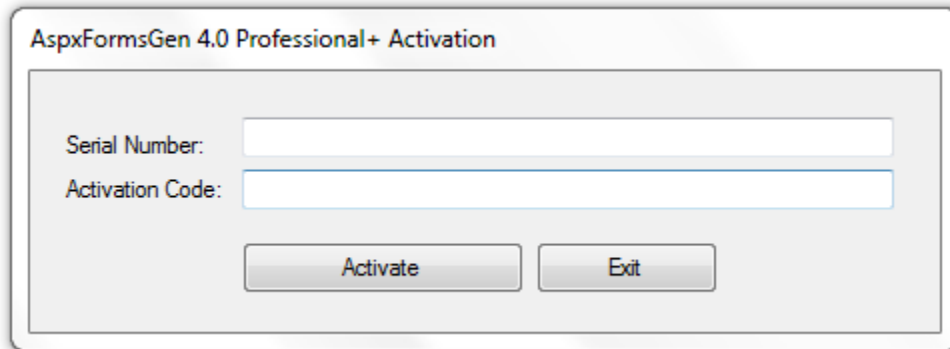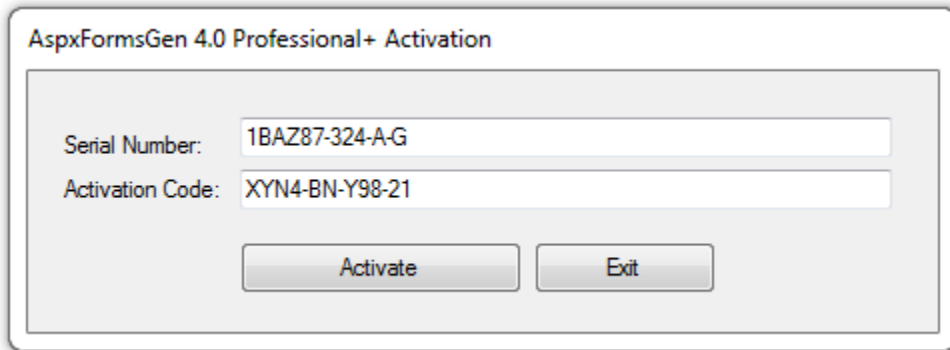# Activating AspxFormsGen 4.0

The first time you open AspxFormsGen 4 Professional Plus edition you will be presented with an activation form as shown in Figure 1. You will not be shown the activation form when using the Express edition.

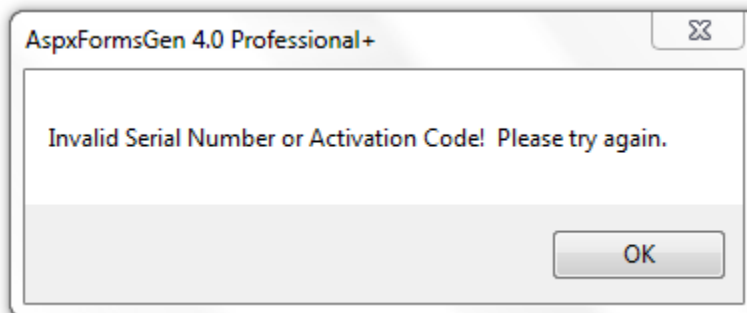
Figure 1 Activation Form

You need to enter the *Serial Number* and *Activation Code* that we provided you in the respective text boxes. The *Serial Number* and *Activation Code* contains dashes "-", make sure to include these dashes when entering them. Then click the *Activate* button. See Figure 2.

Figure 2

If you enter an invalid *Serial Number* or *Activation Code* a warning pops up. When this happens, click the *Ok* button (See Figure 3), then re-enter the *Serial Number* and *Activation Code* Again.

