JOHN ABBOTT
CEGEP/COLLEGE

(420-PS4-AB)
Data Sources : Part 2

Summer 2017

Outline

- ADO .NET
- Intro to LINQ
- ObjectDataSource Control

ADO .NET

What is ADO .NET

- ADO .NET stands for ActiveX Data Objects.
- Essential technology that provides access to data sources
  - SQL Server, XML & Other(OLEDB ODBC).
  - Can be used in different .NET applications.
- Connection to data source:
  - Connection
  - Command
  - DataReader
- Containers of data
  - DataSet
  - DataTable
  - DataRow

ADO .NET Namespaces

- System.Data.SqlClient;
  - SQL Connection
  - SQL Commands
- System.Data;
  - DataSet
  - Data Relations

Direct Access in ADO .NET

- Steps to access a data source:
  - Create a ConnectionString
  - Create an SqlConnection
  - Create a command (sqlCommand)
  - Create SqlDataReader
  - Use Reader
  - Close connection

Demo: Supplier Search Page

- Using Northwind database and ADO .NET create a web form to search for suppliers.

- Test Connection Task:
  - Create connection to database.
  - Test the connection and display result in a label.
- Task 1:
  - The form will have a text box labeled “Supplier ID” and a button search.
  - Up on the click on the button, we will query the database and display the supplier name, address and phone number in a list box.
- Task 2:
  - Add a drop down list to the form that will display all available suppliers’ names.
  - Upon selection of a supplier all the details of supplier will be display in a DetailsView

Demo: New Shipper Page

- Using Northwind database and ADO .NET create a web form to add a new Shipper.

- Task 1:
  - Design the form with required fields.
  - Add proper validations to the fields.
- Task 2:
  - Connect to the database and add the new record.
  - Navigate to a new form and display all available shippers in a GridView.

Exercise: Orders Search

- Using Northwind database and ADO .NET create a web form to Search for Orders details.
- Tasks:
  - Add a text box labeled "Order ID". Next to the text box add a dropdown list with two values:

    Customer Details & Shipper Details.
  - Add a button that the will query the database and will return the order details in DetailsView.
  - Finally, display either *Customer Details* or *Shipper Details* in a list box depending on the dropdown list selection.

