**Dynamic Connection String using Entity Framework 6.\***

**José Lugo García (**[**jlugogar@mdc.edu**](mailto:jlugogar@mdc.edu)**)**

**Created on: 02/06/2017**

**Updated: 03/29/2017**

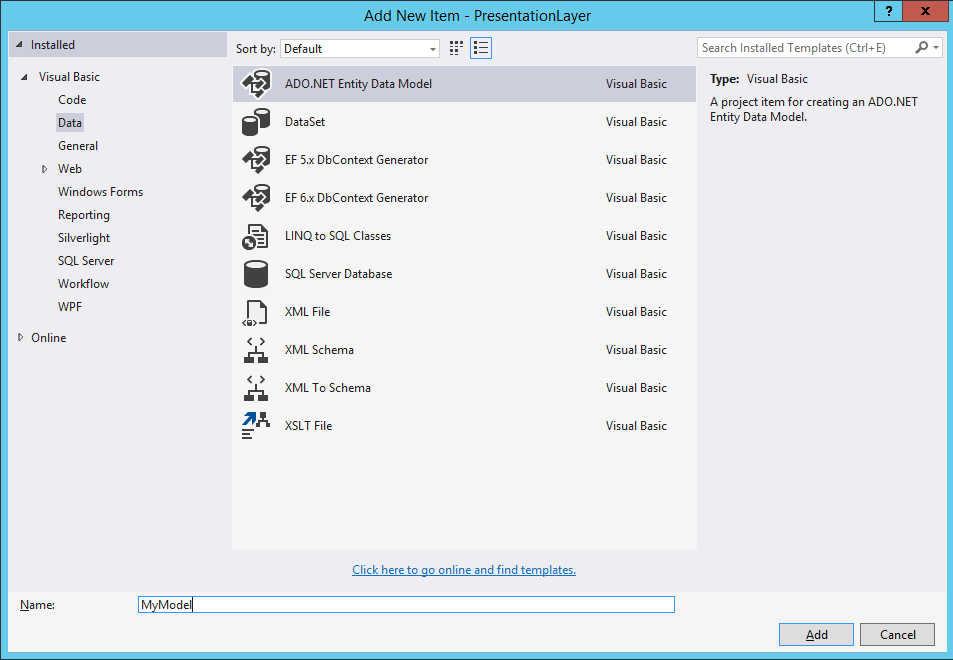
**Description:** Sample solution using dynamic connection string with EntityFramework 6.\*

**IDE:** Visual Studio 2015.

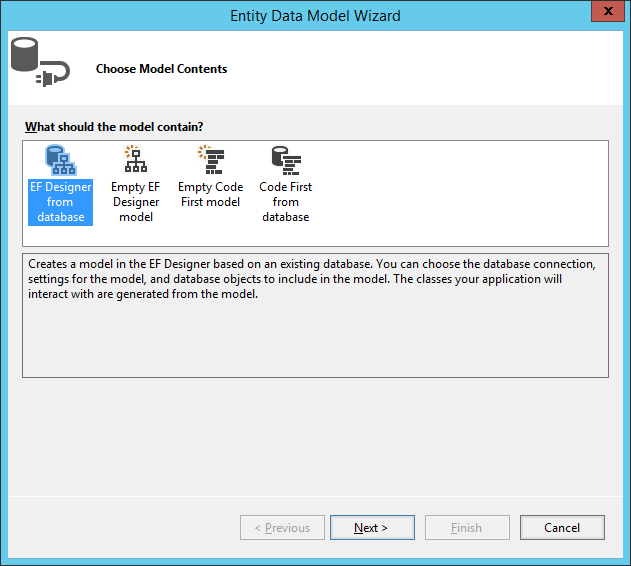
1. Restore a backup of WideWorldImporters-Full.bak to your local SQL Server machine. See:

<https://github.com/Microsoft/sql-server-samples/releases/tag/wide-world-importers-v1.0>

