## Working with External Data Using BCS

**Lab Time**: 60 minutes

**Lab Folder**: [[StudentFolder]]\BCS

**Lab Overview**: In this module you will learn how to read and write to external data sources using the Business Connectivity Services (BCS) capabilities included in SharePoint 2013. You will learn how to create external content types using SharePoint Designer 2013, Visual Studio 2012 and how to include them within SharePoint apps as well as how to create special callouts.

### Exercise 1: Setup Lab Environment

In this exercise you will setup your environment.

All exercises in this lab assume you will work in a new site collection, http://bcs.wingtip.com.

1. Setup a new site collection for this lab:
   1. Ensure you are logged into the **WingtipServer** server as **WINGTIP\Administrator**.
   2. Run a PowerShell script, found in the root lab folder for this module:
      1. Right-click **SetupModule.ps1** and select **Run with PowerShell**. This file can be found in the files associated with this lab:

[..]\BCS

* 1. When the script completes, it will launch a new browser and navigate to the lab site collection.
  2. Close the PowerShell console window.

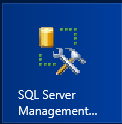
#### Setup Sample Resource Web Services

1. Ensure you are logged into the **WingtipServer** server as **WINGTIP\Administrator**.
2. Check to see if the sample services have already been deployed:
   1. Open Internet Explorer and try to navigate to the following OData services:
      1. <http://cptresources.wingtip.com:81/services/AdventureWorks2012Person.svc>
      2. <http://cptresources.wingtip.com:81/services/AdventureWorks2012Product.svc>
      3. <http://cptresources.wingtip.com:81/services/Calculator.svc>
   2. If the links above work and do not return 404 errors, you can skip to the next exercise **(Exercise 2)** because the sample site & services have already been deployed.
3. If the **Calculator.svc** service works but the two **AdventureWorks2012\*.svc** services **do not work**, this indicates the services were deployed, but two of the services are not able to connect to the local SQL Server sample **AdventureWorks2012** database. The possible issues include:
   1. A connection problem with the IIS website.
   2. The connection strings in the **cptservices** application’s **web.config**.
   3. The sample **AdventureWorks2012** database is not installed in the local SQL Server.
   4. The identity of the hosting app pool does not have rights to the **AdventureWorks 2012** database.
4. If all three of the web services return 404 errors continue on to step 6 to create these.

You must resolve any issues and ensure these web services are working before proceeding to Exercise 2.

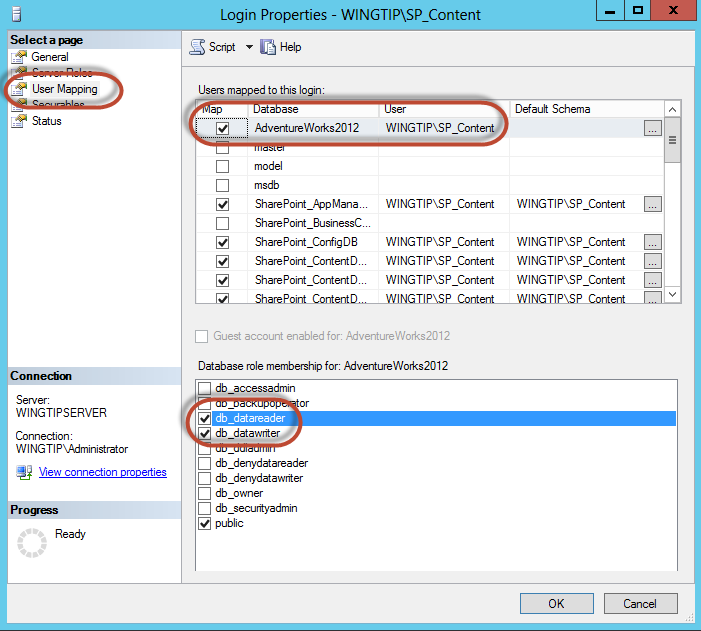
#### Update the App Pool Identity Rights in the SQL Server Adventure Works Database

1. Grant the identity of the hosting web site’s application pool access to the AdventureWorks2012 sample database:
   1. Open **SQL Server Management Studio**:
      1. **Windows Keyboard Key 🡪 SQL Server Management Studio.**



**(Note:** Alternatively after pressing the **Windows Keyboard Key** you can simply start typing the name of the program you are looking for (e.g. SQL); this will filter the results to those that match the letters typed on the keyboard.

