

# Activating AspFormsGen MVC 5

**Note:** You need internet connection to activate AspFormsGen MVC 5. You can install one (1) license of AspFormsGen MVC 5 for up to 2 computers you own or use for work. You need one (1) license per developer and the license cannot be shared with other developers.

The first time you open AspFormsGen MVC 5 Professional Plus edition you will be presented with an activation form right after the Splash screen as shown in Figure 1. You will not be shown the activation form when using the Express edition.

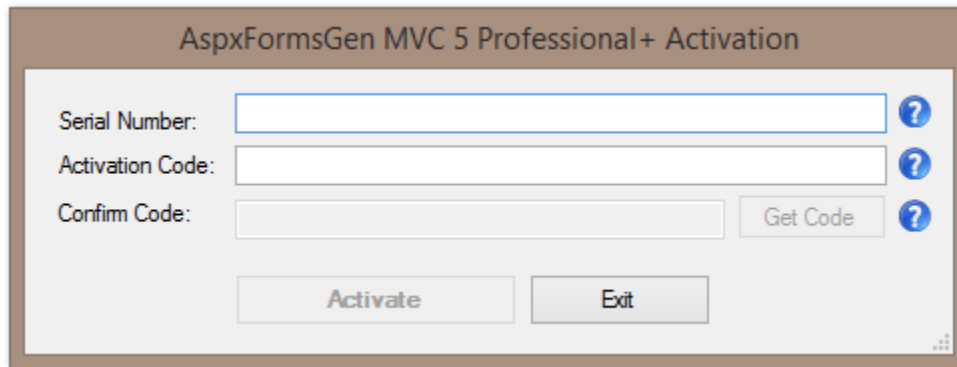
The image shows the 'AspFormsGen MVC 5 Professional+ Activation' dialog box. It features three text input fields: 'Serial Number', 'Activation Code', and 'Confirm Code'. Each field has a blue question mark icon to its right. Below the 'Confirm Code' field is a 'Get Code' button. At the bottom of the dialog are two buttons: 'Activate' and 'Exit'.

Figure 1 Activation Form

Enter the *Serial Number* and *Activation Code* 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 *Get Code* button to get the *Confirm Code*. See Figure 2.

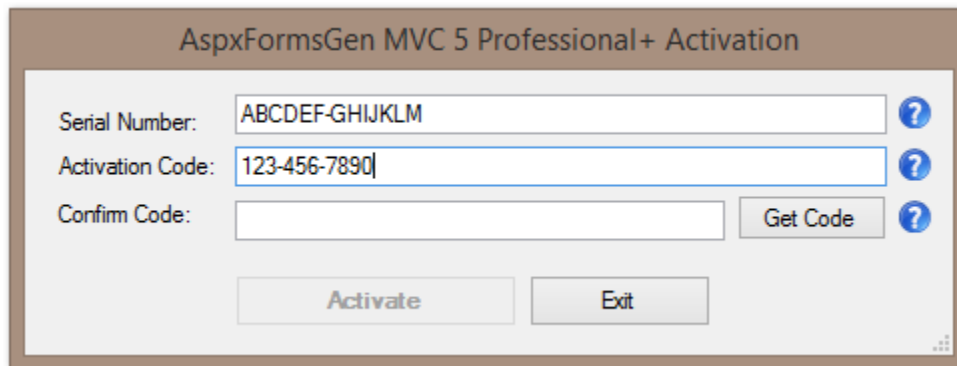
The image shows the same 'AspFormsGen MVC 5 Professional+ Activation' dialog box as in Figure 1, but with data entered. The 'Serial Number' field contains 'ABCDEF-GHIJKLM', the 'Activation Code' field contains '123-456-7890', and the 'Confirm Code' field is empty. The 'Get Code' button is now enabled and highlighted.

Figure 2

The *Confirm Code* will be sent via email to the original purchaser's email address we have on file. When the *Confirm Code* is sent, a pop up message will show as seen in Figure 3. Click the OK button.

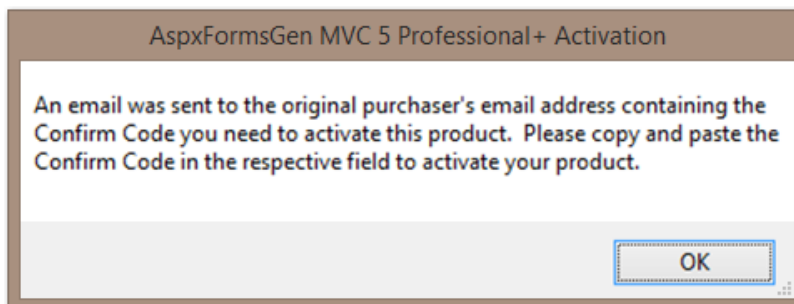
The image shows a pop-up message box with a white background and a brown border. The text inside reads: 'An email was sent to the original purchaser's email address containing the Confirm Code you need to activate this product. Please copy and paste the Confirm Code in the respective field to activate your product.' At the bottom right of the message box is an 'OK' button.

Figure 3 Confirm Code Sent

Enter the *Confirm Code* in the respective box and then click the *Activate* button.

