

Express Edition, In VB.NET

AspxCodeGen 4.0 Express Edition comes in either the C# or VB.NET version. They both work the same way, the only difference is the generated code, one is in C# and the other is in VB.NET. The example here uses the VB.Net version.

The Express Edition is our free product for the AspxCodeGen line. It does not generate as much code as the Professional Plus and the Professional editions. It generates middle tier objects that contain properties and empty methods where you can add your own code.

To generate code make sure you have filled all the required information both in the *Database Settings* tab and the *Generated Code Settings* tab. Please read the *Using AspxCodeGen* document for instructions on how to fill the required information.

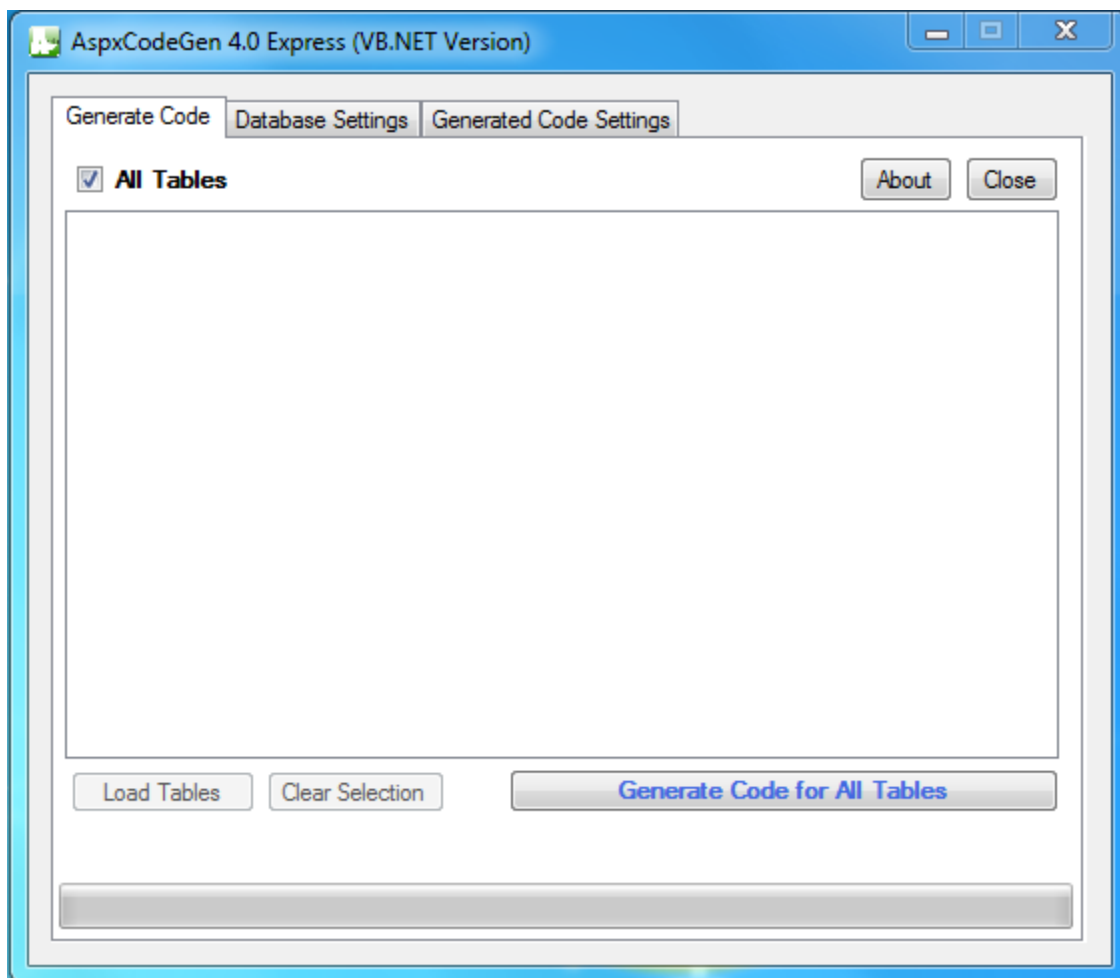


Figure1 Generate Code Window

There are 2 ways you can generate code using AspxCodeGen 4.0:

1. **Generate Code for All Tables:** By default the *All Tables* checkbox is checked. To generate code for the database you specified in the *Database Settings* tab simply click the *Generate Code for All Tables* button. This will generate code for all the tables in your database. See Figure 1.

2. **Generate Code for Selected Tables Only:** To generate code just for the selected tables, uncheck the *All Tables* checkbox. You will notice that the button's text to generate code have changed to *Generate Code for Selected Tables*. Then start choosing the tables you want to generate code for by clicking on the check boxes to the left of them. As soon as you select at least one table, the *Clear* button and the *Generate Code for Selected Tables* button becomes enabled. When done selecting the tables, simply click the *Generate Code for Selected Tables* button. See Figure 2.