* 1. In the **Connect to Server** dialog, enter the following and click **Connect**:
     1. **Server Type:** Database Engine
     2. **Server name:** WINGTIPSERVER
     3. **Authentication:** Windows Authentication
  2. In the **Object Explorer** tool window, expand the tree to show the contents of **WINGTIPSERVER 🡪 Security 🡪 Logins**.
  3. Right-click **WINGTIP\SP\_Content** and select **Properties**.
  4. In the **Select a page** pane on the left-hand side of the **Login Properties – WINGTIP\SP\_Content** dialog, select **User Mapping**.
  5. In the top portion of the dialog, in the **Users mapped to this login**, check the box next to **AdventureWorks2012**.
  6. In the bottom portion of this dialog, in the **Database role membership for AdventureWorks2012**, check the boxes next to:
     1. **db\_datareader**
     2. **db\_datawriter**
  7. Click **OK**



1. Next we need to add the **NT AUTHORITY\IUSR** account the SQL Database to account for an issue with the Single Server image environment.  
   (Note: you would not to do this on a production multi-server farm)
   1. In the **SQL Server Management Studio** **Object Explorer** tool window, expand the tree to show the contents of **WINGTIPSERVER 🡪 Security 🡪 Logins**.
   2. Right-click **Logins** and select **New Logins**.
   3. In the **Login - New** dialog box:
      1. **Login name:** NT AUTHORITY\IUSR
      2. On the left-hand side of the Login– New dialog, select **User Mapping**.
   4. In the top portion of the dialog, in the **Users mapped to this login**, check the box next to **AdventureWorks2012**.
   5. In the bottom portion of this dialog, in the **Database role membership for AdventureWorks2012**, check the boxes next to:
      1. **db\_datareader**
      2. **db\_datawriter**
   6. Click **OK** and **close** SQL Server Management Studio.

#### Create Sample Resource Website & Deploy Sample Services Project

1. Create a new IIS web site to host the sample CPT OData web services:
   1. Run a PowerShell script, found in the **ExtraStudentFiles** folder in the student folder:
      1. Right-click **CreateCptResourcesWebSite.ps1** and select **Run with PowerShell**. This file can be found in the files associated with this course:

[..]\ExtraStudentFiles\Scripts

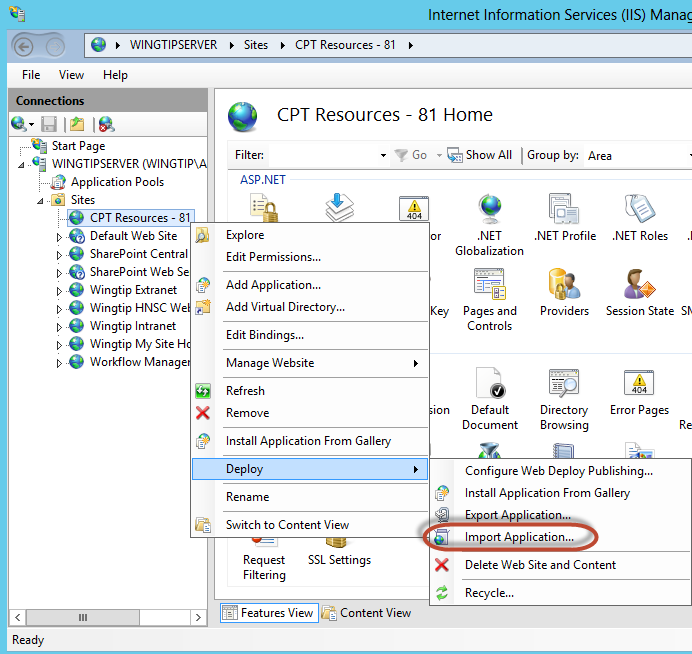
This script will create a new website in IIS named **CPT Resources – 81** with a binding of **http://cptresources.wingtip.com:81**.

* 1. Close the PowerShell console window.

1. Install the CPT Services sample OData Web services project:
   1. Open **Internet Information Services (IIS) Manager**:
      1. **Windows Keyboard Key 🡪 Internet Information Services (IIS) Manager.**



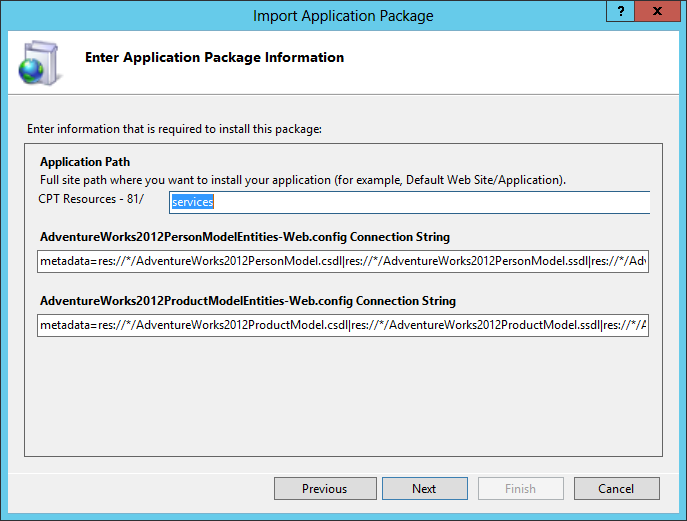
* 1. In the left-hand **Connections** pane, expand the tree to the following location:
     1. **WINGTIPSERVER 🡪 Sites 🡪 CPT Resources – 81**
  2. Right-click **CPT Resources – 81** and select **Deploy 🡪 Import Application**.