Figure 3 Invalid Serial Number or Activation Code

If you enter a valid *Serial Number* and *Activation Code*, you will be presented with the main window of the application, see Figure 4. You will only see the activation form once. After that, you will always go straight to the main window.
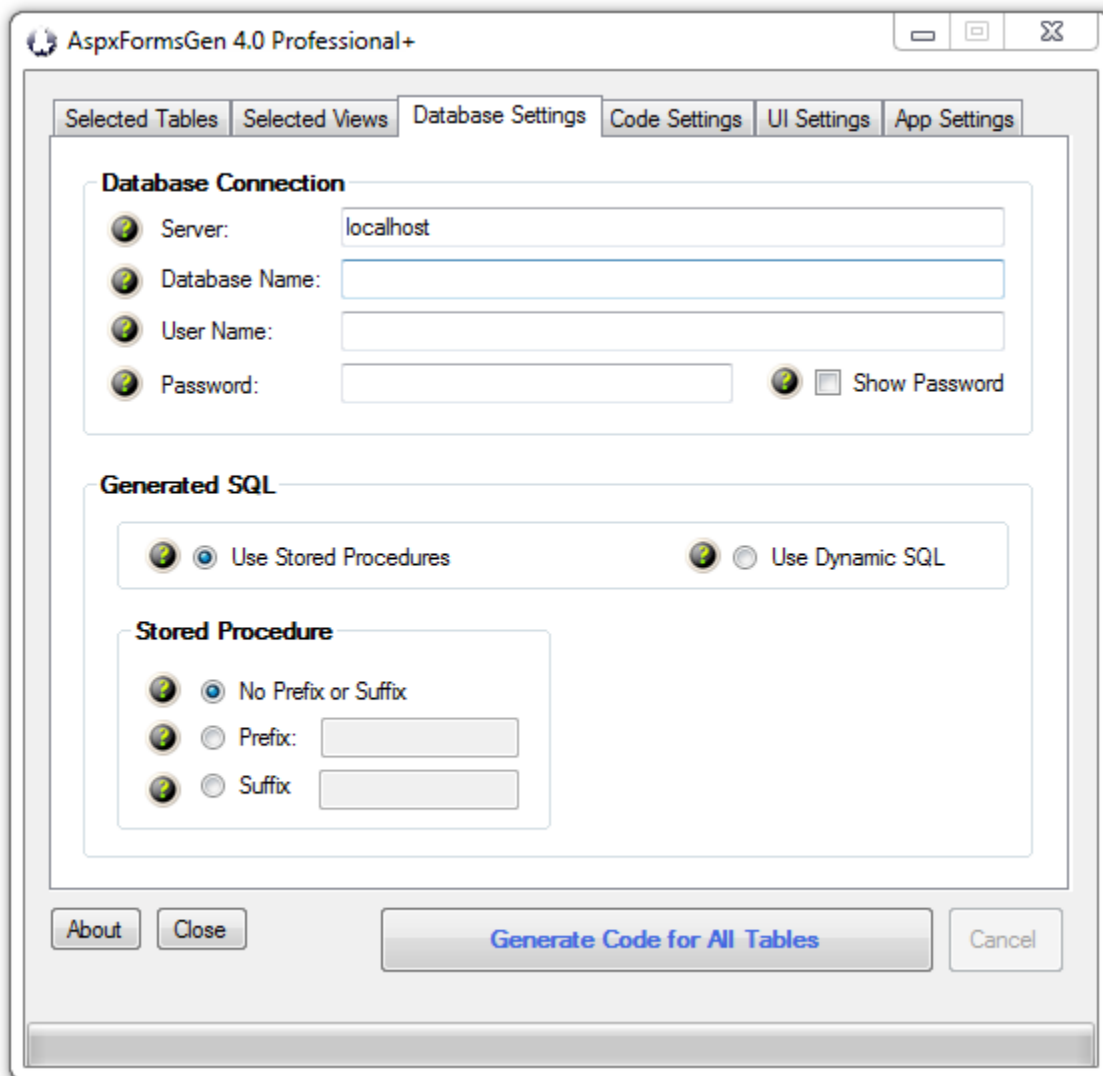
Figure 4 Main Window

A Simple Interface

To keep AspxFormsGen 4.0 simple, there's only one main interface as shown in Figure 4. The main window consists of six (6) tabs.

1. **Selected Tables**: AspxFormsGen generates code from all the tables in your database by default. You can choose to generate from selected tables only from the Code Settings tab, and then select just the tables to generate from on this tab. This is a **new feature** for AspxFormsGen 4.0.

2. **Selected Views**: You can choose to generate from selected views only from the Code Settings tab, and then select just the views to generate from on this tab. This is a **new feature** for AspxFormsGen 4.0.

3. **Database Settings**: This is where you enter the database you want to generate code from and whether you want to generate stored procedures or dynamic SQL. This is probably going to be your most used tab.

4. **Code Settings**: You'll find a selection here on where to generate your objects from: all tables, all views (**new**), selected tables (**new**), or selected views (**new**). This is also where you set the web site name, the root directory where you want the website to be generated, the namespace for your code, and most of all the language (either C# or VB.NET) you want the generated code to be in.

5. **UI Settings**: You can customize your own settings for the generated ASP.NET web forms here. You can choose themes for the GridView control and JQuery UI controls. You can also choose the kind of page validation. You can also select which web forms to generate and the folders or file prefix to use for each web page. This is a **new feature** for AspxFormsGen 4.0.

6. **App Settings**: These are application settings. Almost all generated code/web forms are overwritten every time you use AspxFormsGen 4.0. However, you can choose not to overwrite some key files from here. You can also reset all settings to its original default from here. This is a **new feature** for AspxFormsGen 4.0.

That seems like a lot of features, you're probably asking "*where's the One Click feature?*" Since AspxFormsGen 4.0 remembers the last settings you used such as, e.g. server, database name, directory, namespace, language etc., the next time you open AspxFormsGen 4.0, you can just click the Generate... button, that simple.

A Quick Tour

Let's learn how to generate ASP.NET 4.0 Web Forms, middle-tier, and data-tier codes using AspxFormsGen 4.0. **We're going to use *Microsoft's Northwind database* for this demo. Please google and download it.** Or you can use your own database, just follow the steps shown below.

1. Click the "*Generate Code for All Tables*" button. Notice that an error pops up showing us the required fields to fill. See Figure 5.
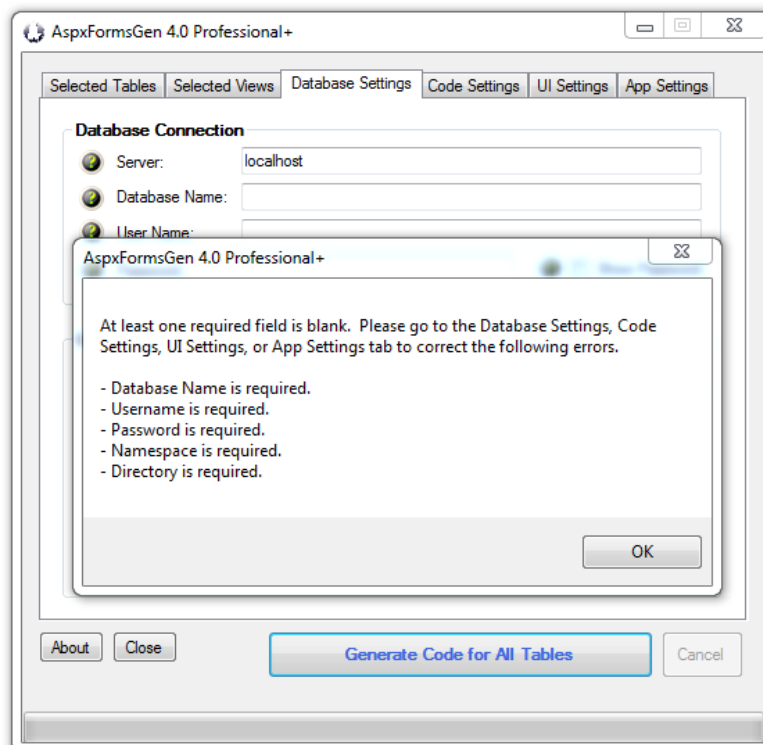