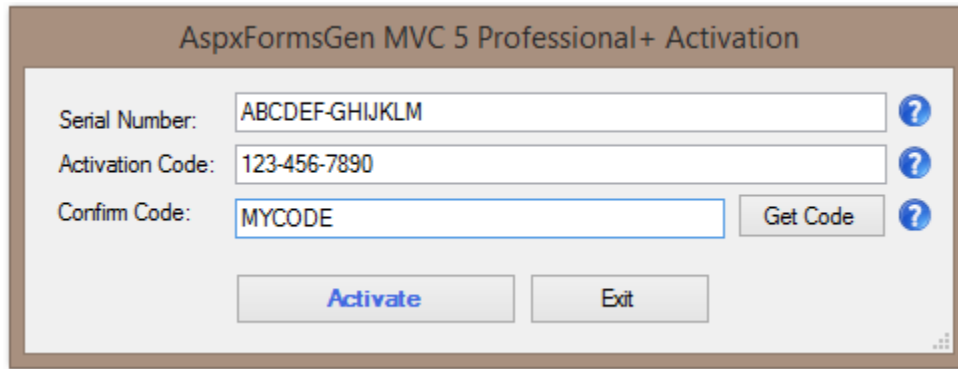


Figure 4 Enter Confirm Code

If you enter an invalid *Serial Number*, *Activation Code*, or *Confirm Code* a warning pops up. When this happens, click the *Ok* button (See Figure 5), then re-enter the *Serial Number*, *Activation Code*, and *Confirm Code* Again.

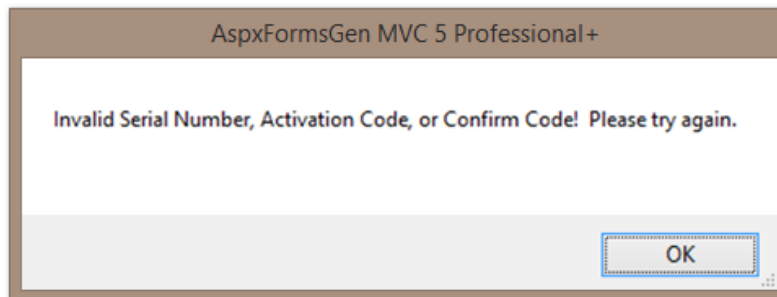


Figure 5 Invalid Serial Number, Activation Code, or Confirm Code

If you enter a valid *Serial Number*, *Activation Code* and *Confirm Code*, you will be presented with the main window of the application, see Figure 6. You will only see the activation form once. After that, you will always go straight to the main window.

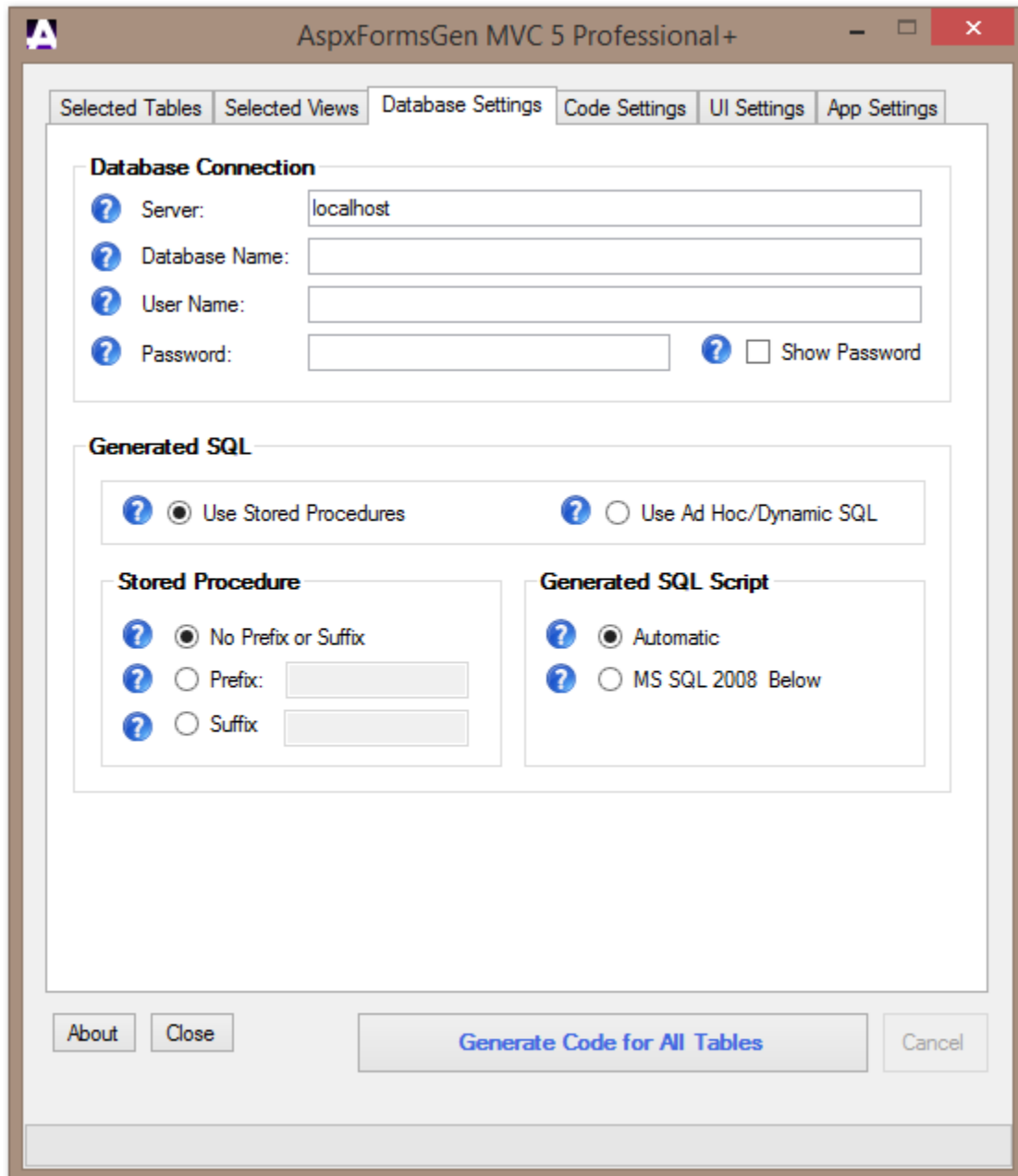


Figure 6 Main Window

## A Simple Interface

To keep AspxFormsGen MVC 5 simple, there's only one main interface as shown in Figure 5. 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.
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.