* 1. In the **Import Application Package** dialog’s **Select the Package** page, browse to the file **CptServices\_WebDeployPackage.zip** found in the files associated with this course:

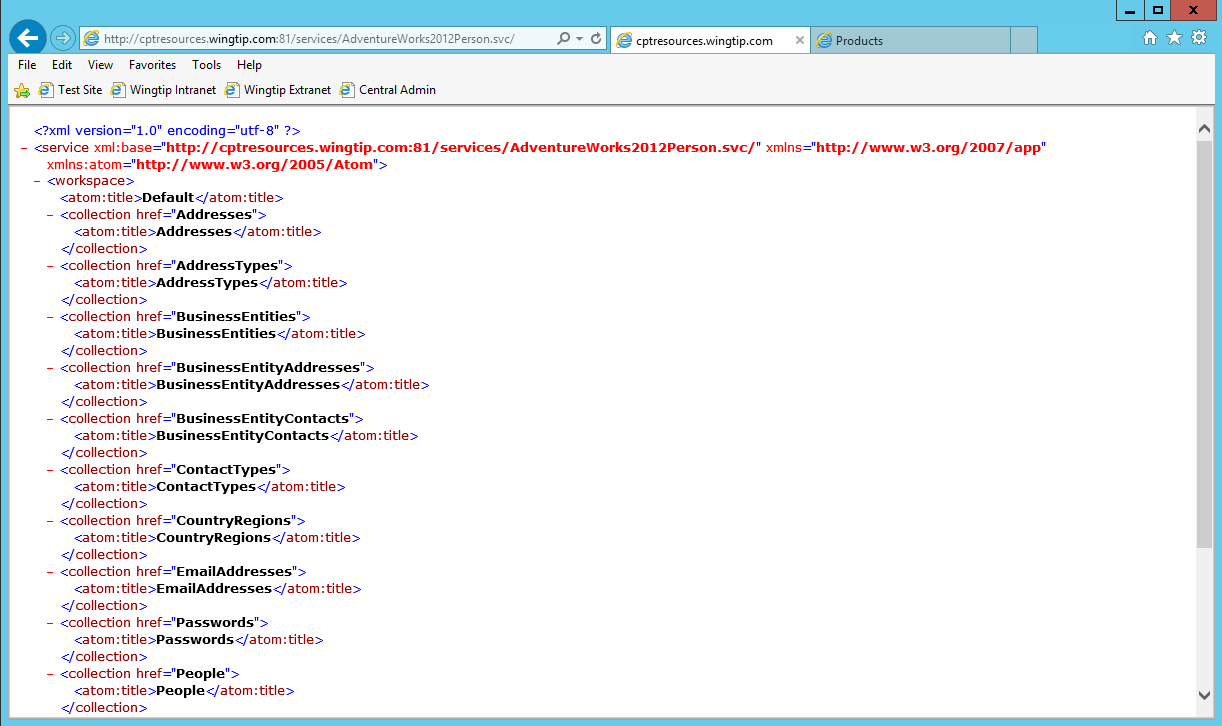
[..]\ExtraStudentFiles\Resources

* 1. Click **Open** and then click **Next**.
  2. On the **Select the Contents of the Package** page, click **Next**.
  3. On the **Enter Application Package Information** page, make sure the **Application Path** is set to **services** and click **Next.**