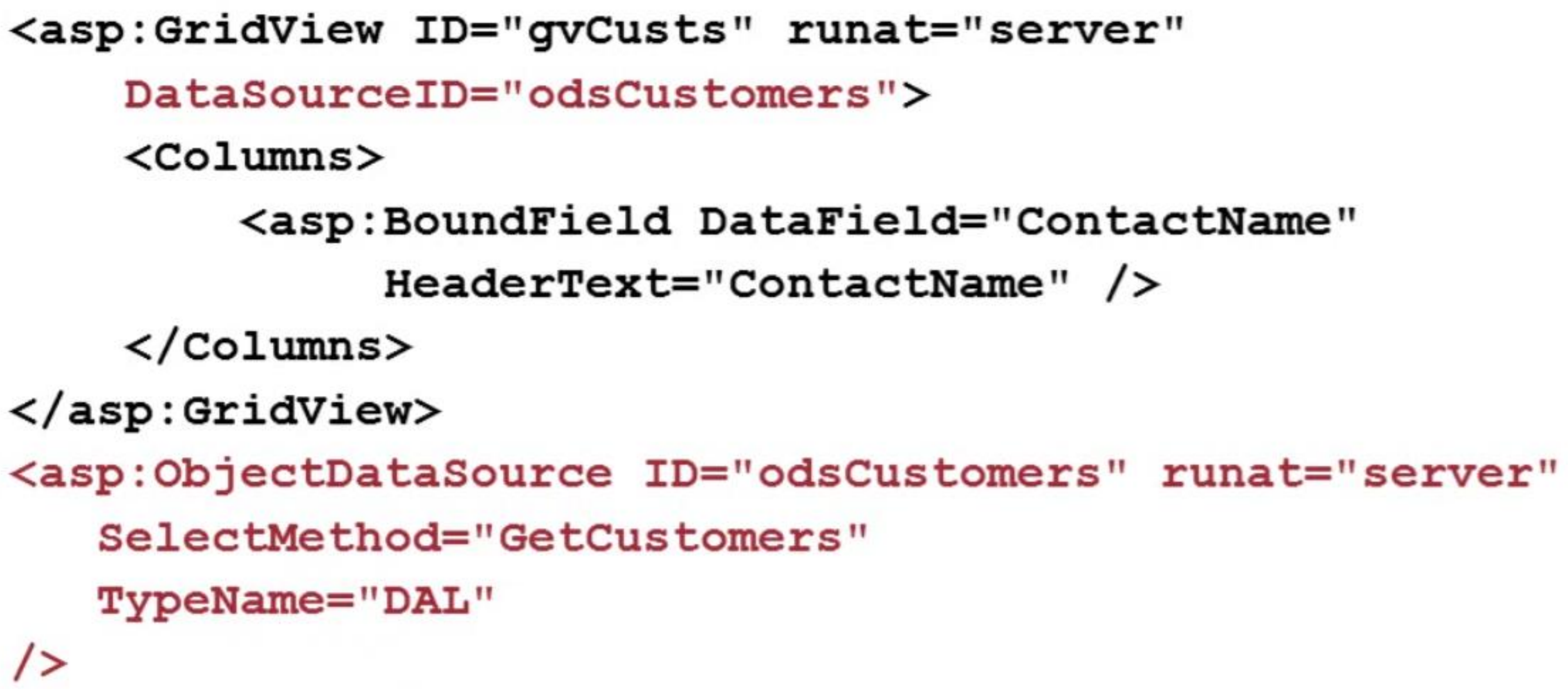
ObjectDataSource Control

The ObjectDataSource Control

- The ObjectDataSource control provides declarative data binding between objects and controls:
  - Supports binding to custom objects such as business objects
  - Allows for N-Tier/N-Layer architectures that separate presentation, business and data layers
  - Supports paging and sorting
  - Supports select, insert, update and delete operations
  - Provides built-in caching support