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, selected tables, or selected views. This is also where you set the web application name, the root directory where you want the web application 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 MVC Views here. You can choose themes for the JQuery UI controls. You can also select which MVC Views to generate and the MVC View's filename prefix to use for each MVC Views.
6. **App Settings:** These are application settings. Almost all generated code/MVC Views are overwritten every time you use AspxFormsGen MVC 5. However, you can choose not to overwrite some key files from here. You can also reset all settings to its original default from here.

That seems like a lot of features, you're probably asking "where's the One Click feature?" Since AspxFormsGen MVC 5 remembers the last settings you used such as, e.g. server, database name, directory, namespace, language etc., the next time you open AspxFormsGen MVC 5, you can just click the Generate... button, that simple.

## A Quick Tour

Let's learn how to generate ASP.NET 4.5 Web Forms, middle-tier, and data-tier codes using AspxFormsGen MVC 5. **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 7.

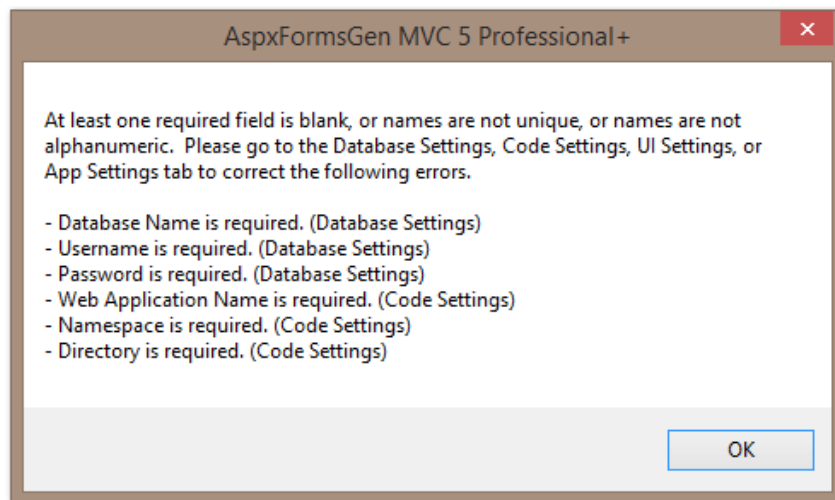


Figure 7 Error Message

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 8 and 9.