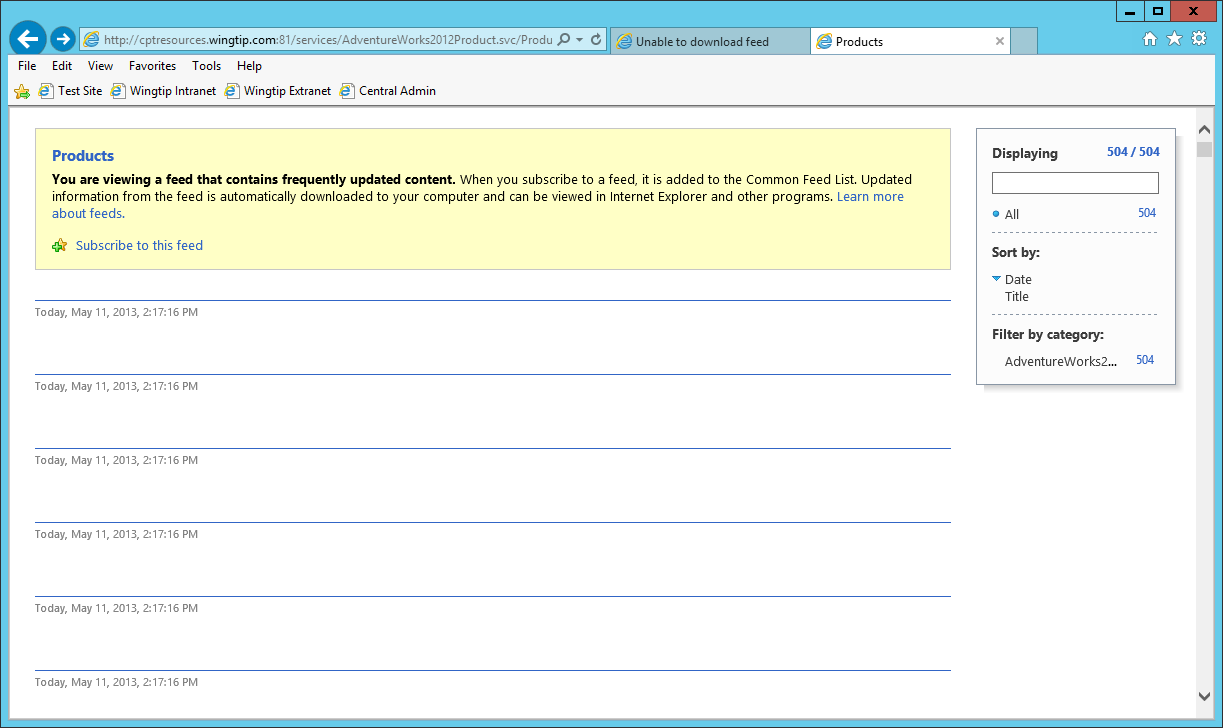
**.**

* 1. After the package is deployed, click **Finish** on the **Installation Progress and Summary** page to close the dialog.

1. Verify the sample OData services are working:
   1. Open Internet Explorer and try to navigate to the following OData services:
      1. <http://cptresources.wingtip.com:81/services/AdventureWorks2012Person.svc>
      2. <http://cptresources.wingtip.com:81/services/AdventureWorks2012Product.svc>
      3. <http://cptresources.wingtip.com:81/services/Calculator.svc>
   2. You should see XML returned by all services as the following figure shows:



* 1. Open Internet Explorer and try to navigate to the following OData services:
     1. <http://cptresources.wingtip.com:81/services/AdventureWorks2012Person.svc>/People
     2. <http://cptresources.wingtip.com:81/services/AdventureWorks2012Product.svc/Products>
  2. With the **Person.svc/People** you should see a message that **Internet Explorer cannot display this feed (The size of this feed exceeds the download limit)**
  3. With the **Products.svc/Products** you should see feed information on Products similar to that shown in the image below:

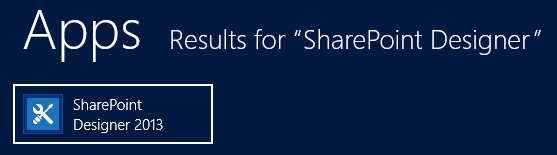


You are now ready to perform the following lab exercises.

### Exercise 2: Create & Use an External Content Type with SharePoint Designer 2013

In this exercise you will create an external content type using SharePoint Designer 2013 and use it in an external list.

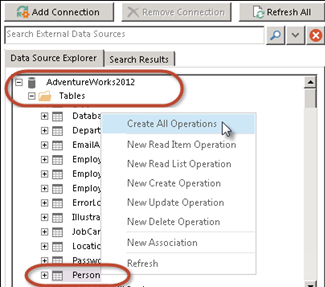
1. Ensure you are logged into the **WingtipServer** server as **WINGTIP\administrator**.
   * 1. Open **SharePoint Designer 2013**: **Windows Keyboard Key 🡪 Type “SharePoint Designer”** and then select the SharePoint Designer 2013 application.