Using ObjectDataSource Control

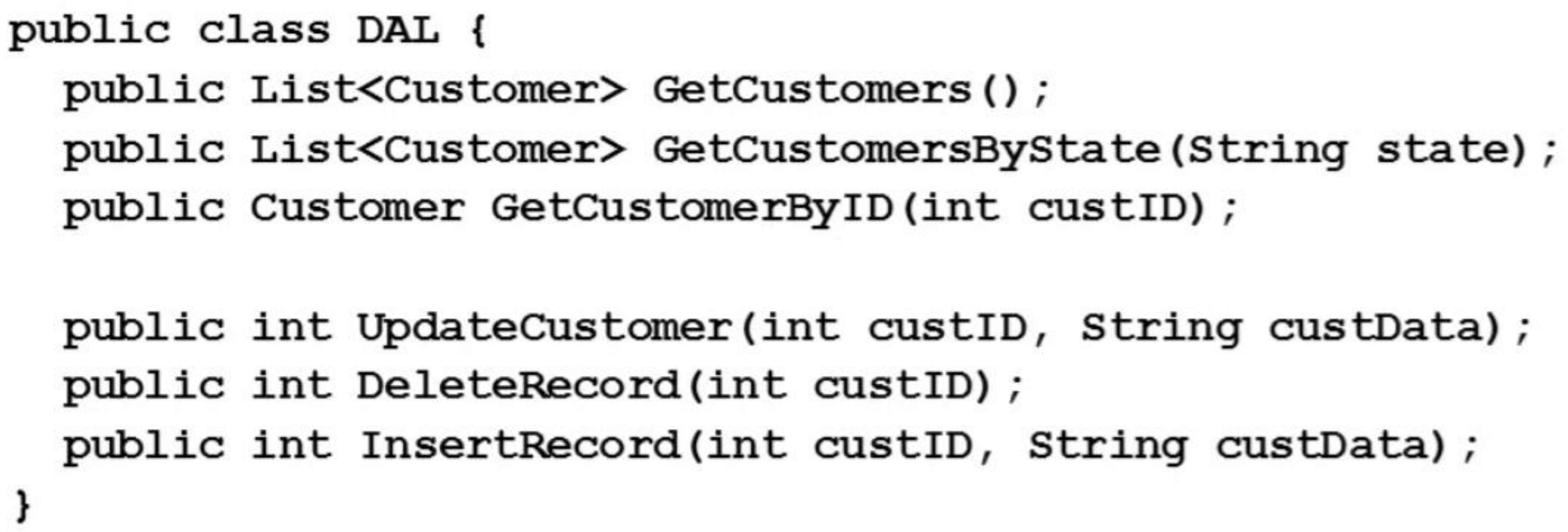
- The ObjectDataSource control allows custom data objects to be bound to ASP .NET controls:



```
<asp:GridView ID="gvCusts" runat="server"
    DataSourceID="odsCustomers">
    <Columns>
        <asp:BoundField DataField="ContactName"
             HeaderText="ContactName" />
    </Columns>
</asp:GridView>
<asp:ObjectDataSource ID="odsCustomers" runat="server"
   SelectMethod="GetCustomers"
   TypeName="DAL"
/>
```