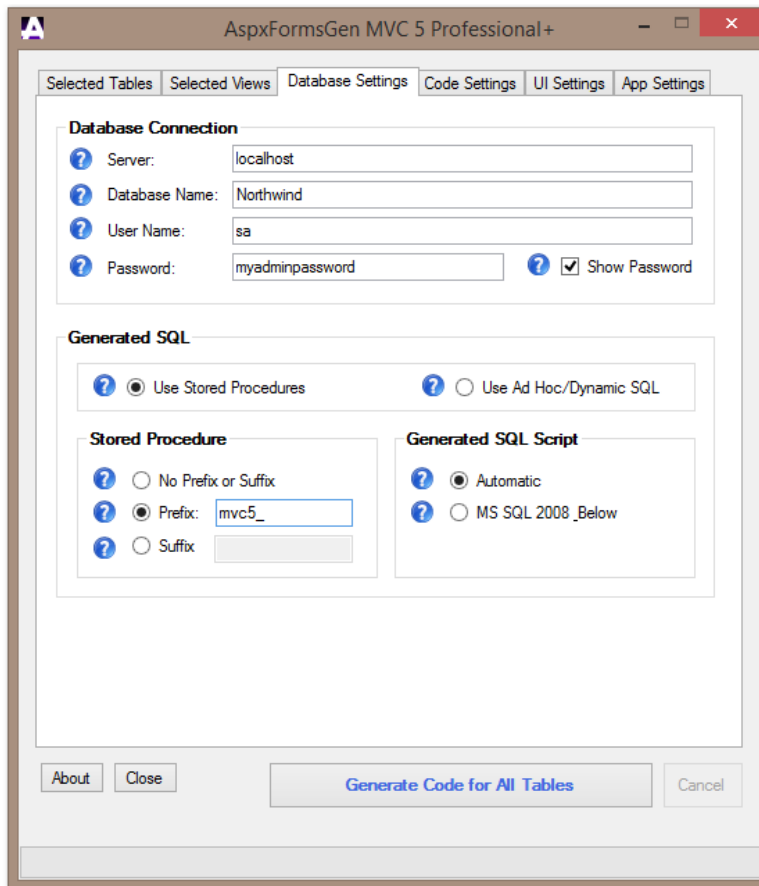


Figure 8 Database Settings Tab

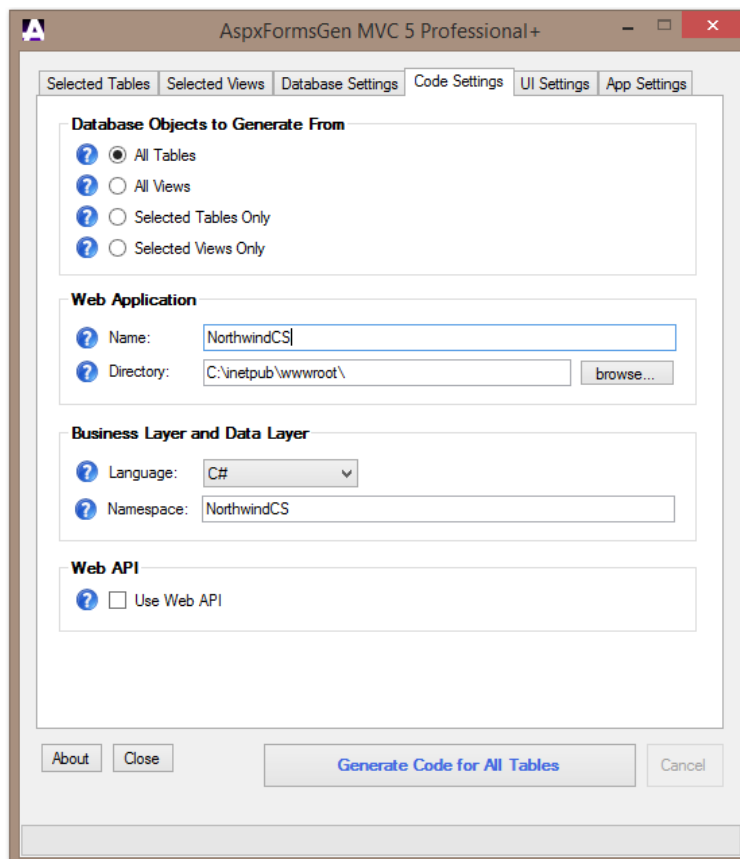

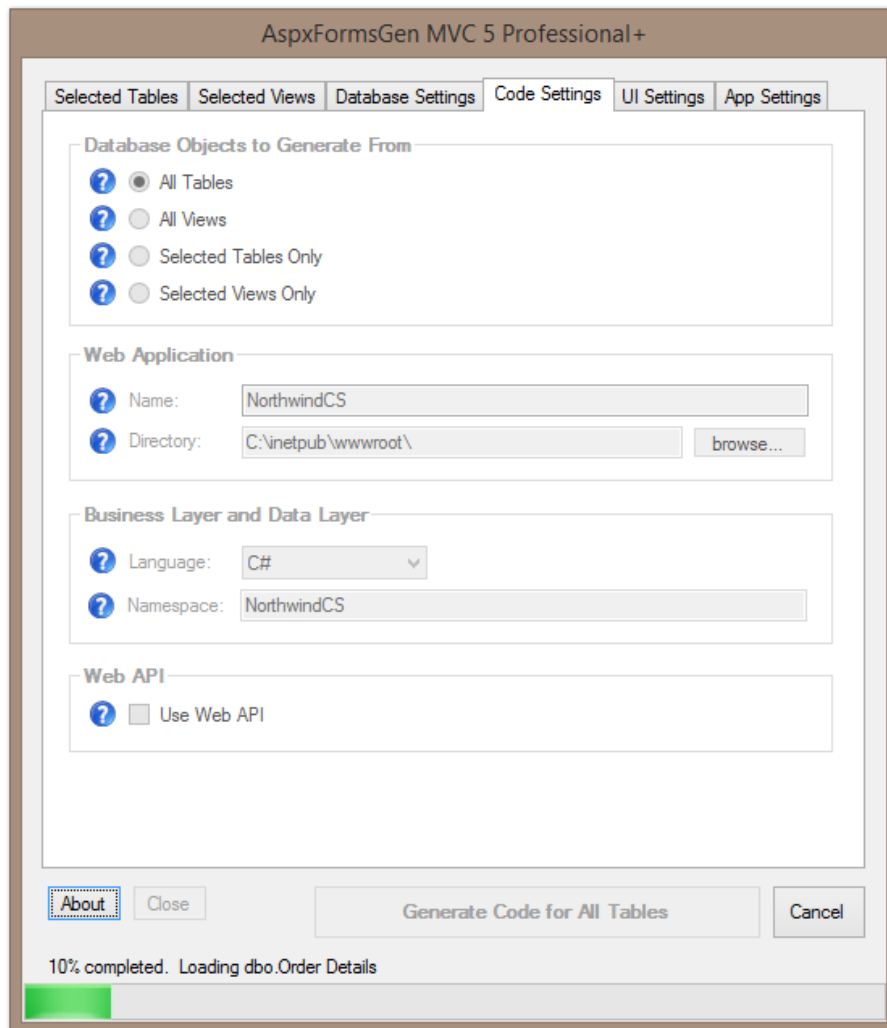


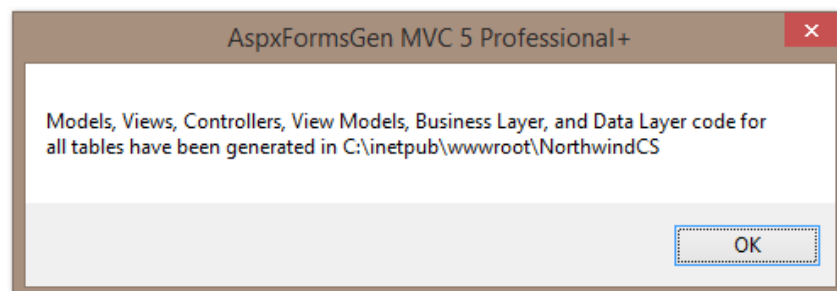
Figure 9 Code Settings Tab

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 MVC 5 will start generating objects. See Figure 10.



**Figure 10 Generating Code**

5. Wait for a few seconds. When AspxFormsGen MVC 5 is done generating objects, a message pops up. See Figure 11.



**Figure 11 Done Generating Objects**

6. Click the *OK* button to close the message. Then close AspxFormsGen MVC 5 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, 2008, 2012, or 2014 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 12.