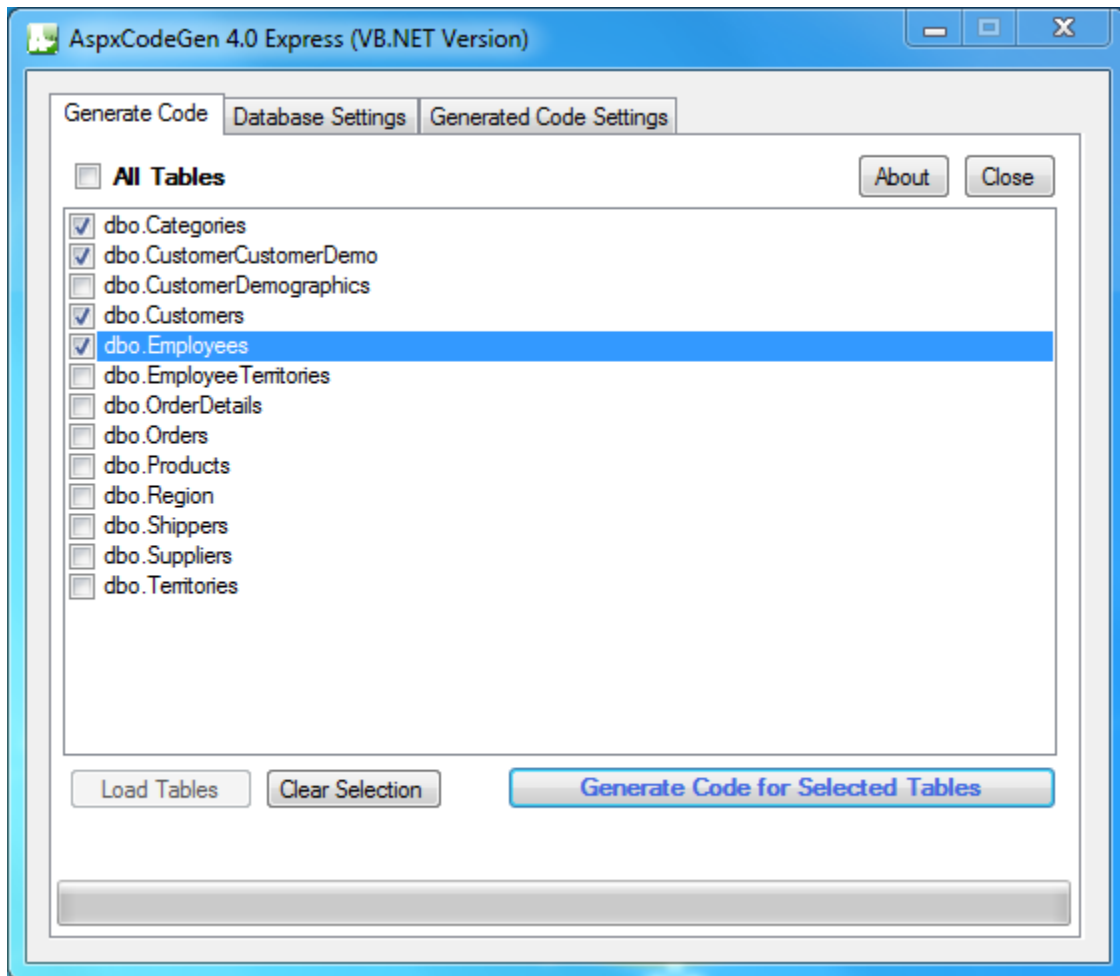


Figure 2 Selecting Specific Tables

The generated code should be found in the *Directory* you specified under *Generated Code Settings*.

Tutorial

For this tutorial, you will need the following:

1. AspxCodeGen 4.0 Express Edition in either the C# or VB.NET version.
2. Northwind database. This can be downloaded at the Microsoft website at: <http://www.microsoft.com/downloads/en/details.aspx?FamilyId=06616212-0356-46A0-8DA2-EEBC53A68034&displaylang=en>.
3. Microsoft SQL Server 2000, 2005, or 2008 (you need to have a *sysadmin* account). You can also attach your database to a Microsoft SQL Express (not discussed here).

4. Microsoft Visual Studio 2008, 2010. To target .NET framework 4.0, you need Visual Studio 2010.

Examining the Database

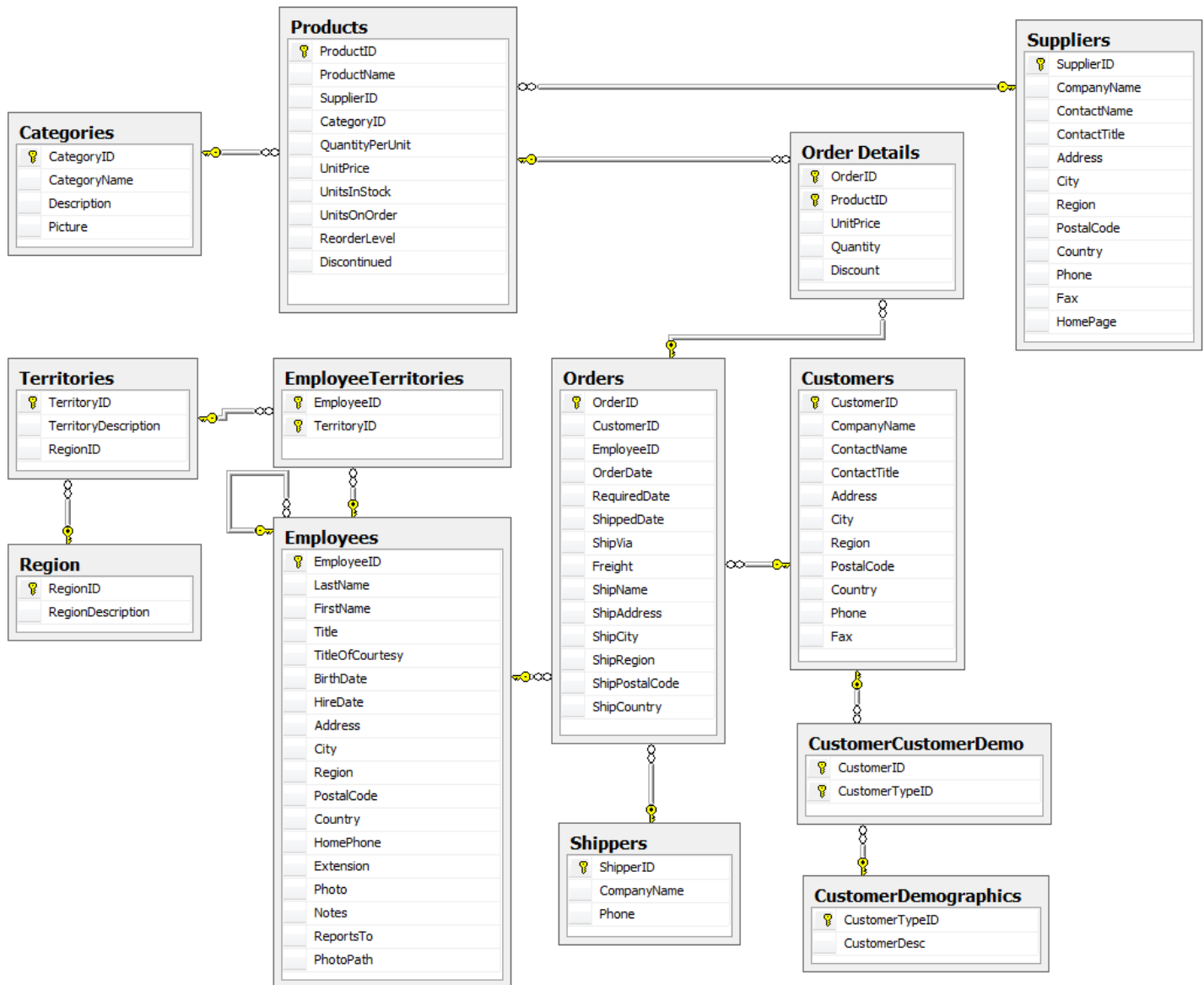


Figure 3 Northwind Database

The Northwind database as shown in Figure 3 has been widely used in examples throughout the development community that's why we're using it here, so some of you may already be familiar with it. It has 13 tables, 11 regular tables and 2 link tables. Link tables are tables that have primary keys and/or foreign keys only, such as the *CustomerCustomerDemo* and *EmployeeTerritories* tables. *Order Details* is also a link table, however for the purposes of this tutorial, since it's not comprised of just primary keys and/or foreign keys, it will not be considered a link table.

