studio 2012**.
2. Open **Notifications.sln**.
3. In the **Solution Explorer**, click the **Notifications** project.
4. In the properties dialog, edit the **Site URL** to point to the site where you created the MiniCRM External List.
5. Right click the **Notifications** project and select **Deploy** from the context menu.

|  |
| --- |
| Description: note_dd**Note** |
| *Deploying the project will install a service for receiving notifications from External Systems and an Announcements list for recording the activity. The following shows the code for the Notify method, which is called by the External System to notify SharePoint of changes made outside of the External List.* |

public void Notify(string EventType, string Details)

{

SPList notificationList =

SPContext.Current.Web.Lists["Notifications"];

SPListItem notificationItem = notificationList.Items.Add();

notificationItem["Title"] = EventType;

notificationItem["Body"] = Details;

notificationItem.Update();

}

1. Open **Internet Explorer**.
2. Starting with the URL of the site hosting the MiniCRM External List, append the following to see the notifications web service: **/\_vti\_bin/notifications/NotificationEndpoint.svc**.
3. **Navigate** to the home page of the SharePoint site that is hosting the External List.
4. Click **Site Contents**.
5. On the **Apps** page, click on the **Notifications** list.
6. In the **Notifications** List, click on **SubscriptionId**.

|  |
| --- |
| Description: note_dd**Note** |
| *Deploying the project from Visual Studio caused a feature receiver to execute. This feature receiver called the Subscribe method of the Customer ECT associated with the MiniCRM database. The Subscribe method passes the Delivery Address and Event Type to the External System, which returns a Subscription ID. The following code shows how the Subscribe method of the ECT is called on feature activation.* |

public override void FeatureActivated(

SPFeatureReceiverProperties properties)

{

SPWeb site = properties.Feature.Parent as SPWeb;

using (SPSite siteCollection = new SPSite(site.Site.ID))

{

//Get the ECT

BdcService service = SPFarm.Local.Services.GetValue<BdcService>();

SPServiceContextScope scope =

new SPServiceContextScope(

SPServiceContext.GetContext(siteCollection));

DatabaseBackedMetadataCatalog catalog =

service.GetDatabaseBackedMetadataCatalog(

SPServiceContext.GetContext(siteCollection));

IEntity ect = catalog.GetEntity(

"http://contoso.com/minicrm", "Customer");

ILobSystemInstance lobi =

ect.GetLobSystem().GetLobSystemInstances()["MiniCRM"];

//This will call the subscribe method

IMethodInstance mi = ect.GetMethodInstance(

"Subscribe",

MethodInstanceType.EventSubscriber);

IParameterCollection parameters = mi.GetMethod().GetParameters();

string endpointUrl = site.Url +

"/\_vti\_bin/Notifications/NotificationEndpoint.svc";

object[] arguments = new object[parameters.Count];

arguments[0] = endpointUrl;

arguments[1] = EntityEventType.ItemAdded;

ect.Execute(mi, lobi, ref arguments);

//This saves the subscription ID

PropertyInfo[] props = arguments[2].GetType().GetProperties();

PropertyInfo prop = props[0];

SqlDataReader reader = (SqlDataReader)(prop.GetValue(

arguments[2], null));

if (reader.HasRows)

{

while (reader.Read())

{

SPList notificationList = site.Lists["Notifications"];

SPListItem notificationItem =

notificationList.Items.Add();

notificationItem["Title"] = "SubscriptionId";

notificationItem["Body"] = reader.GetInt32(0).ToString();

notificationItem.Update();

}

}

}

}

1. Open **SQL Server Management Studio.**
2. In the **Connect to Server** dialog, enter the name of the server where SQL Server is running and click **Connect**.
3. In the **Object Explorer**, expand the **Databases** folder.
4. Expand the **MiniCRM** folder.
5. Expand the **Tables** folder.
6. Right click the **Subscriptions** table and select **Select Top 1000 Rows** from the context menu.

|  |
| --- |
| Description: note_dd**Note** |
| *The Subscriptions table contains the Delivery Address and Event Type for each subscription. The ItemAdded event is from the feature receiver in the project you deployed. The entries “1”, “2”, and “3” are from the alerts you set up on the list previously.* |

1. **Navigate** to the home page of the SharePoint site that is hosting the External List.
2. Click **Site⮚Site Settings**.
3. On the Site Settings page, click **Manage Site Features**.
4. **Deactivate** the **Notifications List** feature.
5. Return to **SQL Server Management Studio**.
6. Refresh the view of the Subscriptions table and verify that the **ItemAdded** subscription is deleted.
7. Return to SharePoint and Activate the **Notifications List** feature.
8. **Navigate** to the External List.
9. In the ribbon, click **List**.
10. In the **Share & Track** group, select **Alert Me⮚Manage My Alerts**.
11. **Delete** the three alerts you set up earlier.
12. Return to **SQL Server Management Studio**.
13. Refresh the view of the Subscriptions table and verify that the alert endpoints are deleted.
14. **Navigate** to the External List and set up the alerts again.

## Step 2 - Call the Notification Service

In this Step, you will use a Windows application to make changes to the MiniCRM database. This application will also call the notification service endpoint and the alert endpoints when changes are made. In the Demo, these notifications are logged to the Announcements list and e-mails are sent out.

1. Return to **Visual Studio**.

|  |
| --- |
| Description: note_dd**Note** |
| *The ExternalApplication project provided is configured to work against a site URL of http://contososerver/bcs. If this is not the Url for your site, then you will have to perform a search and replace for all references to the Url contained in the project to update them for your environment.* |

1. In the Solution Explorer, right click the **ExternalApplication** project and select **Debug⮚Start New Instance** from the context menu.
2. In the **Add a Customer** form, enter **Mr** in the **Title** field.
3. Enter **Brian** in the **First Name** field.
4. Enter **Cox** in the **Last Name** field.
5. Enter **brian@contoso.com** in the **E-Mail Address** field.
6. Enter **555-555-5555** in the **Work Phone** field.
7. Click **Save**.

|  |
| --- |
| Description: note_dd**Note** |
| *The client application uses standard ADO.NET code to add a record to the database. Then it queries the Subscriptions table for all subscriptions. For each delivery address in the table, the client calls the notification endpoint.* |

1. **Navigate** to the **Notifications** List and view the Item Added notification.

|  |
| --- |
| Description: note_dd**Note** |
| *In this Demo, the custom notification is simply logged to a list. In a real-world implementation, the custom notification can be used to take almost any action. For example, it could be used to launch a workflow.* |

1. Open **Microsoft Outlook** and view the alert e-mail.
2. **Navigate** to the **Mini CRM Customers** External List and view the added record.