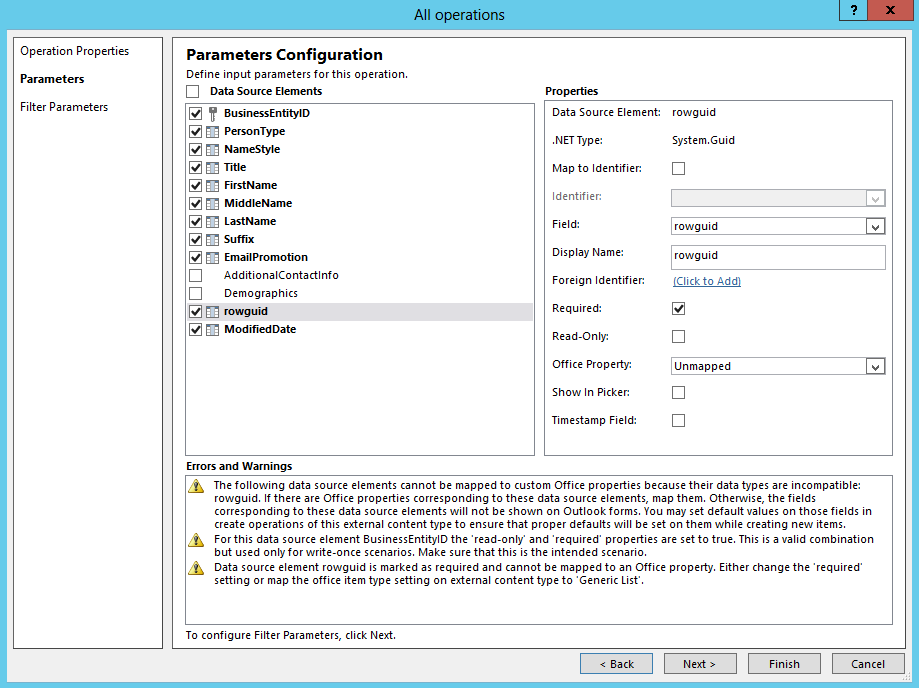
1. Open the lab site collection:
   1. Click the **Open Site** button.
   2. In the **Open Site** dialog, enter <http://bcs.wingtip.com> in the **Site Name** box and click **Open**.
   3. If prompted to login, use the credentials for **WINGTIP\ administrator**.
2. Select **External Content Types** in the **Navigation** pane.
3. Click the **External Content Type** button in the **New** group on the ribbon.
4. Use the following values to create the new external content type:
   1. **Name:** AdventureWorksPerson
   2. **Display Name:** Adventure Works Person
   3. **Office Item Type:** Contact
5. Using the ribbon, click the **Operations Design View** button in the **Views** group.
6. Click the **Add Connection** button.
   1. In the **External Data Source Type Selection** dialog, select **SQL Server** and click the **OK** button.
   2. In the **SQL Server Connection** dialog, use the following and click **OK**:
      1. **SQL Server:** WINGTIPSERVER
      2. **Database Name:** AdventureWorks2012
      3. **Connect with User’s Identity:** checked
7. Expand the **AdventureWorks2012** node and the **Tables** folder and find the **Person** table.

(Note: if a warning dialog box pops up, click **OK**)

1. Right-click the **Person** table, and select **Create All Operations** from the context menu:



1. Use the All Operations wizard dialog to create the operations needed for this external content type:
   1. On the **Operations Properties** page, click **Next**.
   2. On the **Parameters Configuration** page, do the following and click **Next**:
      1. Uncheck the following fields:
         1. **AdditionalContactInfo**
         2. **Demographics**
      2. Select the **FirstName** column and set the following values in the **Properties** section to the right:
      3. **Required**: checked
      4. **Office Property**: First Name (FirstName)
      5. **Show In Picker**: checked
         1. Repeat the previous step for the **MiddleName** and **LastName** column picking the corresponding **Office Property**.



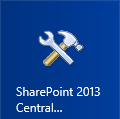
1. On the **Filter Parameters Configuration** page, do the following to create a new limit filter
   1. Click the **Add Filter Parameter** button.
   2. In the **Properties** are of the **All operations** dialog box, click the **Click to Add** link in the **Filter:** area.
      1. In the **Filter Configuration** dialog, set **Filter Type** to **Limit** and click **OK**:
   3. Set the **Default Value** to **200**.
   4. Click **Finish**.
2. Save your changes by clicking the **File** tab in the ribbon and then **Save**.

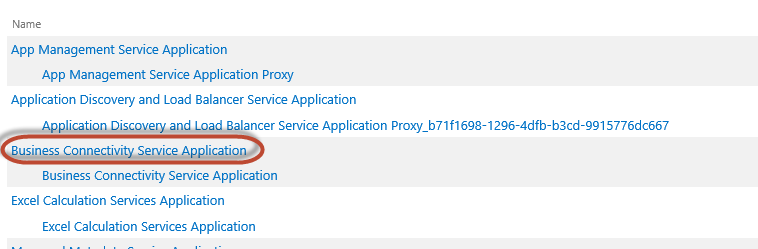
#### Set the Permissions on the External Content Type