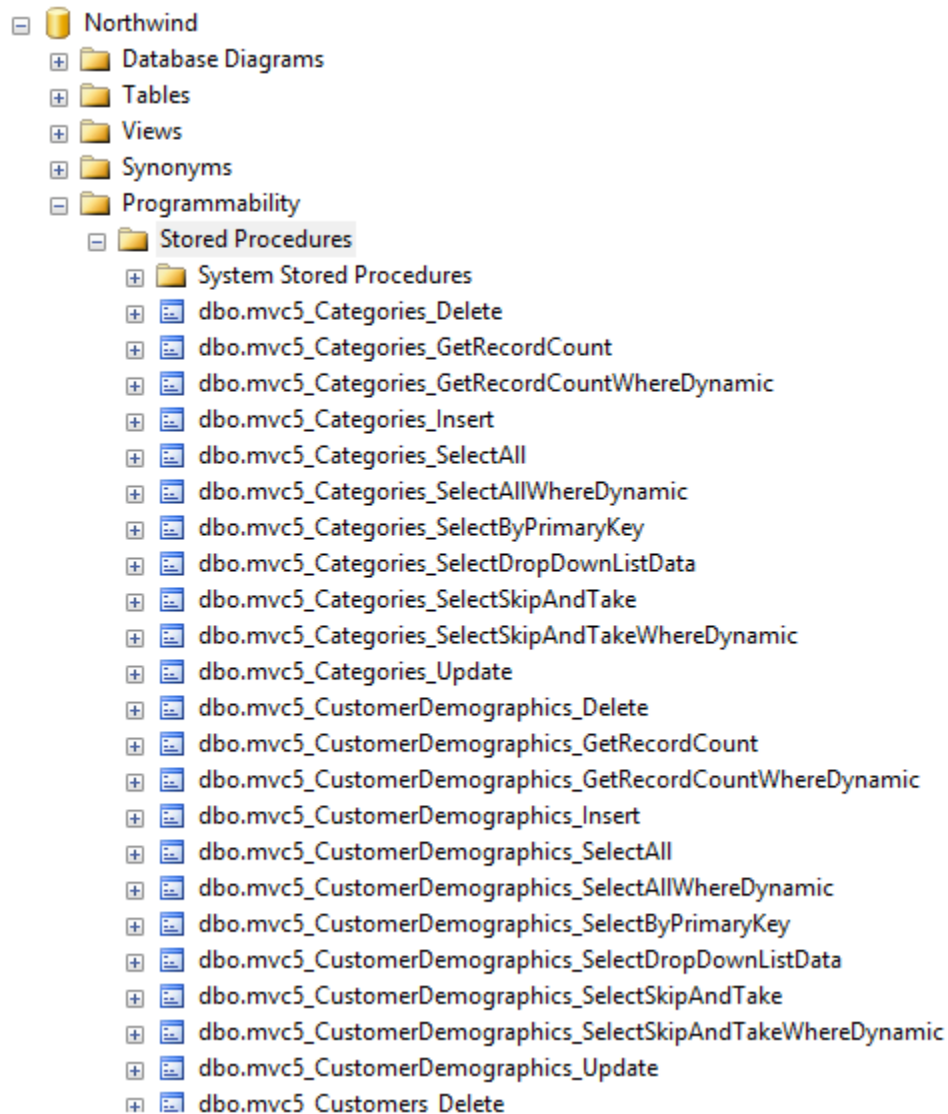


Figure 12 Generated Stored Procedures

8. To view the generated web application, simply go to the directory you specified from the *Code Settings* tab of AspxFormsGen MVC 5. You will see a list like the one shown in Figure 13.