Defining Operations

- Different select, insert, update and delete methods can be defined in business object:



```
public class DAL {
  public List<Customer> GetCustomers();
  public List<Customer> GetCustomersByState(String state);
  public Customer GetCustomerByID(int custID);

  public int UpdateCustomer(int custID, String custData);
  public int DeleteRecord(int custID);
  public int InsertRecord(int custID, String custData);
}
```

Add LINQ to SQL to Visual Studio (1)

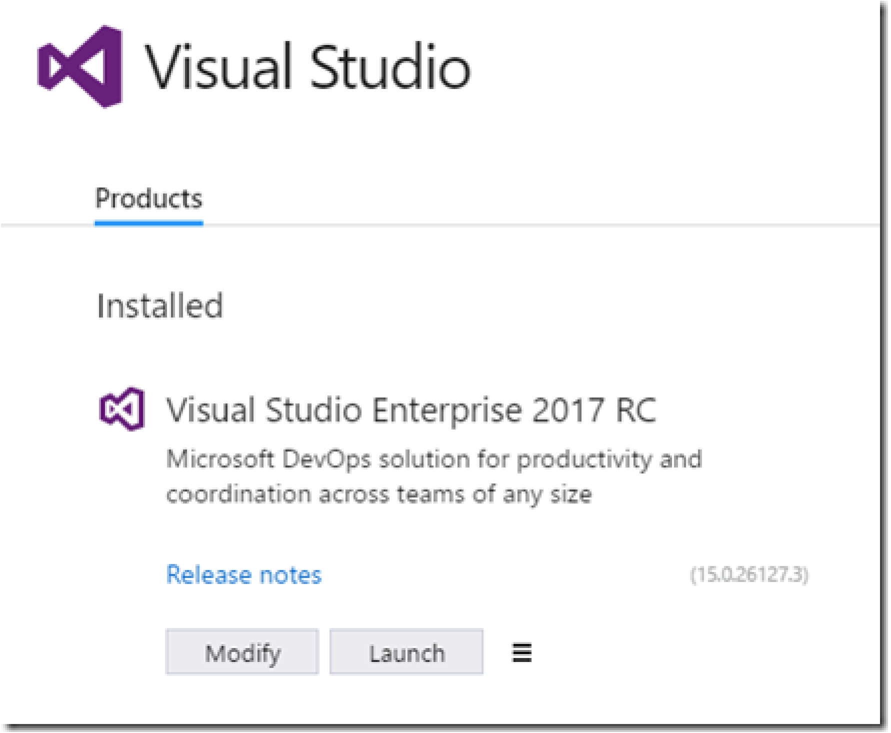
- Go to Control Panel

- Programs → Programs and Feature

- Choose: Visual Studio, right click and **Modify (change)**

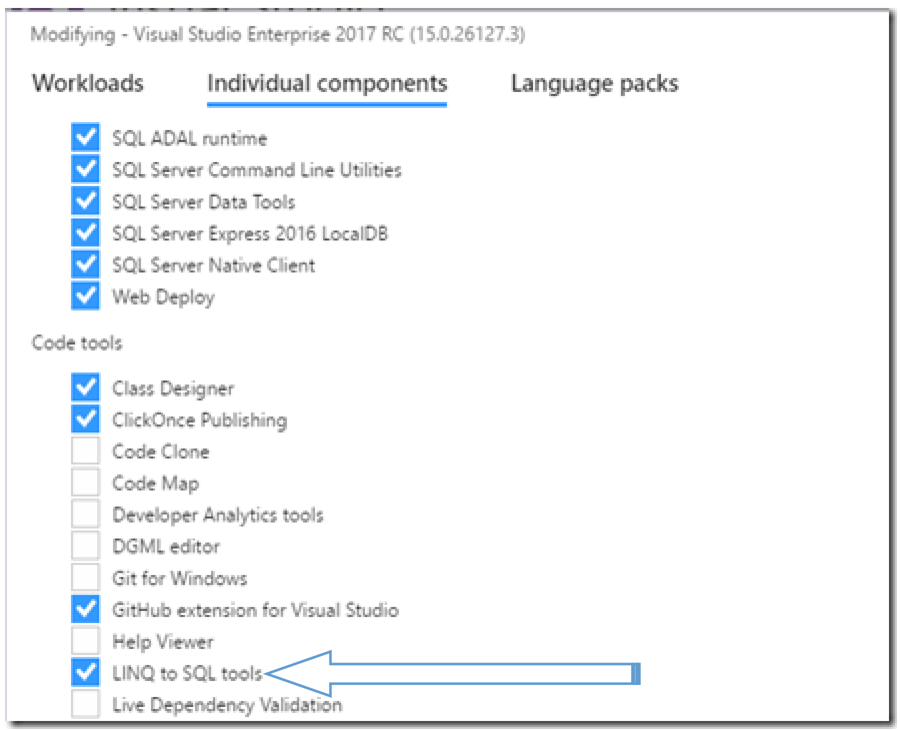
- One the main setup page loads click on **Modify** (might be under **more**)