Creating a Sample Website

For this tutorial, we're going to create a website that will use the generated Business Layer and Data Layer code. Of course, you can also use a winforms application.

1. Create a website using Visual Studio 2010 targeting the 4.0 .NET framework. Open Visual Studio 2010.
2. Select *File, New Web Site*.

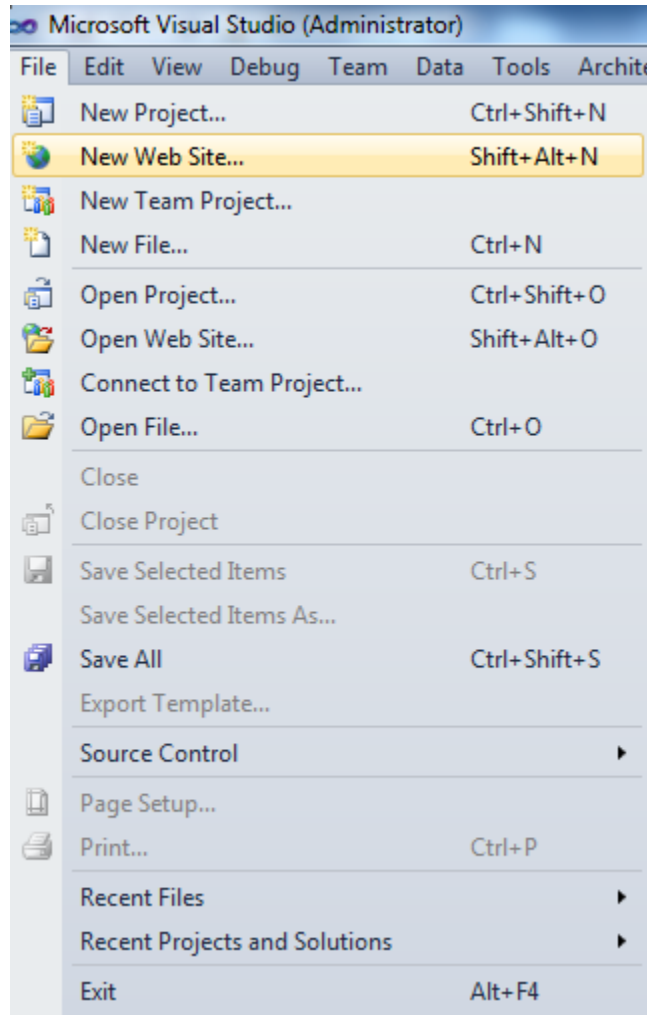


Figure 3 Creating a New Web Site

3. In the *New Web Site* dialog, Select *Visual C#* under *Installed Templates*, select *.NET Framework 4, ASP.NET Empty Web Site*, name the website "*NorthwindWeb*", then click the *OK* button. See Figure 4.
4. A new empty website has now been created, with just the *web.config* file. See Figure 5.
5. Right-click on the *NorthwindWeb* node, select *Add ASP.NET Folder*, then choose *App_Code*. See Figure 6. This is where we're going to place the generated code later on.