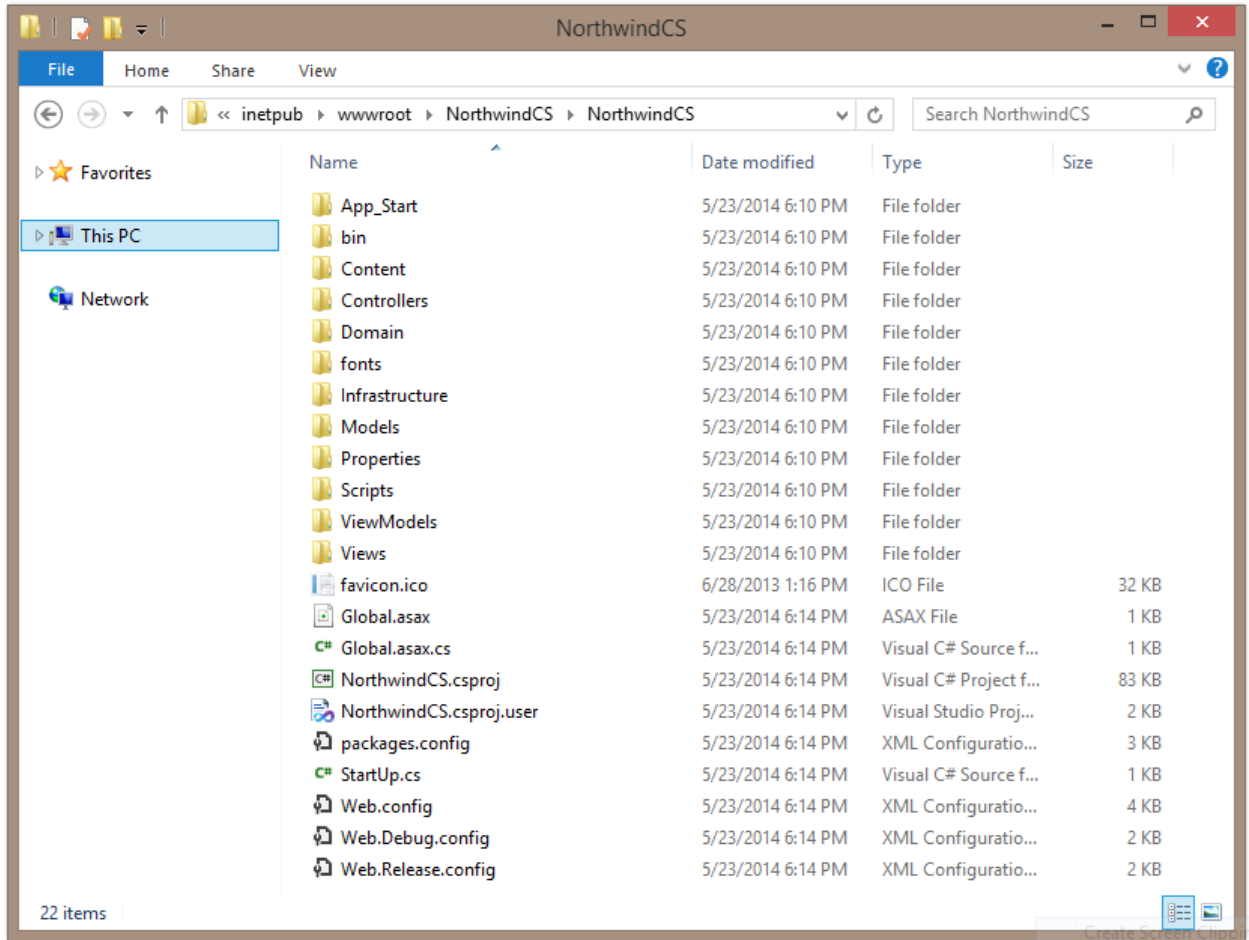


Figure 13 Generated MVC Web Application



9. To see the generated web site in action, fire up Visual Studio 2013. From the *File* menu click *Open Project* as shown in Figure 14.