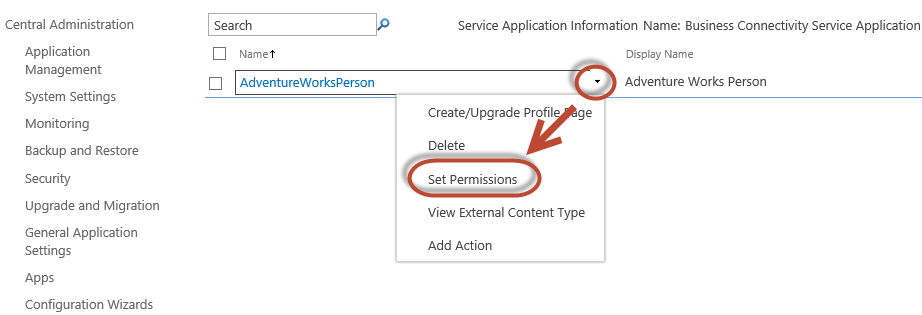
1. Before using the external content type, you must grant users the ability to use it. This is done by editing the external content type in the Business Data Connectivity service application:

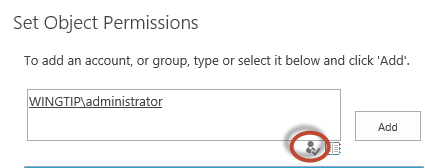
This is different than having rights to access the underlying data. Users must also be granted access to use the external content type that exposes the data to SharePoint and the users.

* 1. Open **Central Administration**:
     1. **Windows Keyboard Key 🡪 SharePoint 2013 Central Administration.**



* 1. In the **Application Management** section of the **default.aspx** screen click on **Manage Service Applications**
  2. On the **Manage Service Applications** screen, select the **Business Data Connectivity Service** service application.  
     (i.e. click on the **Business Data Connectivity Service Application** hyperlink as shown below  
     
  3. Select the **AdventureWorksPerson** external content type **drop down** and in the menu that appears, select **Set Permissions**.



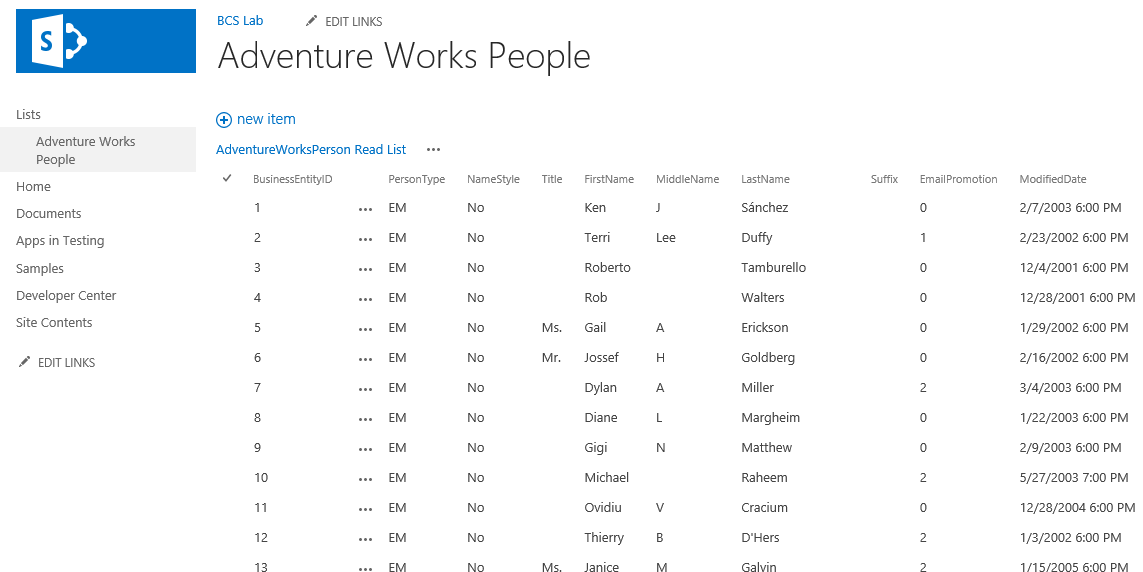
* 1. In the **Set Object Permissions** dialog, do the following:
     1. In the top-most box, type **WINGTIP\Administrator** and click the **Check Names** icon (image with check mark over person).
     2. Click the Check User Button  
        
     3. After the user is validated, click **Add**.
     4. Ensure the user is selected in the middle box.
     5. Check all four permissions for the user and click **OK**.

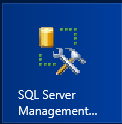
Alternatively you could have granted groups of users (from Active Directory) different rights. Typically you would grant everyone who was allowed to use the object **Execute** rights, which means they can execute operators defined in the external content type.