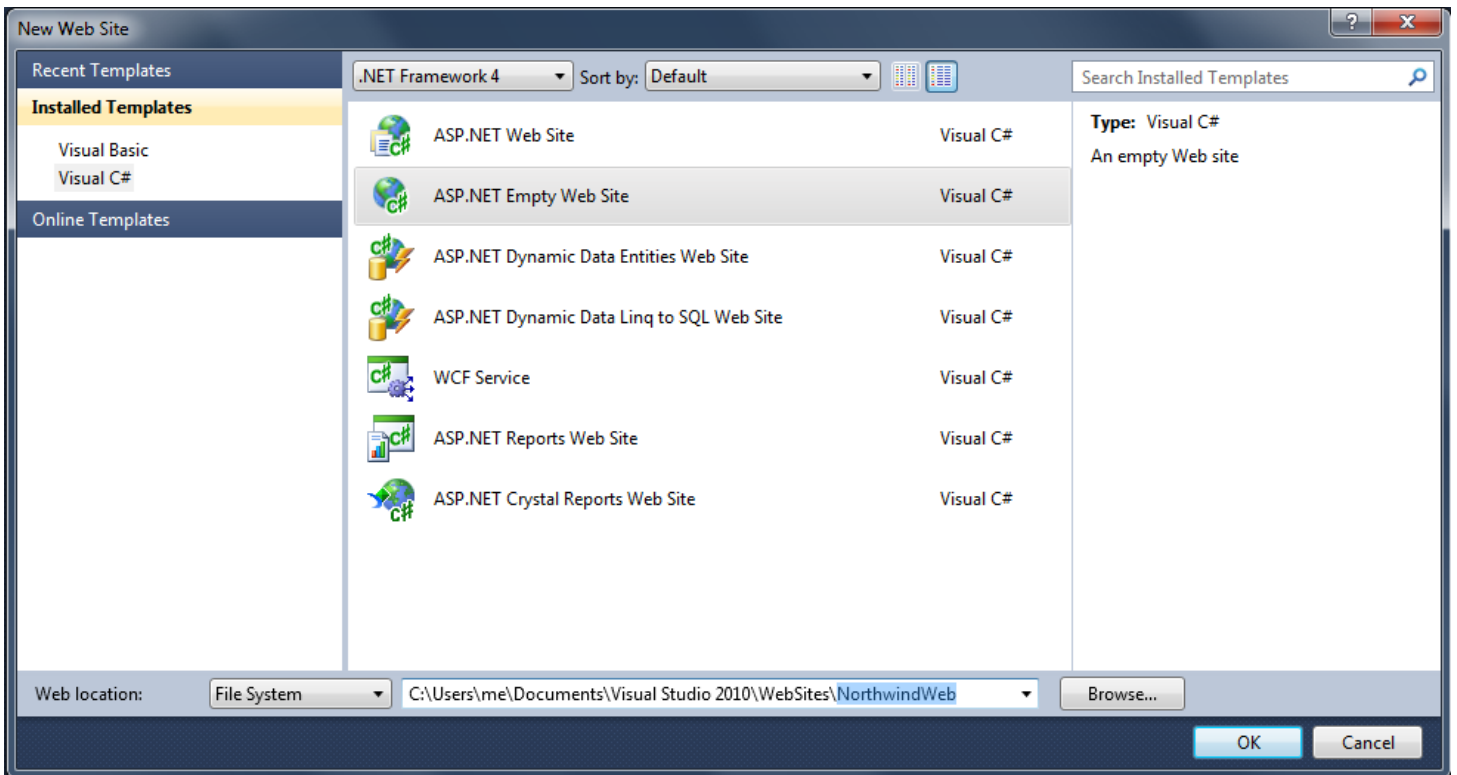


Figure 4 New Web Site

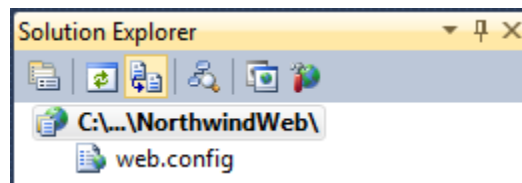


Figure 5 Empty Web Site

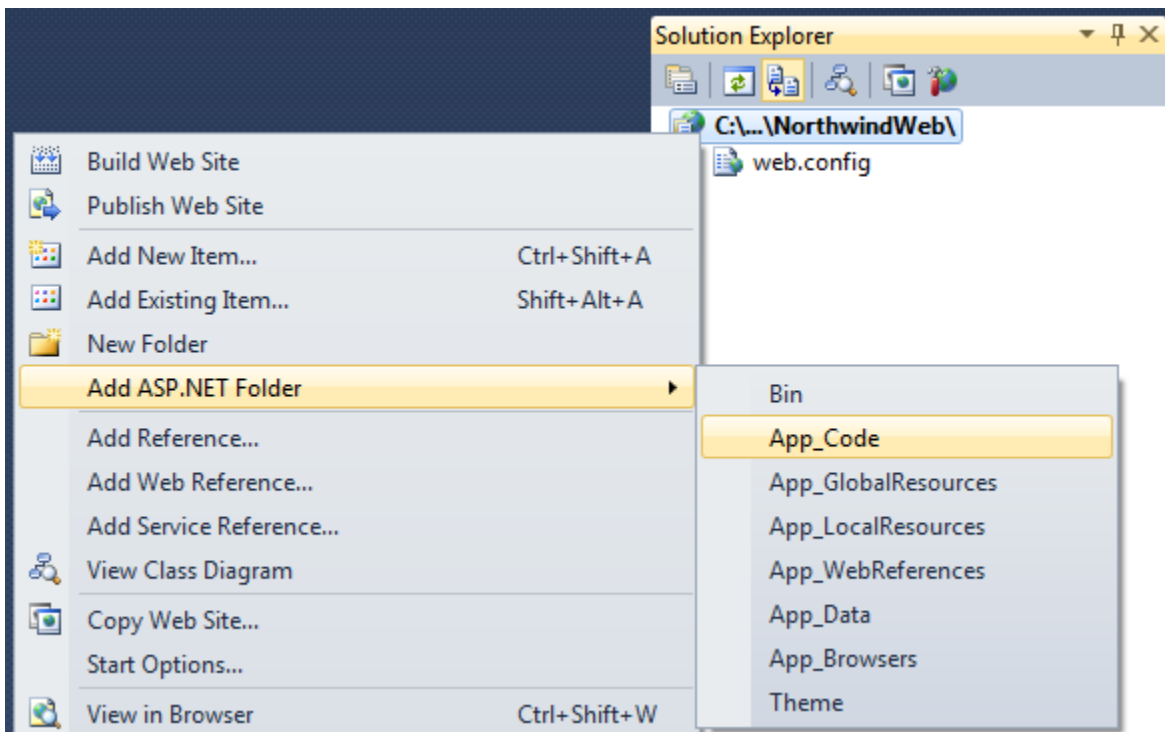


Figure 6 Add ASP.Net Folder App_Code

6. Now add a web form. Right-click on the *NorthwindWeb* node and then choose *Add New Item*. See Figure 7.

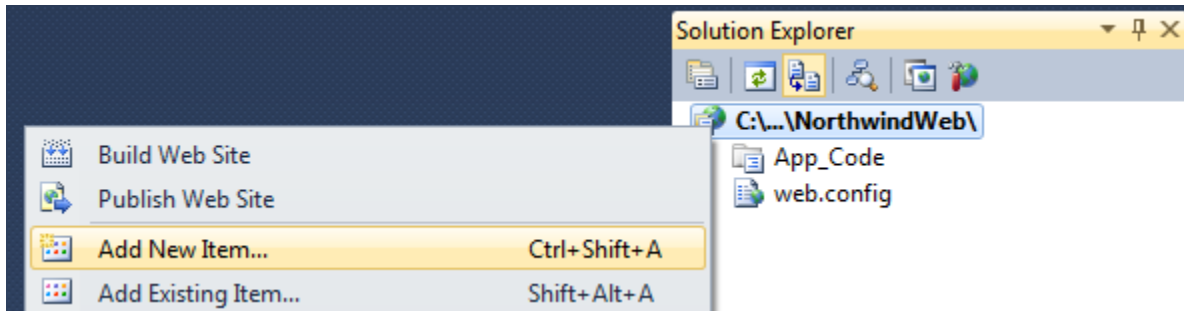


Figure7 Add New Item

7. Choose *Web Form*, then name the web form *Default.aspx*, then click the *Add* button, see Figure 8.

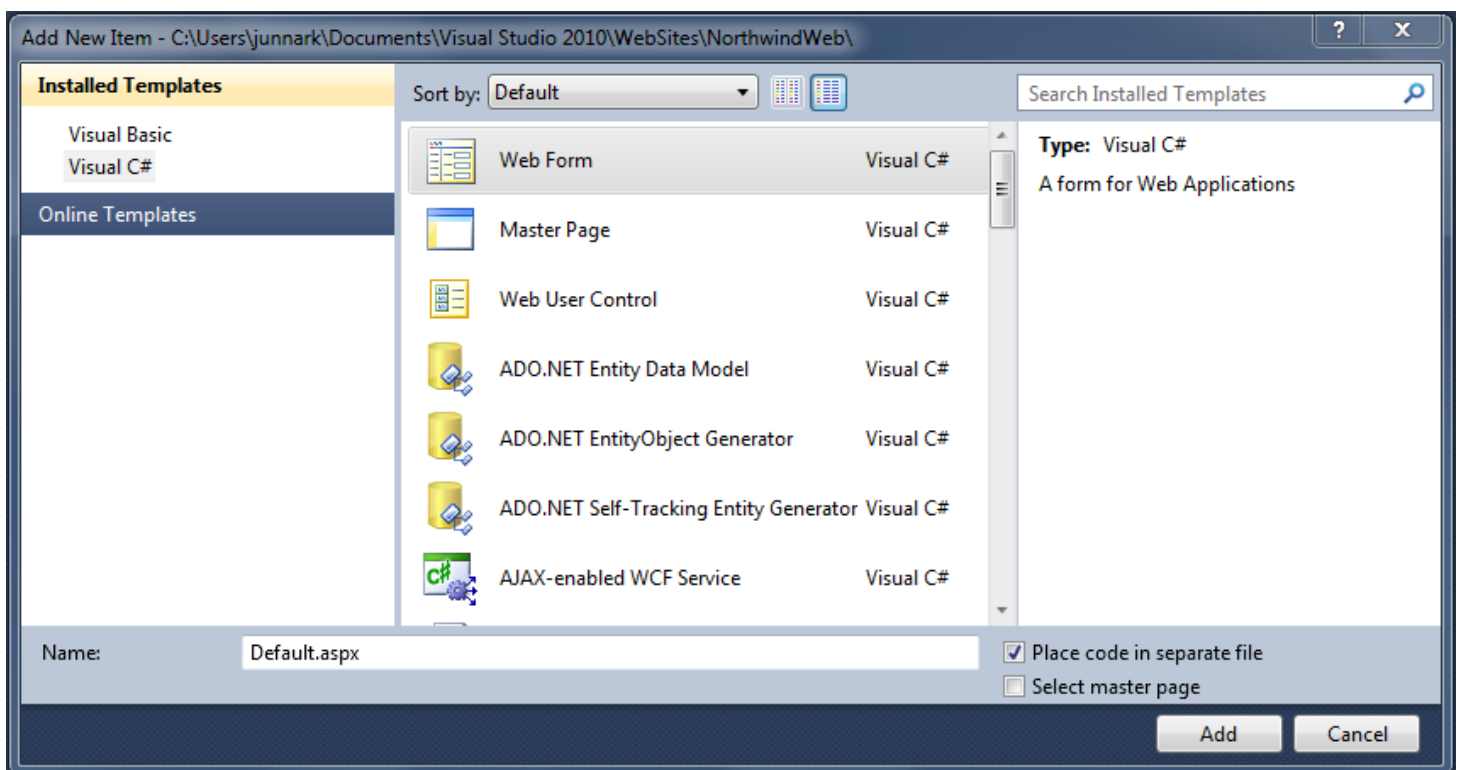


Figure 8 Add Web Form

8. The structure of the newly created website should resemble Figure 9. We will go back to this website later, for now we will be generating code using AspxCodeGen 4.0.

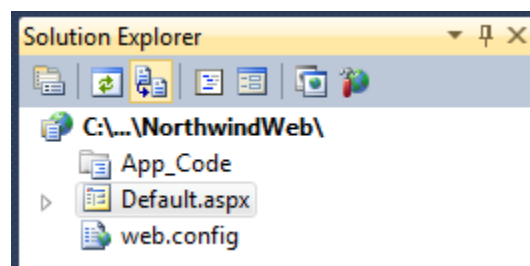


Figure 8 Structure of the New Web Site

Generating Code

1. Open AspCodeGen 4.0.
2. Fill the required information in the *Database Settings* Tab. See *Using AspFormsGen* document for more details. The Express edition does not generate Dynamic SQL or Stored Procedures, that's why they're disabled as shown in Figure 9. Make sure to enter your correct server name, if your MS SQL Server is local to your machine, then using *localhost* may also work for you. Make sure to enter your own User Name and Password.