Add LINQ to SQL to Visual Studio (2)



Visual Studio Enterprise 2017 RC

Microsoft DevOps solution for productivity and coordination across teams of any size


Modifying - Visual Studio Enterprise 2017 RC (15.0.26127.3)

- ☑ SQL ADAL runtime
- ☑ SQL Server Command Line Utilities
- ☑ SQL Server Data Tools
- ☑ SQL Server Express 2016 LocalDB
- ☑ SQL Server Native Client
- ☑ Web Deploy

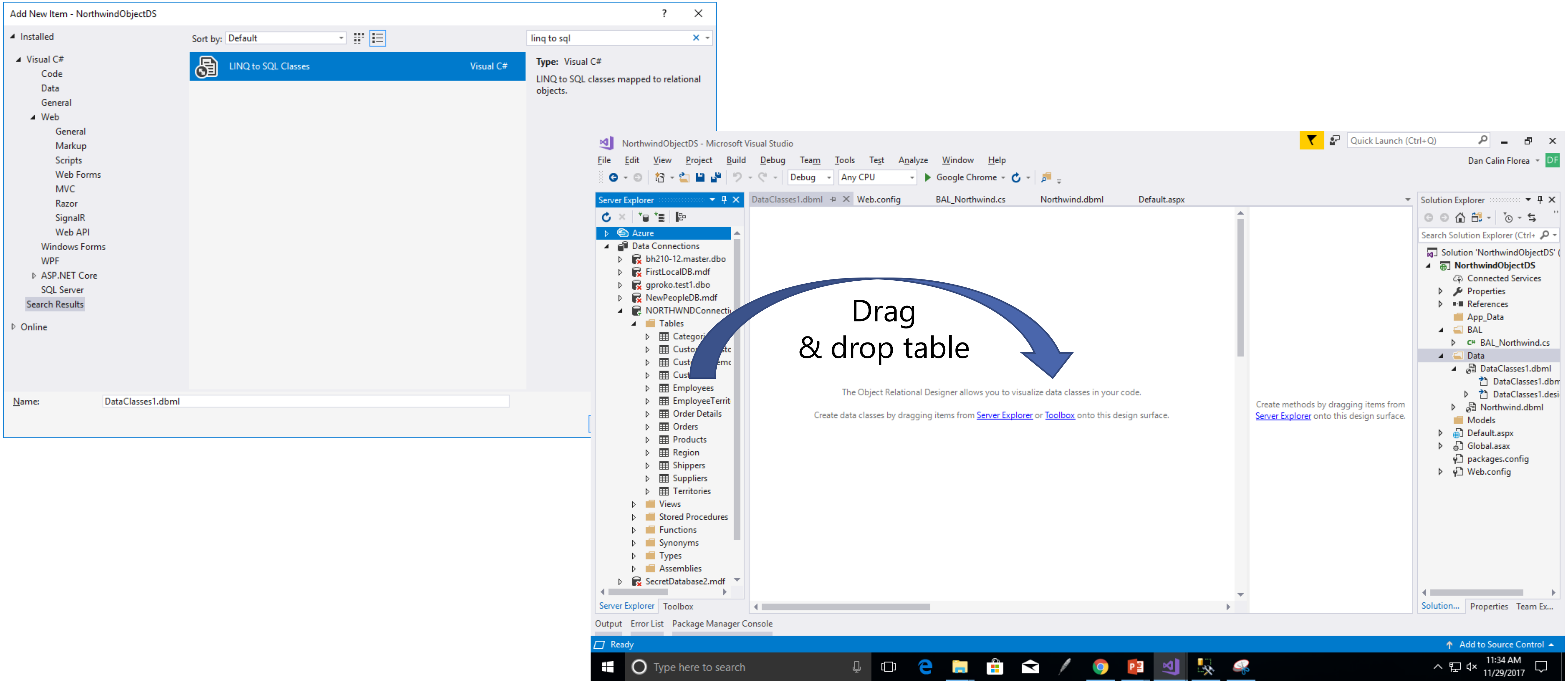
Code tools

- ☑ Class Designer
- ☑ ClickOnce Publishing
- ☐ Code Clone
- ☐ Code Map
- ☐ Developer Analytics tools
- ☐ DGML editor
- ☐ Git for Windows
- ☑ GitHub extension for Visual Studio
- ☐ Help Viewer
- ☑ LINQ to SQL tools
- ☐ Live Dependency Validation

Demo: ObjectDS

- Using the Northwind database, create the following webform.
- Task:
  - Add a LINQ to SQL object
  - Create a class to represent the queries needed.
  - Load all countries in the Customers table in the DB into a drop down list.
  - Upon selection of a country, load all customers from that country into a GridView
  - Final, select a customer from the grid and view all details in a DetailsView.

Demo: Add new item [LINQ to SQL]



Add New Item - NorthwindObjectDS

Sort by: Default

linq to sql

LINQ to SQL Classes Visual C#

Type: Visual C#

LINQ to SQL classes mapped to relational objects.

Name: DataClasses1.dbml

NorthwindObjectDS - Microsoft Visual Studio

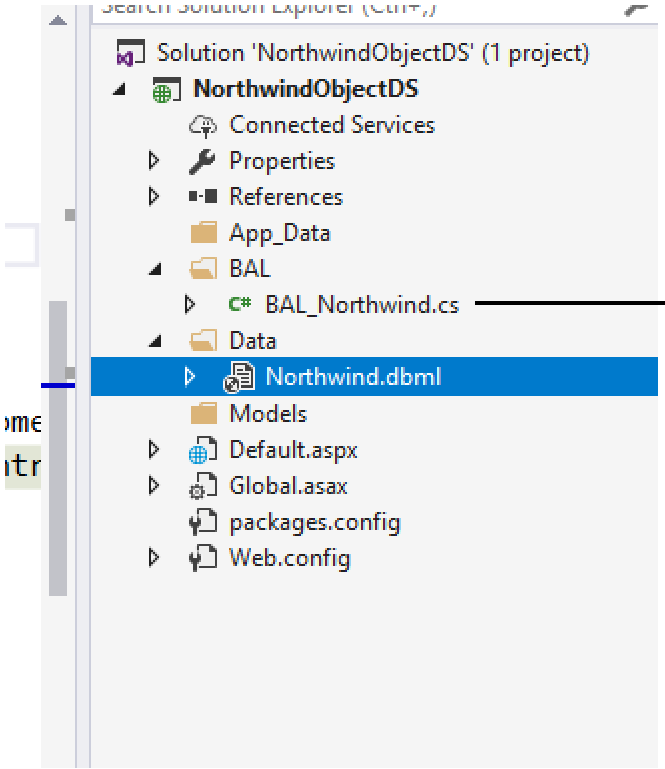
Drag & drop table

The Object Relational Designer allows you to visualize data classes in your code.

Create data classes by dragging items from Server Explorer or Toolbox onto this design surface.

Create methods by dragging items from Server Explorer onto this design surface.