1. Now the external content type can be used by the WINGTIP\administrator within a site.

#### Create an External List Based on the External Content Type

1. Using Internet Explorer, navigate to <http://bcs.wingtip.com>.
2. Create a new external list based on the Adventure Works Person external content type:
   1. Using the **Quick Launch** navigation, select **Site Contents**.
   2. On the **Site Contents** page, click **add an app**.
   3. Select the app **External List**.
   4. In the **Adding External List** dialog, use the following values to create the list::
      1. **Name:** Adventure Works People
      2. **External Content Type:**
         1. Click the icon that has a database picture superimposed on a document (i.e. the **Select External Content Type** icon).
         2. In the **External Content Type Picker** dialog, select **AdventureWorks2012** and click **OK**.
      3. Click **Create**
3. When the Site Contents page loads, select the **Adventure Works People** app. The list should load showing data from the database table:



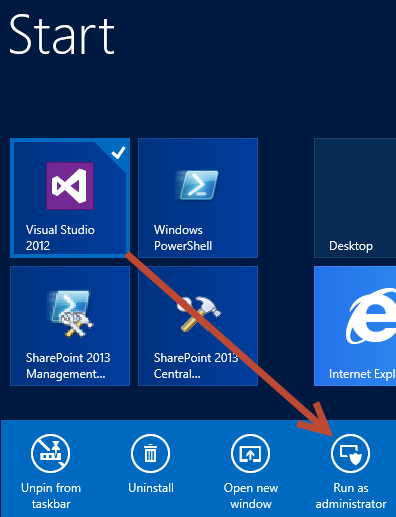
1. Edit an item and see how it changes in the database:
   1. Select the first item, **Ken Sanchez**, in the list.
   2. Using the ribbon, under the **Items** tab, click the **Edit Item** button in the **Manage** group.
   3. Change Ken’s middle name from **J** to **Joseph**.
   4. Click **Save**.
2. Notice how Ken’s middle name in the list changed.
3. Verify Ken Sanchez’s record in the database was updated
   * 1. **Windows Keyboard Key 🡪 SQL Server Management Studio**[**Note:** Alternatively after pressing the **Windows Keyboard Key** you can simply start typing the name of the program you are looking for (e.g. SQL); this will filter the results to those that match the letters typed on the keyboard]
   1. In the **Connect to Server** dialog, enter the following and click **Connect**:
      1. **Server Type:** Database Engine
      2. **Server name:** WINGTIPSERVER
      3. **Authentication:** Windows Authentication
4. After logging in, using the **Object Explorer** tool window in the left-hand side of **SQL Server Management Studio**, expand the tree to see the contents of **Tables**: **WINGTIPSERVER 🡪 Databases 🡪 AdventureWorks2012 🡪 Tables**.
5. Right-click the **Person.Person** table and select **Select Top 1000 Rows**.
6. The first record should be Ken Sanchez with his new middle name.
7. **Close** SQL Server Management Studio and SharePoint Designer.

In this exercise you created a new external content type using SharePoint Designer 2013 and then used it in an External List.

### Exercise 3: Create & Use an External Content Type in an App with Visual Studio 2012

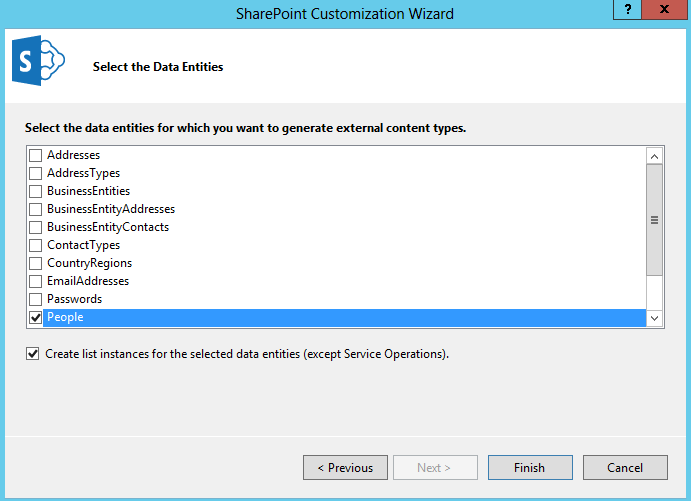
In this exercise you will create an external content type with Visual Studio and use it within a SharePoint app.