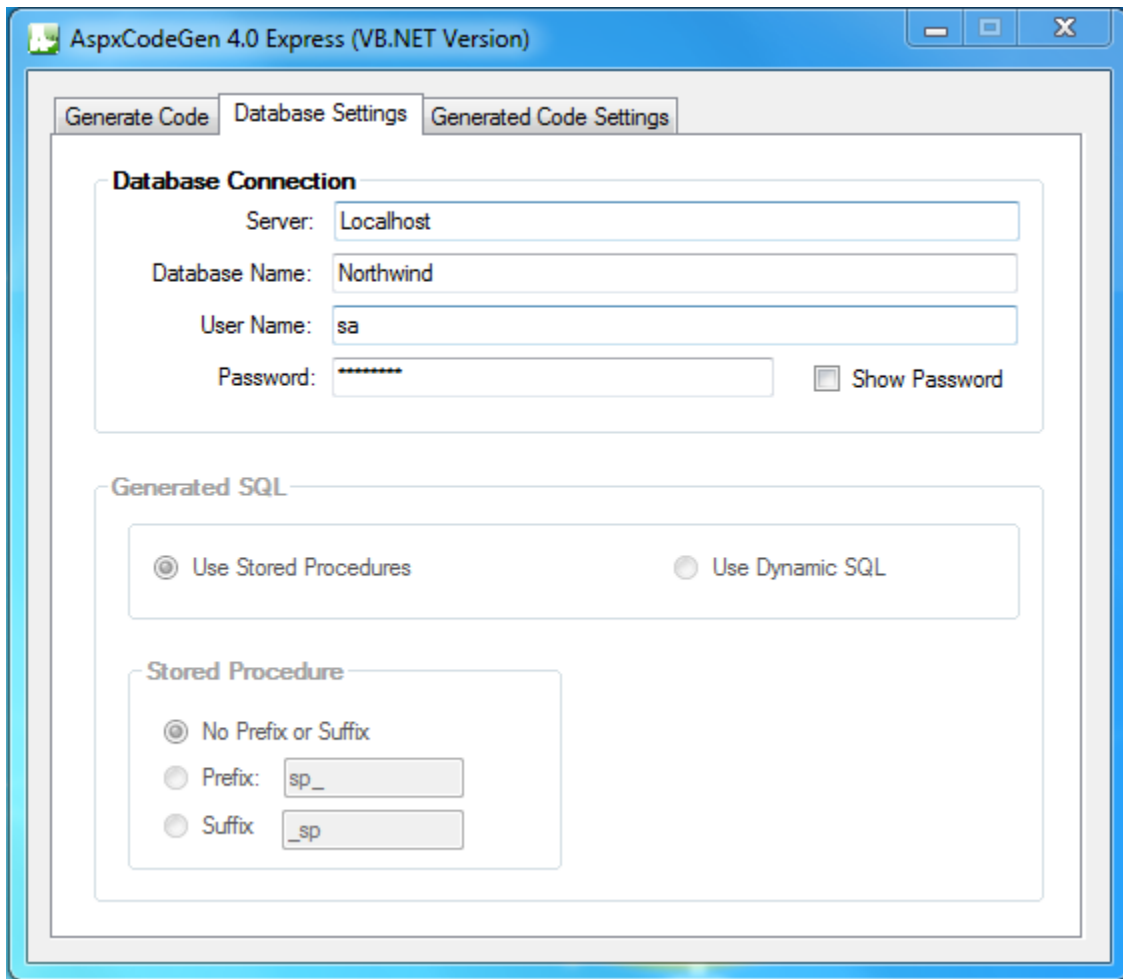


Figure 9 Database Settings Tab

3. Fill the required information in the *Generated Code Settings* Tab as seen in Figure 10. See *Using AspFormsGen* document for more details. In the *Namespace* enter "Northwind", then browse to the *App_Code* directory where you created the *NorthwindWeb* website by clicking the browse button, then click the OK button.
4. Choose *.NET Framework 4.0* under *Target Framework*, see Figure 11.

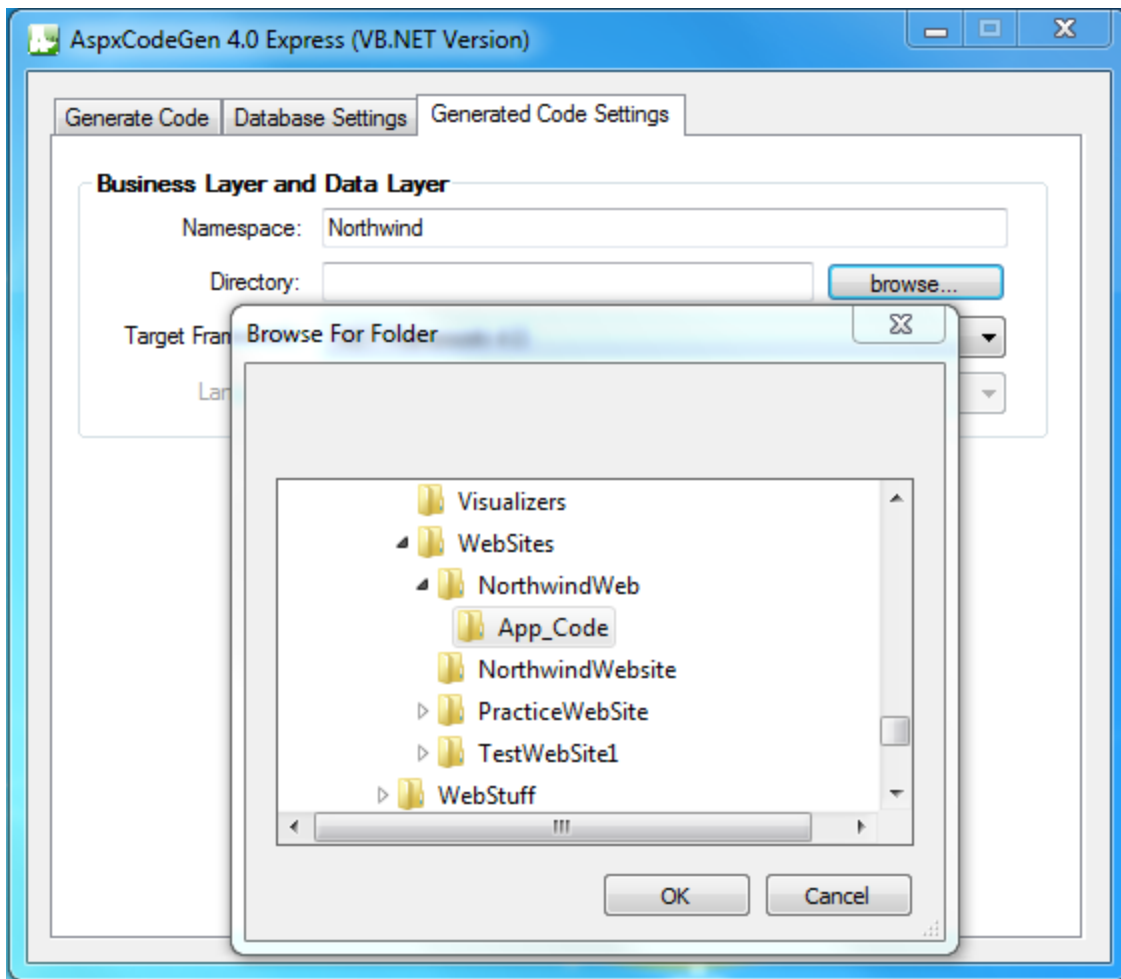


Figure 10 Database Settings Tab

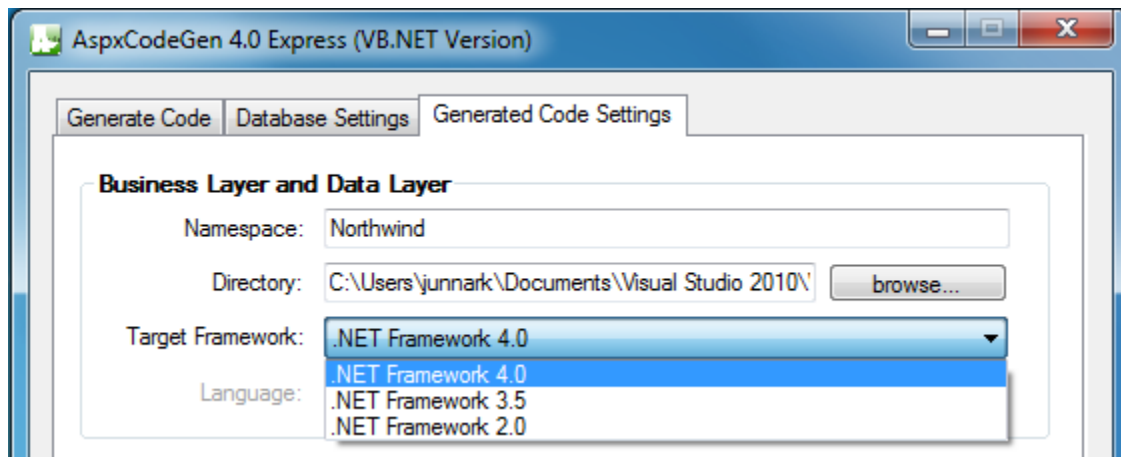


Figure 11 Target Framework

5. Now go back to the Generate Code tab. Click the Generate Code for All Tables button, in a few seconds you will see the progress bar moving, AspxCodeGen 4.0 is now generating code.
6. Once AspxCodeGen 4.0 is done generating code, a pop-up message is shown as seen in Figure 12. Click the OK button. You can now close AspxCodeGen 4.0 by clicking the Close button.