Figure 5 Error Message

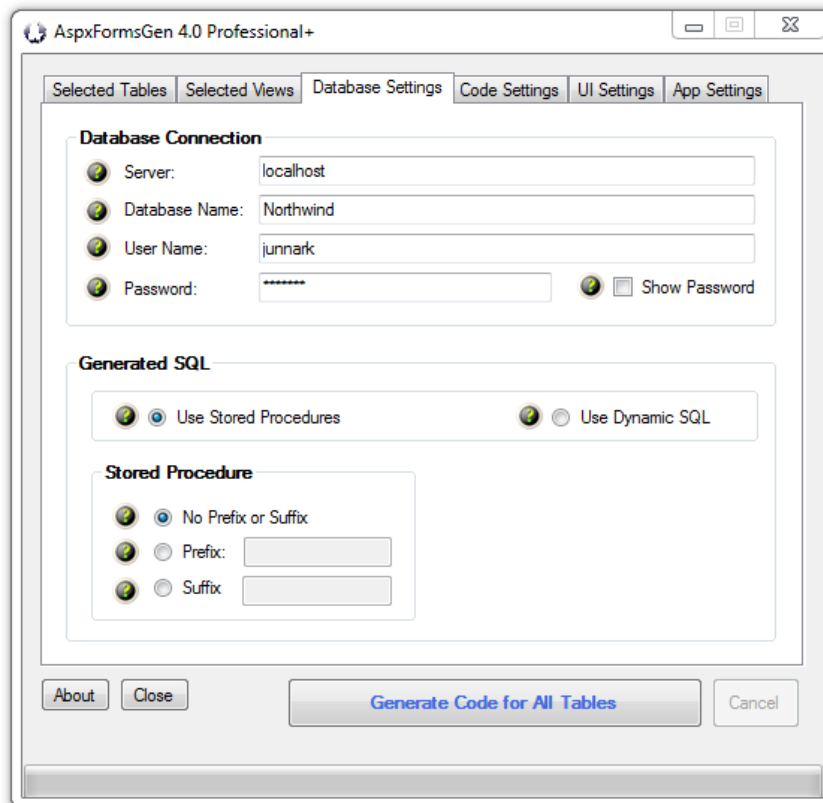2. Click the OK button to close the pop up. Go to the *Database Settings* tab and start filling the required fields. Do the same in the *Code Settings* tab. See Figures 6 and 7.
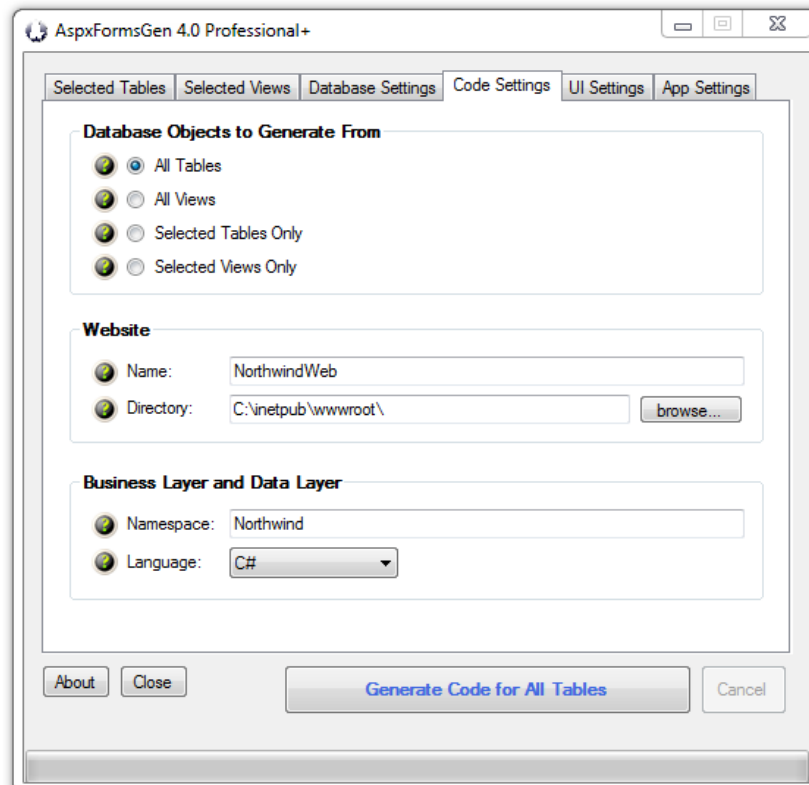