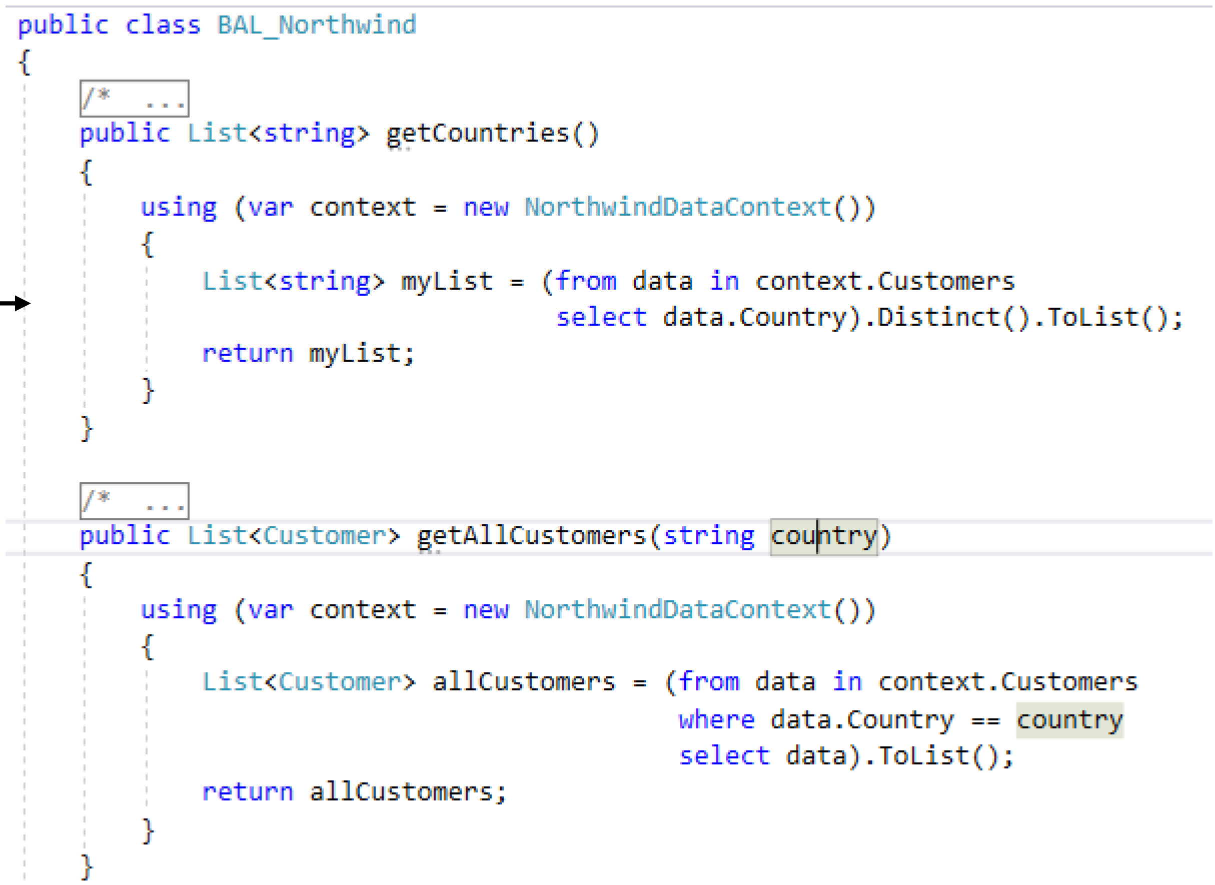
Demo: Add a business access layer class



Search Solution Explorer (Ctrl+;)

- Solution 'NorthwindObjectDS' (1 project)
  - NorthwindObjectDS
    - Connected Services
    - Properties
    - References
    - App_Data
    - BAL
      - BAL_Northwind.cs
    - Data
      - Northwind.dbml
    - Models
    - Default.aspx
    - Global.asax
    - packages.config
    - Web.config

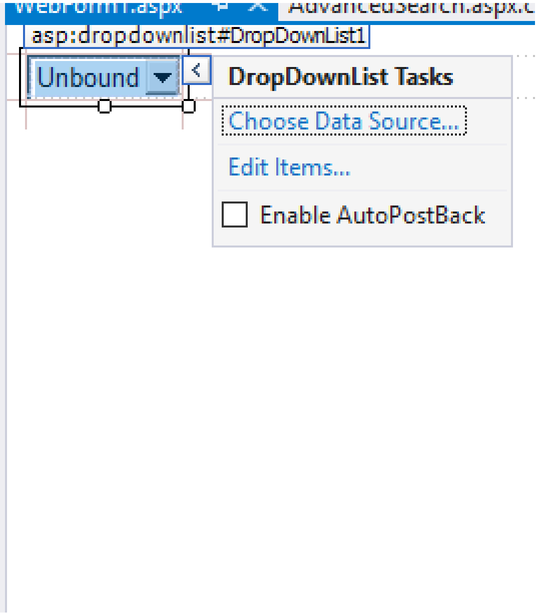
Create a class and write methods to extract data