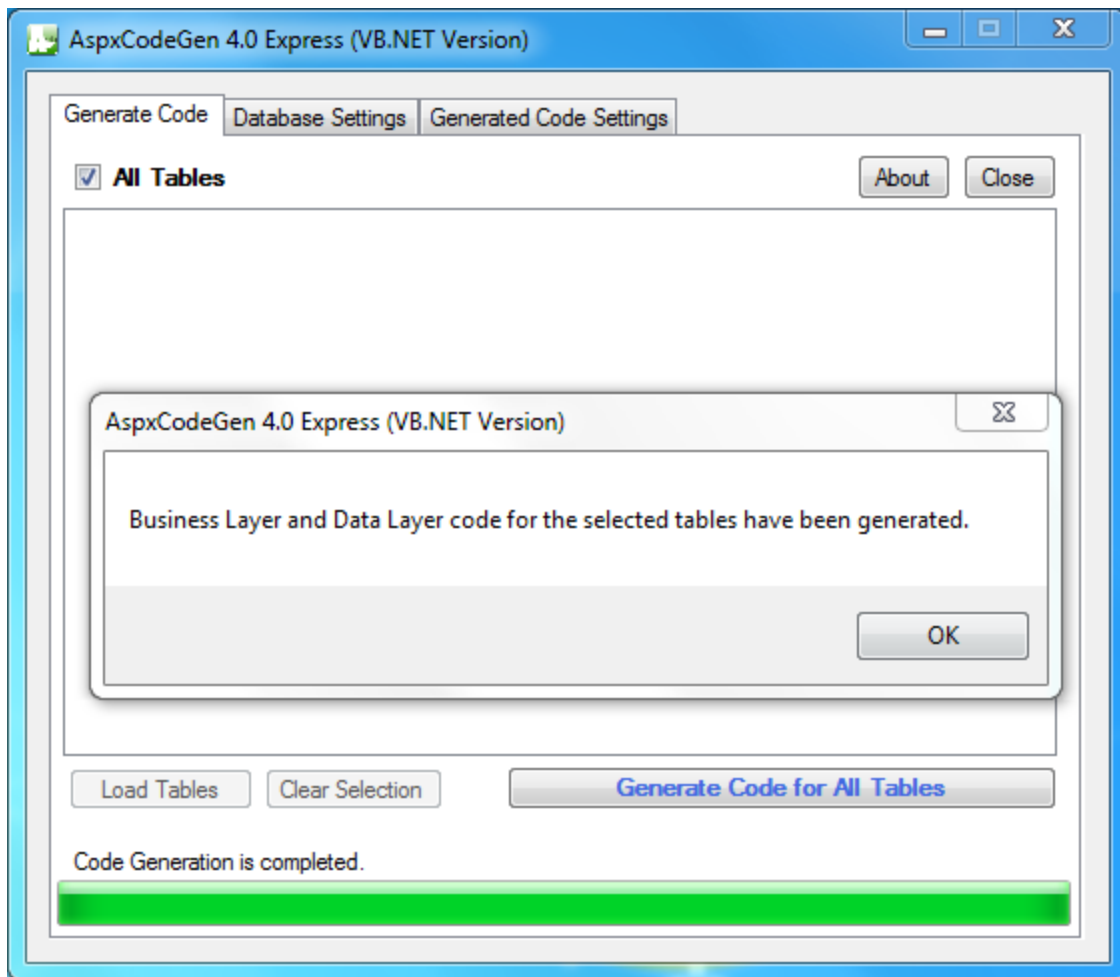


Figure 12 Generating Code

Using the Generated Code

Since we generated the code in the *App_Code* directory of the *NorthwindWeb* web site we can now go back to that web site.

1. From the solution explorer, right-click on the *App_Code* directory, then choose *Refresh Folder*.

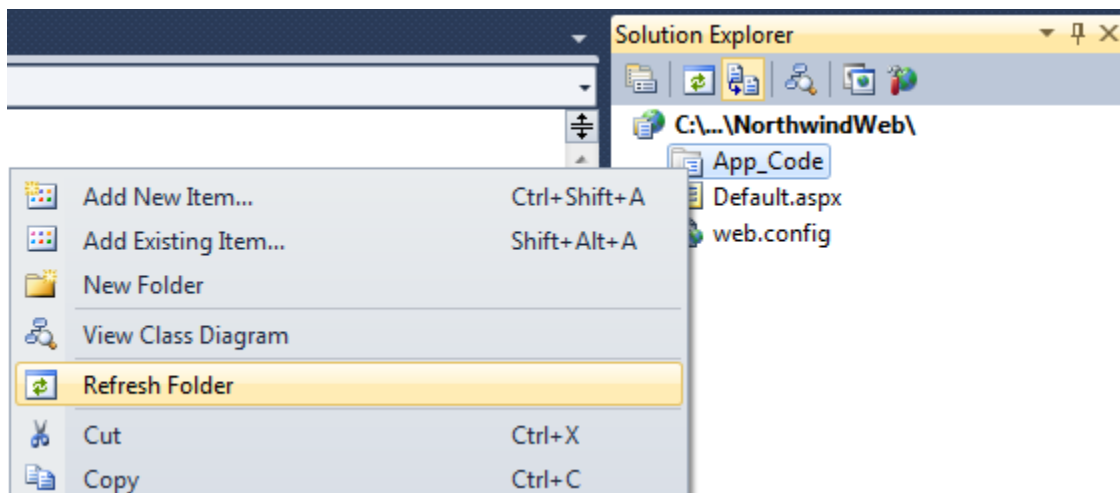


Figure 14 Refresh App_Code Folder

2. You will now notice that there are one (1) new folder under the *App_Code* folder as seen in Figure 15. The generated codes are discussed in another document called *Generated Code Descriptions*.