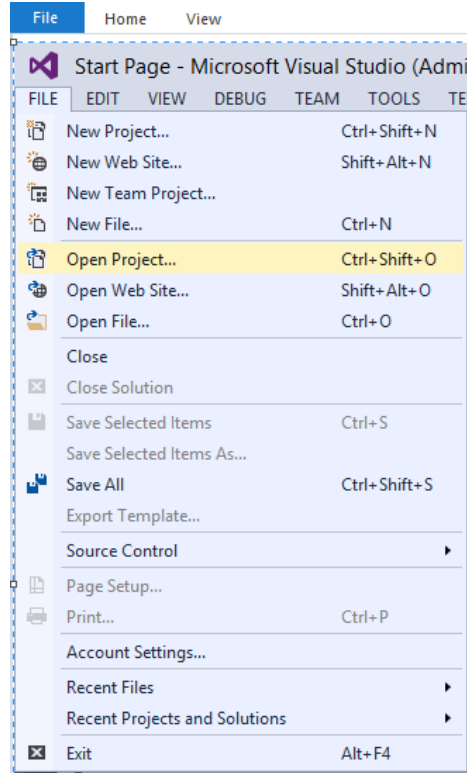


Figure 14 Visual Studio 2013, Open Project

10. Go to the web app directory, and then click the Solution File (.sln). See Figure 15.

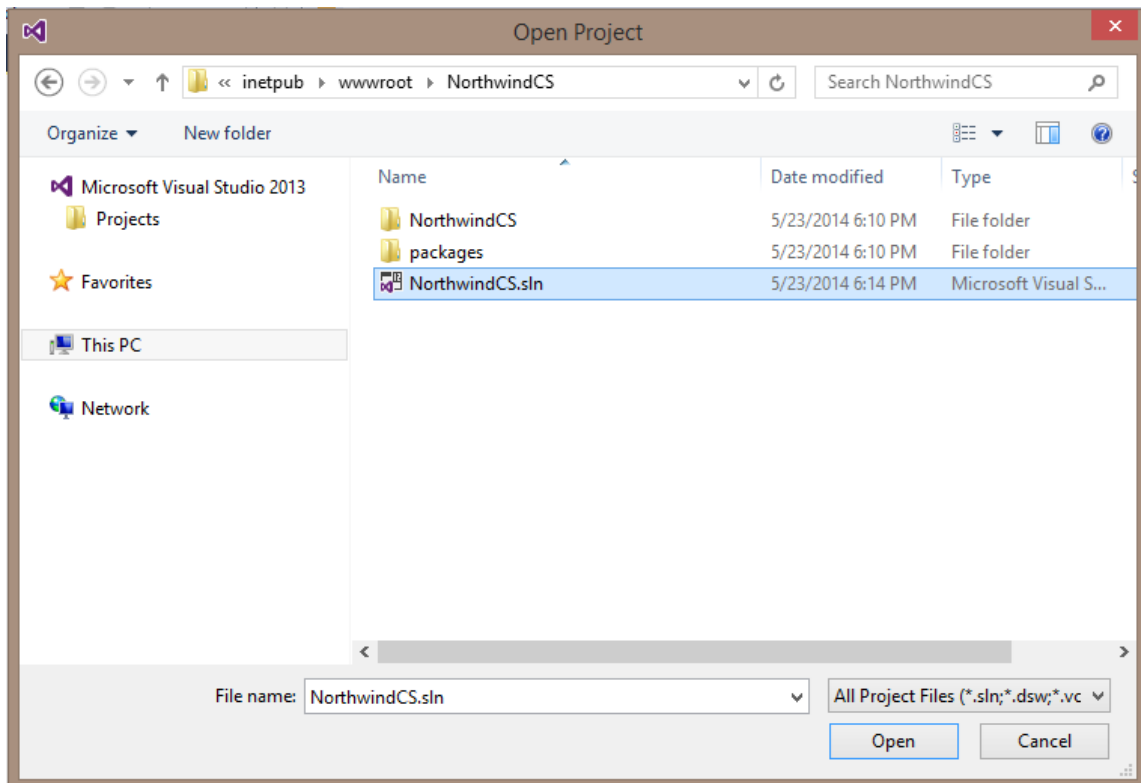


Figure 15 Generated Web Application

- Run Visual Studio by pressing F5. You will see a list of all the generated ASP.NET MVC 5 Objects (models, views, controllers, view models, etc). See Figure 16. 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.

website name here Home

Thank You for using AspFormsGen MVC 5 Professional+. Listed below are the ASP.NET MVC 5 Models, Views, Controllers, and View Models generated by AspFormsGen MVC 5 Professional+.

Please [click here](#) to see the list of generated Middle-Tier (Business Objects) and Data-Tier code, and Stored Procedures or Dynamic SQL.

IEWS - Views Folder	MODELS - Models Folder	VIEW MODELS - ViewModels Folder	CONTROLLERS - Controllers Folder
<p><b>LIST WITH ADD, EDIT REDIRECT, &amp; DELETE VIEWS</b></p> <p><a href="#">Categories/ListCrudRedirect</a>  <a href="#">CustomerDemographics/ListCrudRedirect</a>  <a href="#">Customers/ListCrudRedirect</a>  <a href="#">Employees/ListCrudRedirect</a>  <a href="#">OrderDetails/ListCrudRedirect</a>  <a href="#">Orders/ListCrudRedirect</a>  <a href="#">Products/ListCrudRedirect</a>  <a href="#">Region/ListCrudRedirect</a>  <a href="#">Shippers/ListCrudRedirect</a>  <a href="#">Suppliers/ListCrudRedirect</a>  <a href="#">Territories/ListCrudRedirect</a></p> <p><b>ADD NEW RECORD VIEWS</b></p> <p><a href="#">Categories/Add</a>  <a href="#">CustomerDemographics/Add</a>  <a href="#">Customers/Add</a>  <a href="#">Employees/Add</a>  <a href="#">OrderDetails/Add</a>  <a href="#">Orders/Add</a>  <a href="#">Products/Add</a>  <a href="#">Region/Add</a>  <a href="#">Shippers/Add</a>  <a href="#">Suppliers/Add</a>  <a href="#">Territories/Add</a></p>	<p>CategoriesModel.cs  CustomerDemographicsModel.cs  CustomersModel.cs  EmployeesModel.cs  OrderDetailsModel.cs  OrdersModel.cs  ProductsModel.cs  RegionModel.cs  ShippersModel.cs  SuppliersModel.cs  TerritoriesModel.cs</p> <p><b>MODELS (Base Class) - Models\Base Folder</b></p> <p>CategoriesModelBase.cs  CustomerDemographicsModelBase.cs  CustomersModelBase.cs  EmployeesModelBase.cs  OrderDetailsModelBase.cs  OrdersModelBase.cs  ProductsModelBase.cs  RegionModelBase.cs  ShippersModelBase.cs  SuppliersModelBase.cs  TerritoriesModelBase.cs</p>	<p>CategoriesViewModel.cs  CustomerDemographicsViewModel.cs  CustomersViewModel.cs  EmployeesViewModel.cs  OrderDetailsViewModel.cs  OrdersViewModel.cs  ProductsViewModel.cs  RegionViewModel.cs  ShippersViewModel.cs  SuppliersViewModel.cs  TerritoriesViewModel.cs</p> <p><b>VIEW MODELS (Base Class) - ViewModels\Base Folder</b></p> <p>CategoriesViewModelBase.cs  CustomerDemographicsViewModelBase.cs  CustomersViewModelBase.cs  EmployeesViewModelBase.cs  OrderDetailsViewModelBase.cs  OrdersViewModelBase.cs  ProductsViewModelBase.cs  RegionViewModelBase.cs  ShippersViewModelBase.cs  SuppliersViewModelBase.cs  TerritoriesViewModelBase.cs</p>	<p>CategoriesController.cs  CustomerDemographicsController.cs  CustomersController.cs  EmployeesController.cs  OrderDetailsController.cs  OrdersController.cs  ProductsController.cs  RegionController.cs  ShippersController.cs  SuppliersController.cs  TerritoriesController.cs</p> <p><b>CONTROLLERS (Base Class) - Controllers\Base Folder</b></p> <p>CategoriesControllerBase.cs  CustomerDemographicsControllerBase.cs  CustomersControllerBase.cs  EmployeesControllerBase.cs  OrderDetailsControllerBase.cs  OrdersControllerBase.cs  ProductsControllerBase.cs  RegionControllerBase.cs  ShippersControllerBase.cs  SuppliersControllerBase.cs  TerritoriesControllerBase.cs</p>

Figure 16 List of Generated MVC Objects

- When you click any of the links you are redirected to that specific page as seen in Figure 17. Play around to see the functionality of each web page.

website name here Home

### List of Products

[Add New Products](#)