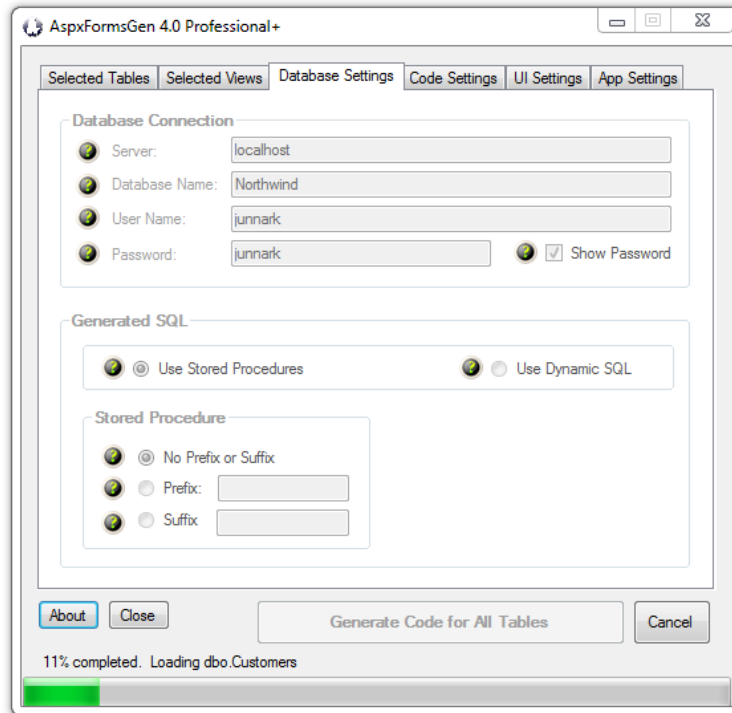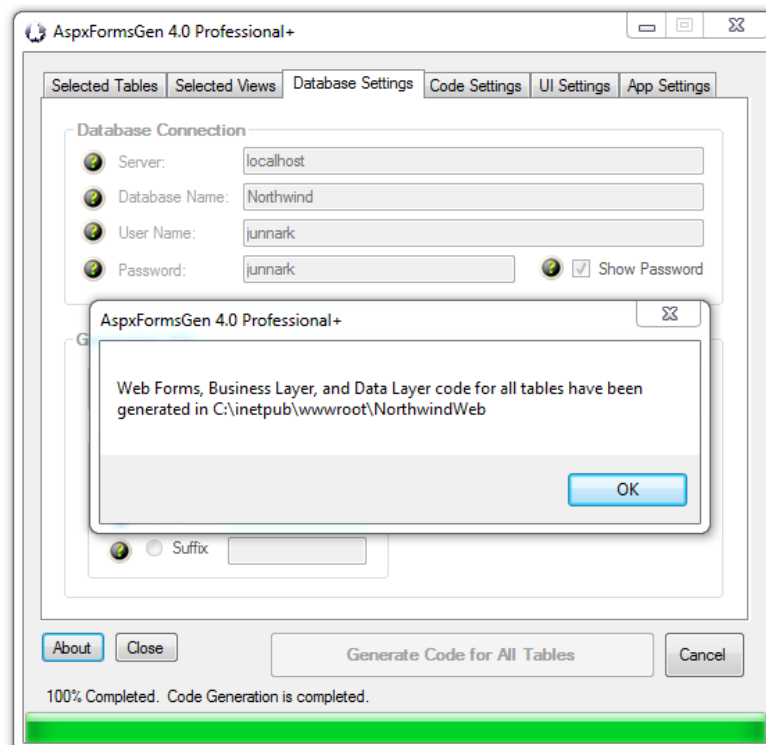**Figure 6 Database Settings Tab**

Figure 7 Code Settings Tab

3. You can see the descriptions of each of these fields in the *Complete Guide* tutorial. For now, simply hover over the *Question Mark* image ⊙ if you need information from the respective fields.

4. Click the "*Generate Code for All Tables*" button. AspxFormsGen 4.0 will start generating objects. See Figure 8.
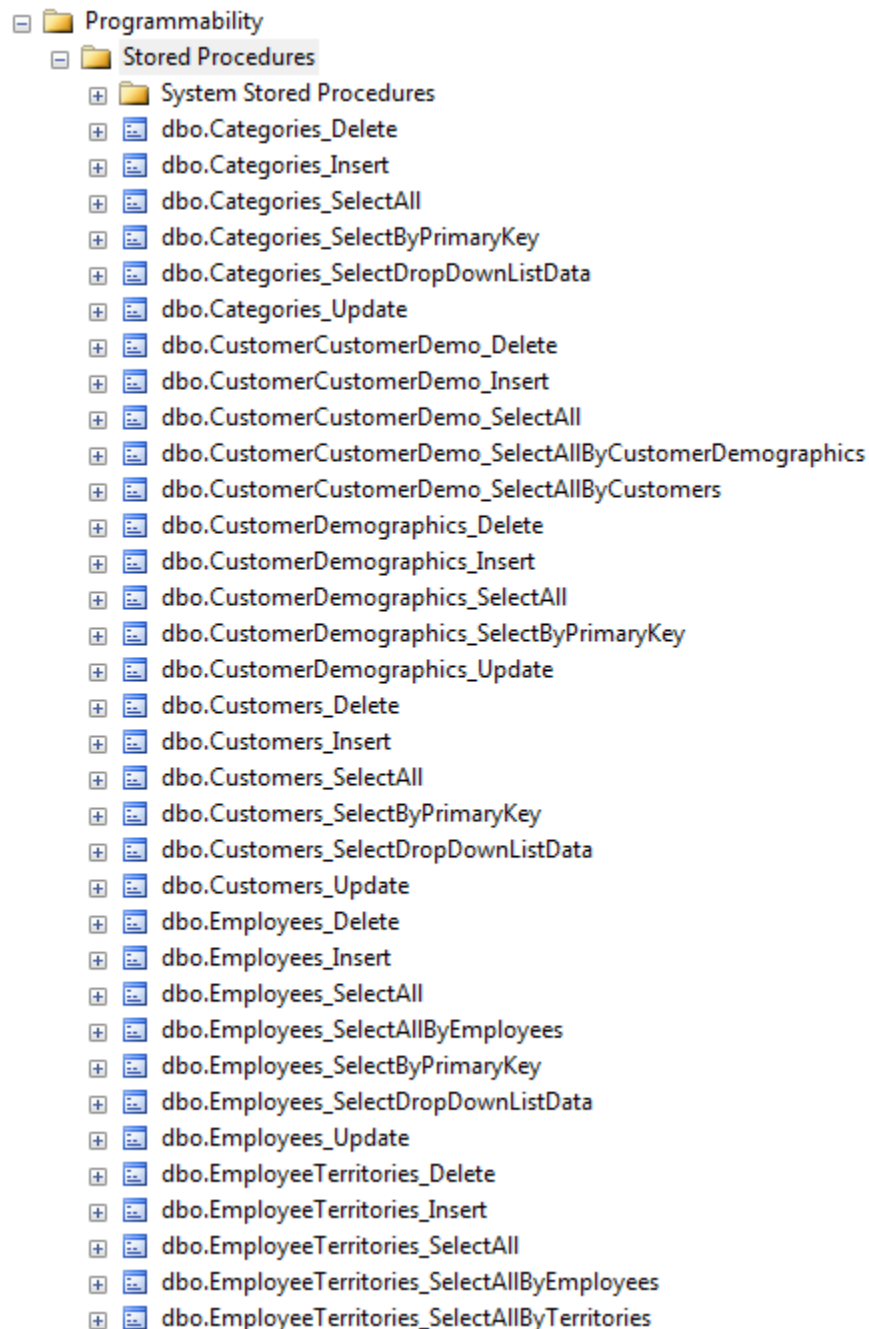