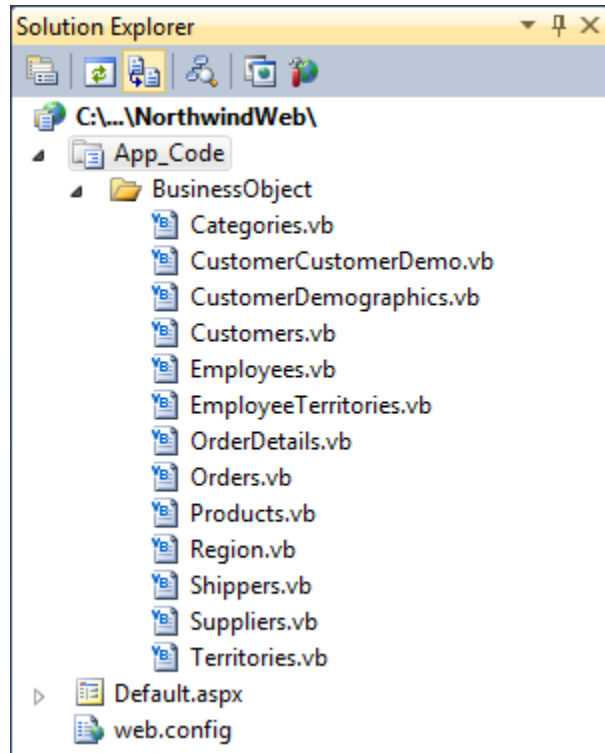


Figure 15 New Folders Under App_Code

3. For now, when you double-click on the Business Object Folder under the *App_Code*. As you can see there are 13 business object classes, one for each of the tables in the *Northwind* database.
4. Double click the Categories.vb class. This will open in the Visual Studio IDE. The code is shown below.

```
Imports System
Imports System.Data
Imports System.Collections.Generic

Namespace Northwind.BusinessObject

    Public Class Categories
        ' private members
        Private _categoryID As Integer
        Private _categoryName As String
        Private _description As String

        ''' <summary>
        ''' Gets or Sets CategoryID
        ''' </summary>
        Public Property CategoryID() As Integer
            Get
                Return _categoryID
            End Get
            Set(ByVal value As Integer)
                _categoryID = value
            End Set
        End Class
    End Namespace
```

```

End Property

''' <summary>
''' Gets or Sets CategoryName
''' </summary>
Public Property CategoryName() As String
    Get
        Return _categoryName
    End Get
    Set(ByVal value As String)
        _categoryName = value
    End Set
End Property

''' <summary>
''' Gets or Sets Description
''' </summary>
Public Property Description() As String
    Get
        Return _description
    End Get
    Set(ByVal value As String)
        _description = value
    End Set
End Property

''' <summary>
''' Constructor
''' </summary>
Public Sub New()
End Sub

''' <summary>
''' Selects a record by primary key(s)
''' </summary>
Public Shared Function SelectByPrimaryKey(ByVal categoryID As Integer)
    As Categories
    Throw New NotImplementedException()
End Function

''' <summary>
''' Selects all records as a collection (List) of Categories
''' </summary>
Public Shared Function SelectAll() As List(Of Categories)
    Throw New NotImplementedException()
End Function

''' <summary>
''' Inserts a record
''' </summary>
Public Function Insert() As Integer
    Throw New NotImplementedException()
End Function

''' <summary>
''' Updates a record
''' </summary>
Public Sub Update()
    Throw New NotImplementedException()
End Sub

```

```

''' <summary>
''' Deletes a record based on primary key(s)
''' </summary>
Public Shared Sub Delete(ByVal categoryID As Integer)
    Throw New NotImplementedException()
End Sub
End Class
End Namespace

```

- As you can see the fields in the Categories table are now represented here as Properties. Each (CRUD) operation is represented as a method. The Express edition does not fully generate all the code needed to do an operation, which is why the methods are not implemented. You can now add your own code inside these methods and remove the `Throw New NotImplementedException()` line of code.
- We can now access these middle tier objects from your client (web form, win form, web services, etc.). In this tutorial, we're going to access these objects from a web form. Open the code behind for the Default.aspx.vb and in the page load event enter the codes shown below. Notice that you will need to reference your middle tier objects as highlighted below.

```

Imports Northwind.BusinessObject

Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
        Handles Me.Load

        ' select all
        Dim objCategories = Categories.SelectAll()

        ' select by primary key
        Dim objCategory = Categories.SelectByPrimaryKey(1)

        ' insert a new record
        Dim cat As New Categories()
        cat.CategoryID = 1
        cat.CategoryName = "New Category"
        cat.Description = "Description"
        cat.Insert()

        'update an existing record
        Dim objCat As New Categories()
        objCat.CategoryID = 1 ' update this category
        objCat.CategoryName = "Updated Category"
        objCat.Description = "Update Description"
        objCat.Update()

        ' delete a record by primary key
        Categories.Delete(1)
    End Sub
End Class

```

- Coding this is really simple because Visual Studio's intellisense picks up on the properties and methods that are available through these middle tier objects, See Figure 15 and 16.

```

1 Imports Northwind.BusinessObject
2
3 Partial Class _Default
4     Inherits System.Web.UI.Page
5
6     Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
7         ' select all
8         Dim objCategories = Categories.SelectAll()
9
10        ' select by primary key
11        Dim objCategory = Categories.
12    End Sub
13 End Class
14

```

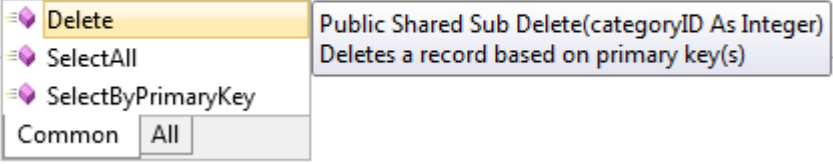


Figure 15 Available Middle Tier Methods

```

1 Imports Northwind.BusinessObject
2
3 Partial Class _Default
4     Inherits System.Web.UI.Page
5
6     Protected Sub Page_Load(ByVal sender As Object, ByVal e As
7         ' select all
8         Dim objCategories = Categories.SelectAll()
9
10        ' select by primary key
11        Dim objCategory = Categories.SelectByPrimaryKey(1)
12
13        ' insert a new record
14        Dim cat As New Categories()
15        cat.
16    End Sub
17 End Class
18

```

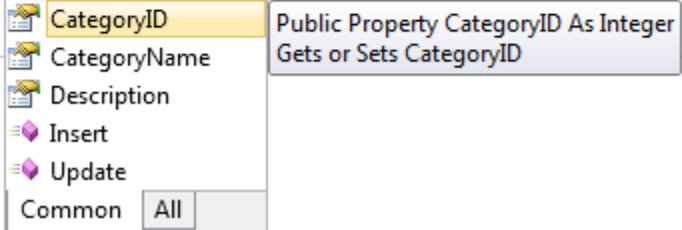


Figure 16 Available Middle Tier Properties

End of tutorial.