1. Create a new project in Visual Studio 2012:
   1. Launch **Visual Studio 2012** as administrator:
      1. **Windows Keyboard Key** 🡪 **Right click** on the **Visual Studio 2012** tile and select **Run as administrator.**



* 1. In Visual Studio select **File 🡪 New 🡪 Project**.
  2. In the **New Project** dialog:
     1. Find the **App for SharePoint 2013** template under the **Templates** 🡪 **Visual C#** 🡪 **Office / SharePoint** 🡪 **Apps** section.
     2. Name: **AppWithEct**
     3. Location**:** **[..]\BCS\Exercises\Ex3**  
        (Where [..] represents the location of the student files (e.g. c:\student\..)
     4. **Uncheck** the **Create directory for solution** checkbox
     5. Click **OK**
  3. In the **New App for SharePoint** wizard, use the following values to complete the wizard and click **Finish**.
     1. **What is the name of your App for SharePoint?** App With ECT
     2. **What site do you want to use for debugging?** <http://bcs.wingtip.com>
     3. **How do you want to host your app for SharePoint?** SharePoint-hosted

1. After the project is created, create an External Content Type:
   1. Right-click the project **AppWithEct** in the **Solution Explorer** tool window and select **Add 🡪 Content Types for an External Data Source** from the context menu.
   2. In the **Specify OData Source** wizard, on the **Specify OData Source** page, use the following values and click **Next**:
      1. **What OData Service URL do you want to use to create the external data source?** <http://cptresources.wingtip.com:81/services/AdventureWorks2012Person.svc>
      2. **Data Source Name:** Adventure Works OData People Service
   3. In the **Specify OData Source** wizard, on the **Select the Data Entities** page, use the check **People** and click **Finish**:

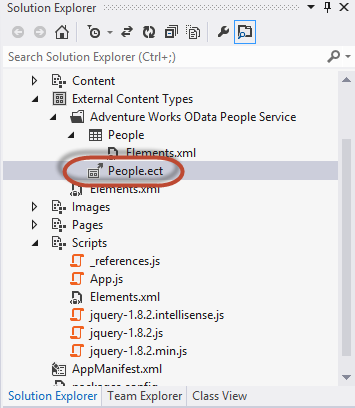


#### Code the SharePoint App

1. Update the homepage of the app:
   1. Right-click the **Pages \ Default.aspx** file in the **Solution Explorer** tool window and select **View Markup**.
   2. Replace the contents of the ASP.NET content placeholder **PlaceHolderMain** control with the following code:

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

1. Examine the names of the fields that will be added to the External List based on the External Content Type:
   1. Right-click the **People.ect** file in the Solution Explorer tool window and select **Open**:



* 1. The values in the **Column Name** column are what will appear in the list when the app is deployed.

1. Update the client-side code for the app:
   1. Right-click the **Scripts \ App.js** file in the **Solution Explorer** tool window and select **Open**.
   2. Next, delete the following variables and functions that aren’t used:

var context = SP.ClientContext.get\_current();

var user = context.get\_web().get\_currentUser();

getUserName() {..}

onGetUserNameSuccess() {..}

onGetUserNameFail() {..}

* 1. Next, add the following code to the top of the **App.js** file just underneath **‘use strict;** :

//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('People')/items?" +

"$select=BdcIdentity,FirstName,LastName",

headers: {

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

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

},

context: this,

success: this.showItems

});

},

showItems: function (data) {

var items = [];

items.push("<table>");

items.push("<tr>" +

"<td>First Name</td><td>Last Name</td></tr>");

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

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

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

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

});

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

this.hostElement.html(items.join(''));

}

};

function getProducts() {

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

\_spPageContextInfo.webServerRelativeUrl);

grid.init();

}

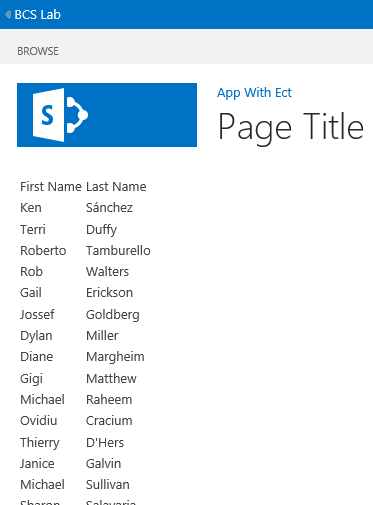
* 1. Lastly, replace the contents of the **$(document).ready()** function with the following:

getProducts();

1. Save all changes: **File 🡪 Save All**.

#### Build and Test the Project

1. Build and test your application by pressing **[F5]** or **Debug 🡪 Start Debugging**.
2. Once the solution has been deployed, Internet Explorer will launch and navigate to the <http://bcs.wingtip.com> site.
3. When prompted by SharePoint to grant permissions, click **Trust It**.
4. On the **Site Contents > Your Apps**, click the **App With ECT** app in the **Lists, Libraries and other Apps** section.
5. You should see the app load and display items from the OData Feed:



In this exercise you created an external content type that resided within a SharePoint-Hosted App and displayed data from a remote OData feed in the browser.