Figure 8 Generating Code

5. Wait for a few seconds. When AspxFormsGen 4.0 is done generating objects, a message pops up. See Figure 9.

Figure 9 Done Generating Objects

6. Click the *OK* button to close the message. Then close AspxFormsGen 4.0 by clicking the *Close* button.

7. We're not going to go into any details on how to operate MS SQL Server Management Studio, but if you want to see the generated Stored Procedures, fire up MS SQL Server Management Studio, versions 2000, 2005, or 2008, then click on the *Stored Procedures* node of your respective database, for this example the *Northwind* database, you will see a list of generated stored procedures. See Figure 10.

```
└ 📁 Programmability
   └ 📁 Stored Procedures
      ⊞ 📁 System Stored Procedures
      ⊞ 🗐 dbo.Categories_Delete
      ⊞ 🗐 dbo.Categories_Insert
      ⊞ 🗐 dbo.Categories_SelectAll
      ⊞ 🗐 dbo.Categories_SelectByPrimaryKey
      ⊞ 🗐 dbo.Categories_SelectDropDownListData
      ⊞ 🗐 dbo.Categories_Update
      ⊞ 🗐 dbo.CustomerCustomerDemo_Delete
      ⊞ 🗐 dbo.CustomerCustomerDemo_Insert
      ⊞ 🗐 dbo.CustomerCustomerDemo_SelectAll
      ⊞ 🗐 dbo.CustomerCustomerDemo_SelectAllByCustomerDemographics
      ⊞ 🗐 dbo.CustomerCustomerDemo_SelectAllByCustomers
      ⊞ 🗐 dbo.CustomerDemographics_Delete
      ⊞ 🗐 dbo.CustomerDemographics_Insert
      ⊞ 🗐 dbo.CustomerDemographics_SelectAll
      ⊞ 🗐 dbo.CustomerDemographics_SelectByPrimaryKey
      ⊞ 🗐 dbo.CustomerDemographics_Update
      ⊞ 🗐 dbo.Customers_Delete
      ⊞ 🗐 dbo.Customers_Insert
      ⊞ 🗐 dbo.Customers_SelectAll
      ⊞ 🗐 dbo.Customers_SelectByPrimaryKey
      ⊞ 🗐 dbo.Customers_SelectDropDownListData
      ⊞ 🗐 dbo.Customers_Update
      ⊞ 🗐 dbo.Employees_Delete
      ⊞ 🗐 dbo.Employees_Insert
      ⊞ 🗐 dbo.Employees_SelectAll
      ⊞ 🗐 dbo.Employees_SelectAllByEmployees
      ⊞ 🗐 dbo.Employees_SelectByPrimaryKey
      ⊞ 🗐 dbo.Employees_SelectDropDownListData
      ⊞ 🗐 dbo.Employees_Update
      ⊞ 🗐 dbo.EmployeeTerritories_Delete
      ⊞ 🗐 dbo.EmployeeTerritories_Insert
      ⊞ 🗐 dbo.EmployeeTerritories_SelectAll
      ⊞ 🗐 dbo.EmployeeTerritories_SelectAllByEmployees
      ⊞ 🗐 dbo.EmployeeTerritories_SelectAllByTerritories
```