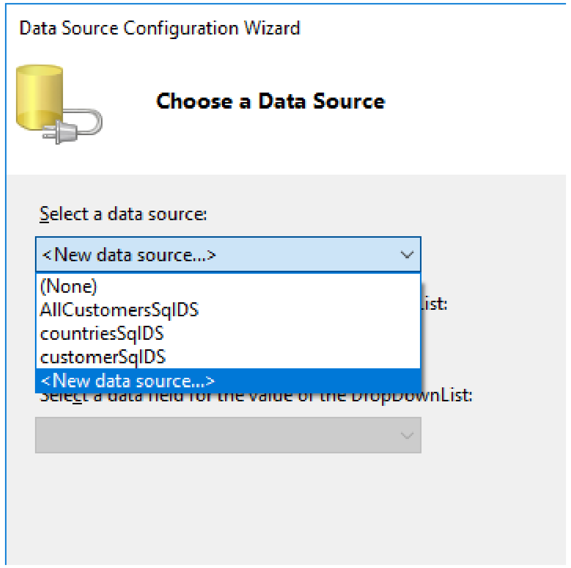


```
public class BAL_Northwind
{
    /* ...
    public List<string> getCountries()
    {
        using (var context = new NorthwindDataContext())
        {
            List<string> myList = (from data in context.Customers
                                   select data.Country).Distinct().ToList();
            return myList;
        }
    }

    /* ...
    public List<Customer> getAllCustomers(string country)
    {
        using (var context = new NorthwindDataContext())
        {
            List<Customer> allCustomers = (from data in context.Customers
                                           where data.Country == country
                                           select data).ToList();
            return allCustomers;
        }
    }
```

Demo: Add an Object Data Source





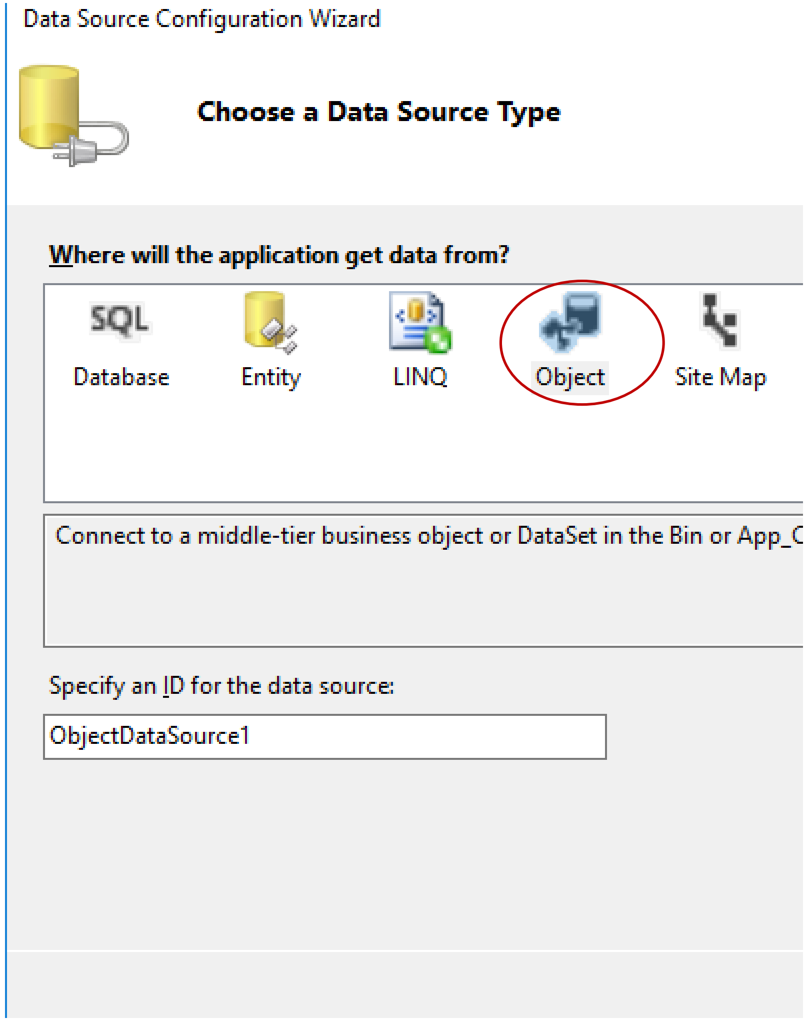
Data Source Configuration Wizard

Choose a Data Source

Select a data source: <New data source...>

- (None)
- AllCustomersSqlDS
- countriesSqlDS
- customerSqlDS
- <New data source...>

Select a data field for the value of the DropDownList:



Data Source Configuration Wizard

Choose a Data Source Type

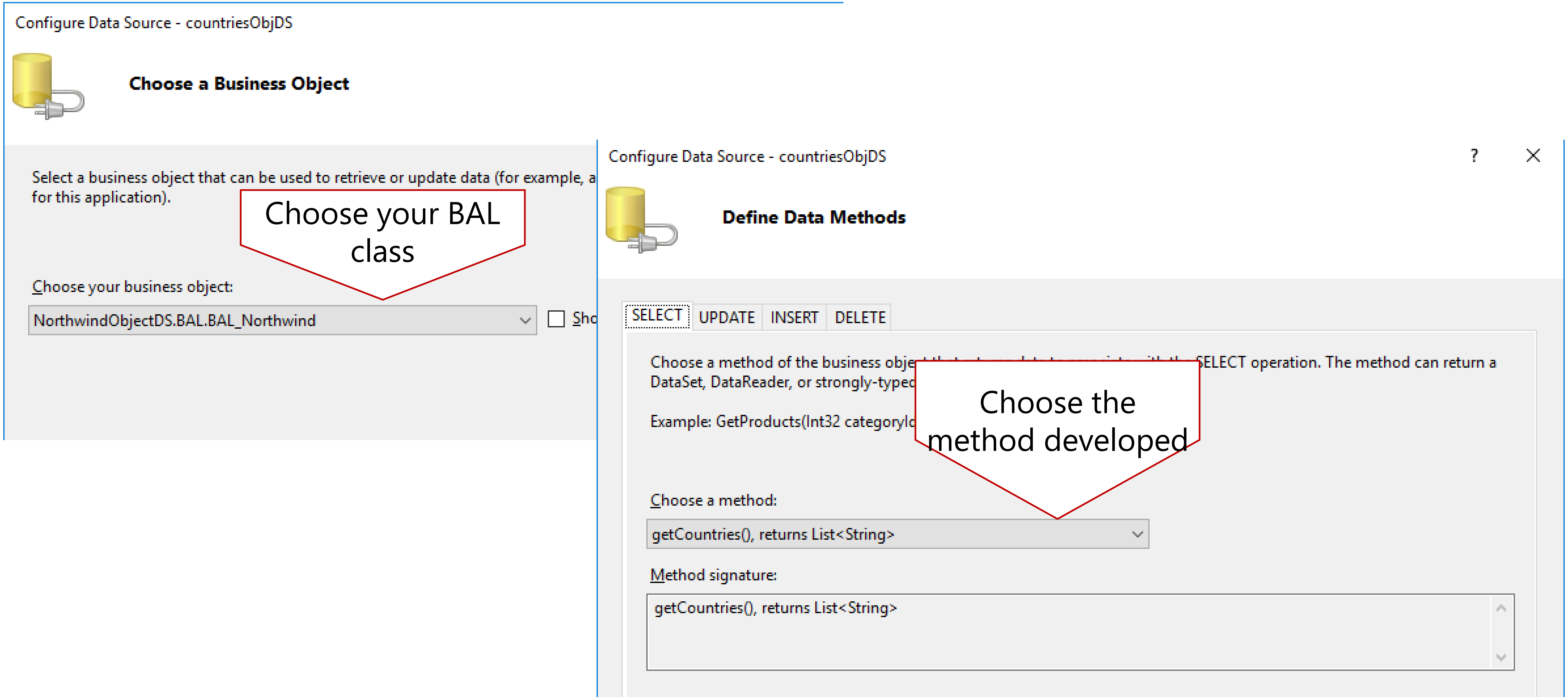
Where will the application get data from?

Connect to a middle-tier business object or DataSet in the Bin or App_C

Specify an ID for the data source: ObjectDataSource1

Choose Data Source → New Data Source

Demo: Load BAL Class



Configure Data Source - countriesObjDS

Choose a Business Object

Select a business object that can be used to retrieve or update data (for example, a
for this application).

Choose your business object:

NorthwindObjectDS.BAL.BAL_Northwind

□ Sho

Choose your BAL class

Configure Data Source - countriesObjDS

Define Data Methods

Choose a method of the business obje SELECT operation. The method can return a DataSet, DataReader, or strongly-typed

Example: GetProducts(Int32 categoryId

Choose a method:

getCountries(), returns List<String>

Method signature:

getCountries(), returns List<String>

Choose the method developed

Exercise: Order Search

- Using Northwind database and Object Data Source (& LINQ to SQL) build the following Order Search web form:
  - Add a calendar web control to your form and add a title for it Order date.
    - *(Note: Set selected date and visible data properties to 1/1/1996)*
  - Below the calendar add a grid view that will display all orders ordered in the date selected in the calendar control.
  - Once a user selects an order from the grid view, the details of the products in the order will be displayed in a DetailsView bellow.

    Product ID, Product Name, Price, Quantity & Discount

Q & A

JOHN ABBOTT
CEGEP/COLLEGE