Product ID	Product Name	Supplier ID	Category ID	Quantity Per Unit	Unit Price	Units In Stock	Units On Order	Reorder Level	Discontinued		
1	Chai	1	1	10 boxes x 20 bags	\$18.02	39	0	12	<input checked="" type="checkbox"/>		
2	Chang	1	1	24 - 12 oz bottles	\$15.00	17	40	25	<input type="checkbox"/>		
3	Aniseed Syrup	1	2	12 - 550 ml bottles	\$10.00	13	70	25	<input type="checkbox"/>		
4	Chef Anton's Cajun	2	2	48 - 6 oz jars	\$21.00	3	4	5	<input type="checkbox"/>		
5	Chef Anton's Gum	2	2	36 boxes	\$21.35	0	0	0	<input type="checkbox"/>		
6	Grandma's Boysenbe	3	2	12 - 8 oz jars	\$25.00	120	0	25	<input type="checkbox"/>		
7	Uncle Bob's Organic	3	7	12 - 1 lb pkgs.	\$30.00	15	0	10	<input type="checkbox"/>		
8	Northwoods Cranber	3	2	12 - 12 oz jars	\$40.00	6	0	0	<input type="checkbox"/>		
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	\$97.00	29	0	0	<input checked="" type="checkbox"/>		
10	Ikura	4	8	12 - 200 ml jars	\$31.00	31	0	0	<input type="checkbox"/>		

Page 1 of 15 | 10 | View 1 - 10 of 145

© 2014 - Your Company Name Here Create Screen Clipping (Windows+5)

Figure 17 Sample Generated MVC View

- Go back to the List of Generated MVC Objects page (Figure 16). Click the link where it's directing you to go to the List of Generated middle-tier classes, data-tier code, and stored procedures (or dynamic SQL classes). See Figure 18. We're not going to discuss this in this tutorial. Please see the *Complete Guide* tutorial for more information.

Thank You for using AspFormsGen MVC 5 Professional+. Listed below are the Middle-Tier, Data-Tier, and SQL code generated by AspFormsGen MVC 5 Professional+.

Business Object Classes ( <i>Infrastructure/BusinessObject folder</i> )	Features
<ul style="list-style-type: none"> <li>• Categories.cs</li> <li>• CustomerDemographics.cs</li> <li>• Customers.cs</li> <li>• Employees.cs</li> <li>• OrderDetails.cs</li> <li>• Orders.cs</li> <li>• Products.cs</li> <li>• Region.cs</li> <li>• Shippers.cs</li> <li>• Suppliers.cs</li> <li>• Territories.cs</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Note: The only code you call from your application</b></li> <li>• Used as the gateway middle layer object the client calls</li> <li>• Most CRUD calls can be made in one (1) line of code</li> <li>• Inherits from the respective BusinessObjectBase class</li> <li>• You can add additional code here (it will not be overwritten by the generator)</li> <li>• One Class is generated per table</li> <li>• Located in the \BusinessObject\ folder</li> </ul>
Business Object Base Classes ( <i>Infrastructure/BusinessObjectBase folder</i> )	Features
<ul style="list-style-type: none"> <li>• CategoriesBase.cs</li> <li>• CustomerDemographicsBase.cs</li> <li>• CustomersBase.cs</li> <li>• EmployeesBase.cs</li> <li>• OrderDetailsBase.cs</li> <li>• OrdersBase.cs</li> <li>• ProductsBase.cs</li> <li>• RegionBase.cs</li> <li>• ShippersBase.cs</li> <li>• SuppliersBase.cs</li> <li>• TerritoriesBase.cs</li> </ul>	<ul style="list-style-type: none"> <li>• Used as the base class to the Business Object class</li> <li>• Do not add or edit code here</li> <li>• Contains table fields as properties</li> <li>• Encapsulates calls to the data layer</li> <li>• One Class is generated per table</li> <li>• Located in the \BusinessObjectBase\ folder</li> </ul>
Business Object Collection Classes ( <i>Infrastructure/BusinessObjectCollection folder</i> )	Features
<ul style="list-style-type: none"> <li>• CategoriesCollection.cs</li> <li>• CustomerDemographicsCollection.cs</li> <li>• CustomersCollection.cs</li> </ul>	<ul style="list-style-type: none"> <li>• Used as the Collection of the Business Object Class</li> <li>• Do not add or edit code here</li> <li>• One Class is generated per table</li> </ul>

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

14. Close the web page and go back to Visual Studio 2013. The generated middle-tier and data-tier classes can be found under the *Infrastructure* folder. Please see Figure 19.

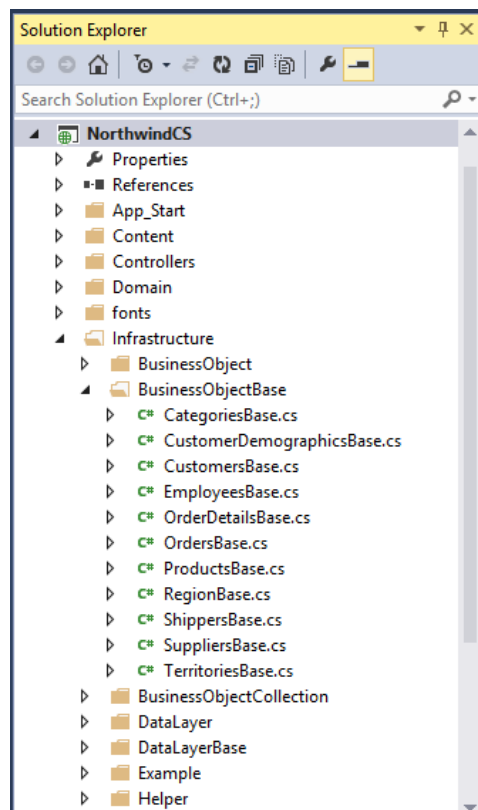


Figure 19 Middle-Tier and Data-Tier Classes Under Infrastructure Folder

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

End of tutorial.