Figure 10 Generated Stored Procedures

8. To view the generated web site, simply go to the directory you specified from the *Code Settings* tab of AspxFormsGen 4.0. You will see a list like the one shown in Figure 11.
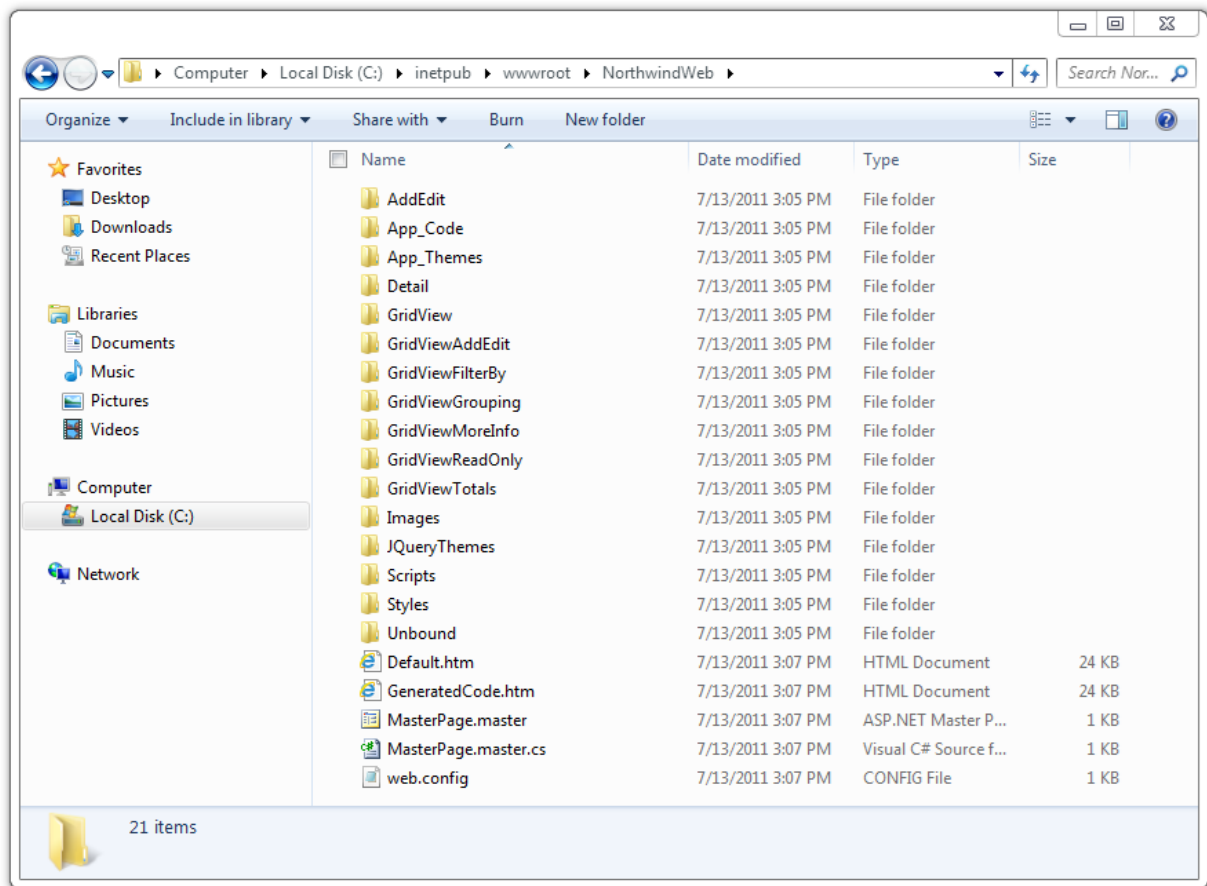
Figure 11 Generated Web Site

9. To see the generated web site in action, fire up Visual Studio 2010. From the *File* menu click *Open Web Site* as shown in Figure 12.
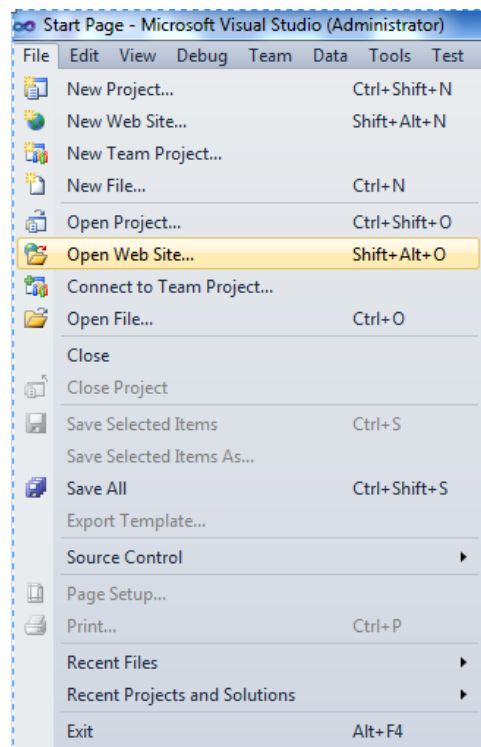

Figure 12 Visual Studio 2010, Open Web Site

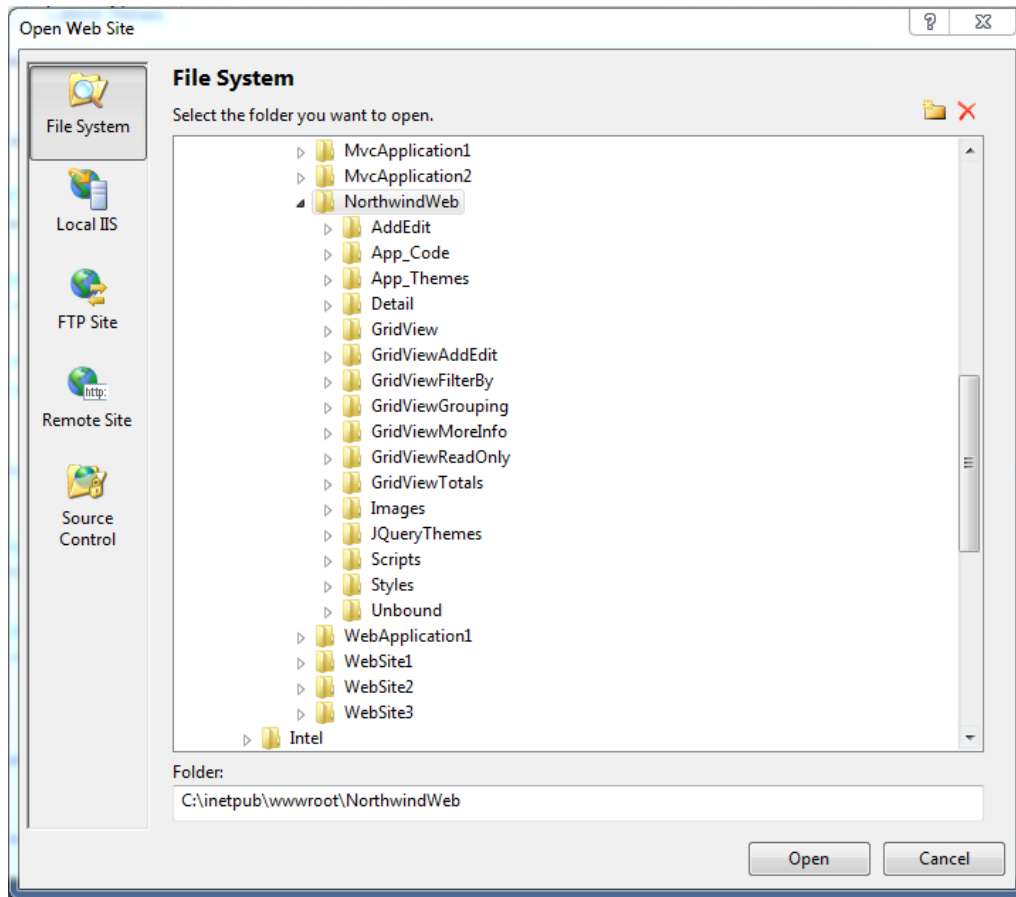10. Point to the web site directory, and then click *Open*. See Figure 13.

Figure 13 Generated Web Site

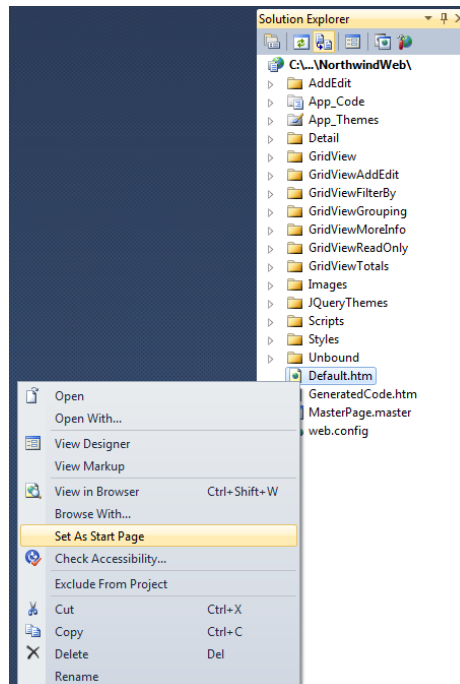11. From the *Solution Explorer*, right-click on the *Default.htm* and then click *Set As Start Page*. See Figure 14.

Figure 14 Set As Start Page

12. Run Visual Studio by pressing F5. You will see a list of all the generated ASP.NET 4.0 web forms. See Figure 15. You can click any link to preview the functionality of each of the generated web form. However, we're not going to discuss this in this tutorial. Please see the *Complete Guide* tutorial for more information.

Thank You for using AspxFormsGen 4.0 Professional+. Listed below are the ASP.Net 4.0 Web Forms generated by AspxFormsGen 4.0 Professional+.

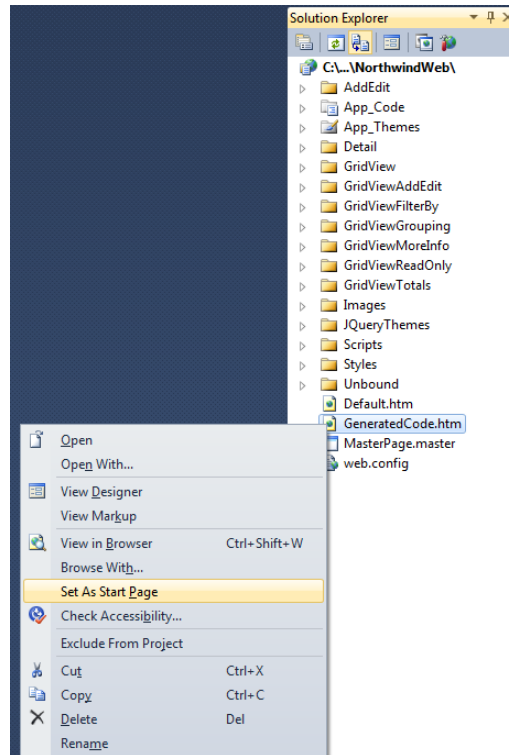| GridView with Add, Edit Redirect, & Delete	Features
• GridView/Categories_Web.aspx • GridView/CustomerDemographics_Web.aspx • GridView/Customers_Web.aspx • GridView/Employees_Web.aspx • GridView/OrderDetails_Web.aspx • GridView/Orders_Web.aspx • GridView/Products_Web.aspx • GridView/Region_Web.aspx • GridView/Shippers_Web.aspx • GridView/Suppliers_Web.aspx • GridView/Territories_Web.aspx	• **Can be used in the administration part of your website** • Contains a GridView Server Control that has CRUD (Create, Retrieve, Update, Delete) funtionality. • Adding a new record redirects to another page • Updating and existing record redirects to another page • Delete funtionality uses a JQuery UI Pop-up for delete confirmation • A link to a read-only Web Form is also provided for all Foreign Key columns (for details on the foreign key) • GridView uses a Sort Direction Image in the header • GridView uses Numeric Paging in the footer • One ASP.NET 4.0 Web Form is generated per table
GridView with Add, Edit, & Delete (Functionality on the Same Page)	**Features**
• GridViewAddEdit/Categories_Web.aspx • GridViewAddEdit/CustomerDemographics_Web.aspx • GridViewAddEdit/Customers_Web.aspx • GridViewAddEdit/Employees_Web.aspx • GridViewAddEdit/OrderDetails_Web.aspx • GridViewAddEdit/Orders_Web.aspx • GridViewAddEdit/Products_Web.aspx • GridViewAddEdit/Region_Web.aspx • GridViewAddEdit/Shippers_Web.aspx • GridViewAddEdit/Suppliers_Web.aspx • GridViewAddEdit/Territories_Web.aspx	• **Can be used in the administration part of your website** • Contains a GridView Server Control that has CRUD (Create, Retrieve, Update, Delete) funtionality. • Add a new record on the same page with JQuery animation • Update an existing record on the same page with JQuery animation • Delete funtionality uses a JQuery UI Pop-up for delete confirmation • A JQuery Tooltip pop-up link is provided for all Foreign Key columns (for details on the foreign key) • GridView uses a Sort Direction Image in the header • GridView uses Numeric Paging in the footer • One ASP.NET 4.0 Web Form is generated per table

Figure 15 List of Generated Web Forms

13. When you click any of the links you are redirected to that specific web page as seen in Figure 16. Play around to see the functionality of each web page.

Product ID	Product Name ↑	Supplier ID	Category ID	Quantity Per Unit	Unit Price	Units In Stock	Units On Order	Reorder Level	Discontinued		
17	Alice Mutton	7	6	20 - 1 kg tins	$39.00	0	0	0	☑	✎	🗑
3	Aniseed Syrup	1	2	12 - 550 ml bottles	$10.00	13	70	25	☐	✎	🗑
40	Boston Crab Meat	19	8	24 - 4 oz tins	$18.40	123	0	30	☐	✎	🗑
60	Camembert Pierrot	28	4	15 - 300 g rounds	$34.00	19	0	0	☐	✎	🗑
18	Carnarvon Tigers	7			$62.50	42	0	0	☐	✎	🗑
1	Chai	1			$18.00	39	0	10	☐	✎	🗑
2	Chang	1			$19.00	17	40	26	☐	✎	🗑
39	Chartreuse verte	18			$18.00	69	0	5	☐	✎	🗑
4	Chef Anton's Cajun Seasoning	2			$22.00	53	0	0	☐	✎	🗑
5	Chef Anton's Gumbo Mix	2			$21.35	0	0	0	☑	✎	🗑
48	Chocolade	22			$12.75	15	70	25	☐	✎	🗑
38	Côte de Blaye	18			$263.50	17	0	15	☐	✎	🗑
58	Escargots de Bourgogne	27	8	24 pieces	$13.25	62	0	20	☐	✎	🗑
52	Filo Mix	24	5	16 - 2 kg boxes	$7.00	38	0	25	☐	✎	🗑
71	Flotemysost	15	4	10 - 500 g pkgs.	$21.50	26	0	0	☐	✎	🗑
33	Geitost	15	4	500 g	$2.50	112	0	20	☐	✎	🗑

Supplier ID: 28
Company Name: Gai pâturage
Contact Name: Eliane Noz
Contact Title: Sales Representative
Address: Bat. B 3, rue des Alpes
City: Annecy
Region:
Postal Code: 74000
Country: France
Phone: 38.76.98.06
Fax: 38.76.98.58
Home Page:

| 1 | 2 | 3 | 4 | 5 |

Figure 16 Sample Generated Web Page (GridView w/ Add, Edit, Delete for Products Table)

14. Close the web page and go back to Visual Studio 2010. From the *Solution Explorer*, right-click on the *GeneratedCode.htm* and then click *Set As Start Page*. See Figure 17.

Figure 17 Set As Start Page

15. Run Visual Studio by pressing F5. You will see a list of all the generated middle-tier classes, data-tier classes, and stored procedures (or dynamic SQL classes). See Figure 18. You can hover over each of the link to see where each file is located. However, we're not going to discuss this in this tutorial. Please see the *Complete Guide* tutorial for more information.

Business Object Classes	Features
• Categories.cs • CustomerDemographics.cs • Customers.cs • Employees.cs • OrderDetails.cs • Orders.cs • Products.cs • Region.cs • Shippers.cs • Suppliers.cs • Territories.cs	• **Note: The only code you call from your application** • **Used as the gateway middle layer object the client calls** • Most CRUD calls can be made in one (1) line of code • Inherits from the respective BusinessObjectBase class • You can add additional code here (it will not be rewritten by the generator) • One Class is generated per table • Located in the \BusinessObject\ folder
Business Object Base Classes	**Features**
• CategoriesBase.cs • CustomerDemographicsBase.cs • CustomersBase.cs • EmployeesBase.cs • OrderDetailsBase.cs • OrdersBase.cs • ProductsBase.cs • RegionBase.cs • ShippersBase.cs • SuppliersBase.cs • TerritoriesBase.cs	• **Used as the base class to the Business Object class** • **Do not add or edit code here** • Encapsulates calls to the data layer • Contains table fields as properties • One Class is generated per table • Located in the \BusinessObjectBase\ folder

Figure 18 List of Middle-Tier, Data-Tier, and Stored Procedures (or Dynamic SQL)

16. Close the web page and go back to Visual Studio 2010. The generated middle-tier and data-tier classes can be found under the *App_Code* folder. Please see Figure 18.
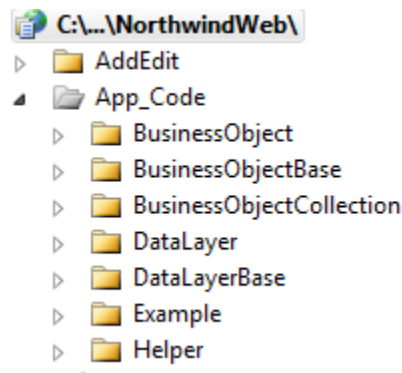

Figure 18 Middle-Tier and Data-Tier Classes Under App_Code Folder

Note: Some features shown here are not available in the Express Edition.

End of tutorial.