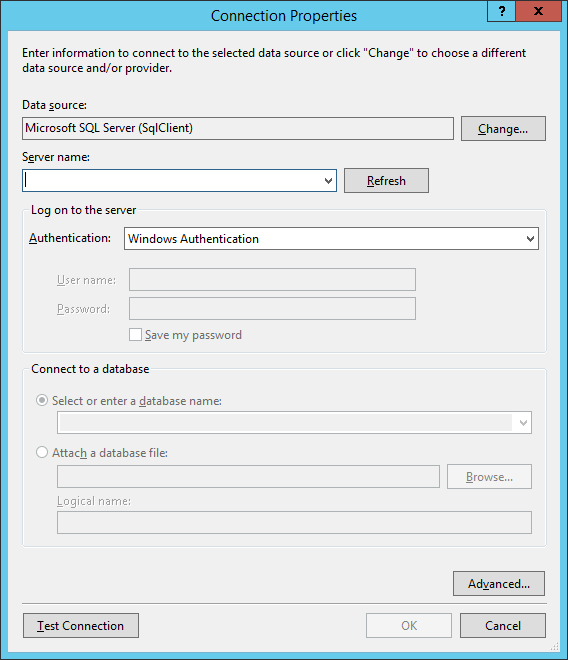
1. Create a new ASP.NET web application project.
2. Right click on top of the created project, Add-> New Item.
3. Select “Data” from the left corner of the dialog.
4. Click on “ADO.NET Entity Data Model”:



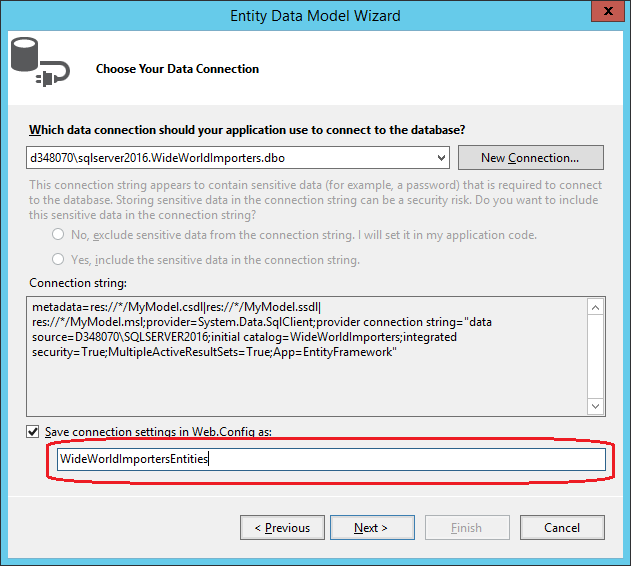
1. Specify a name and click “Add”. This will prompt the next window:



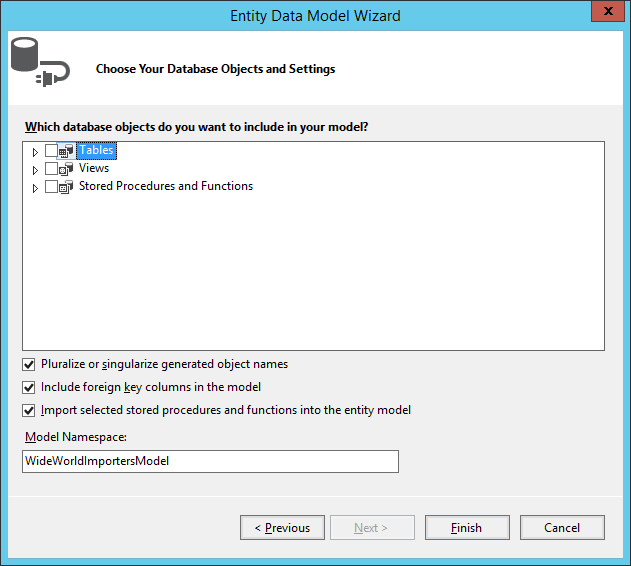
1. As the model we want to map already exists in database, select the first option “EF Designer from database”, and click “Next”. A screen prompting for establishing a new connection pops up. Click on New Connection, complete the parameters as you usually do, and establish a connection to the *WideWorldImporters* database:



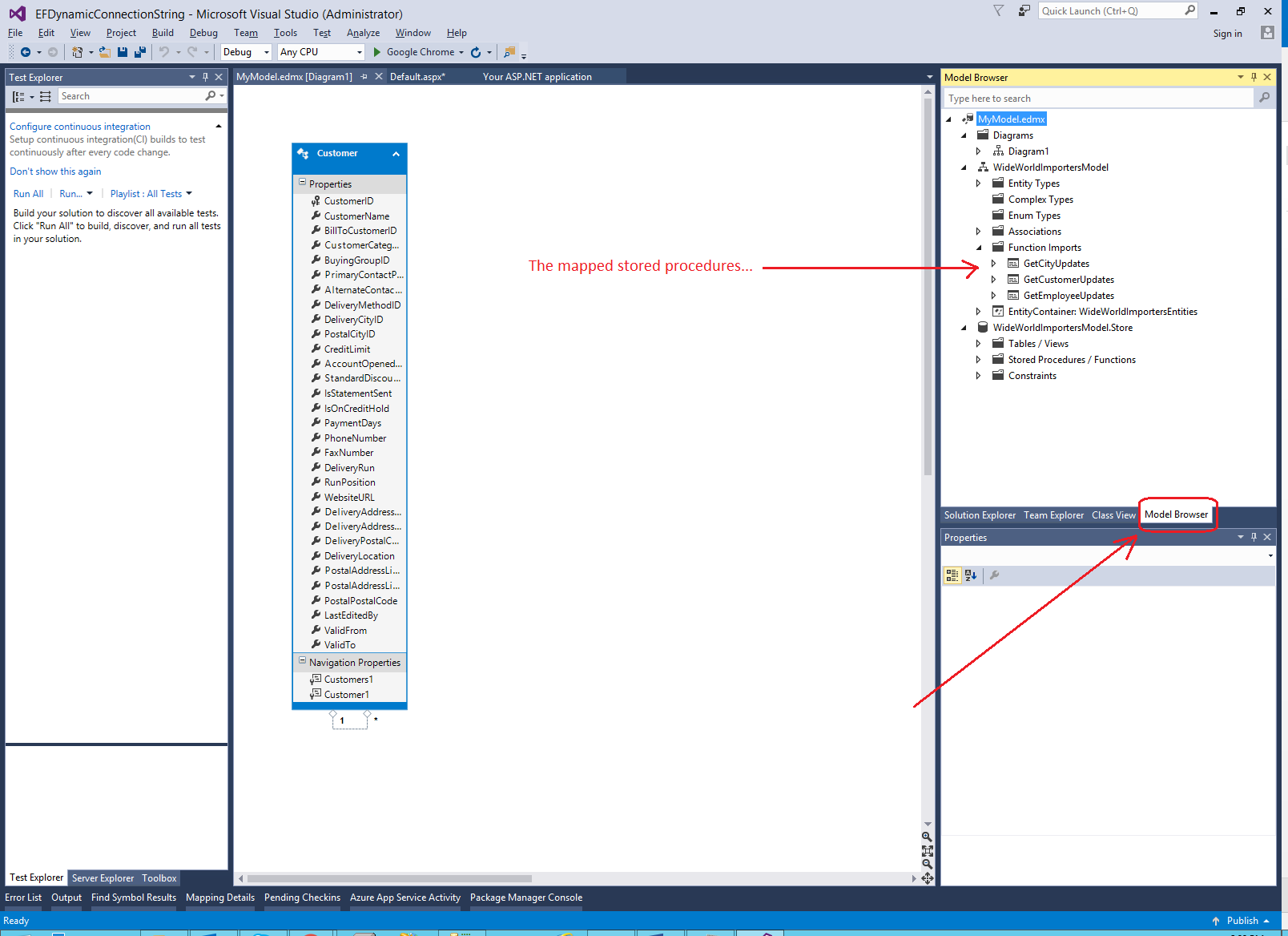
1. Once the connection is established, click “OK”. This will bring you back to the previous screen and should looking like this:



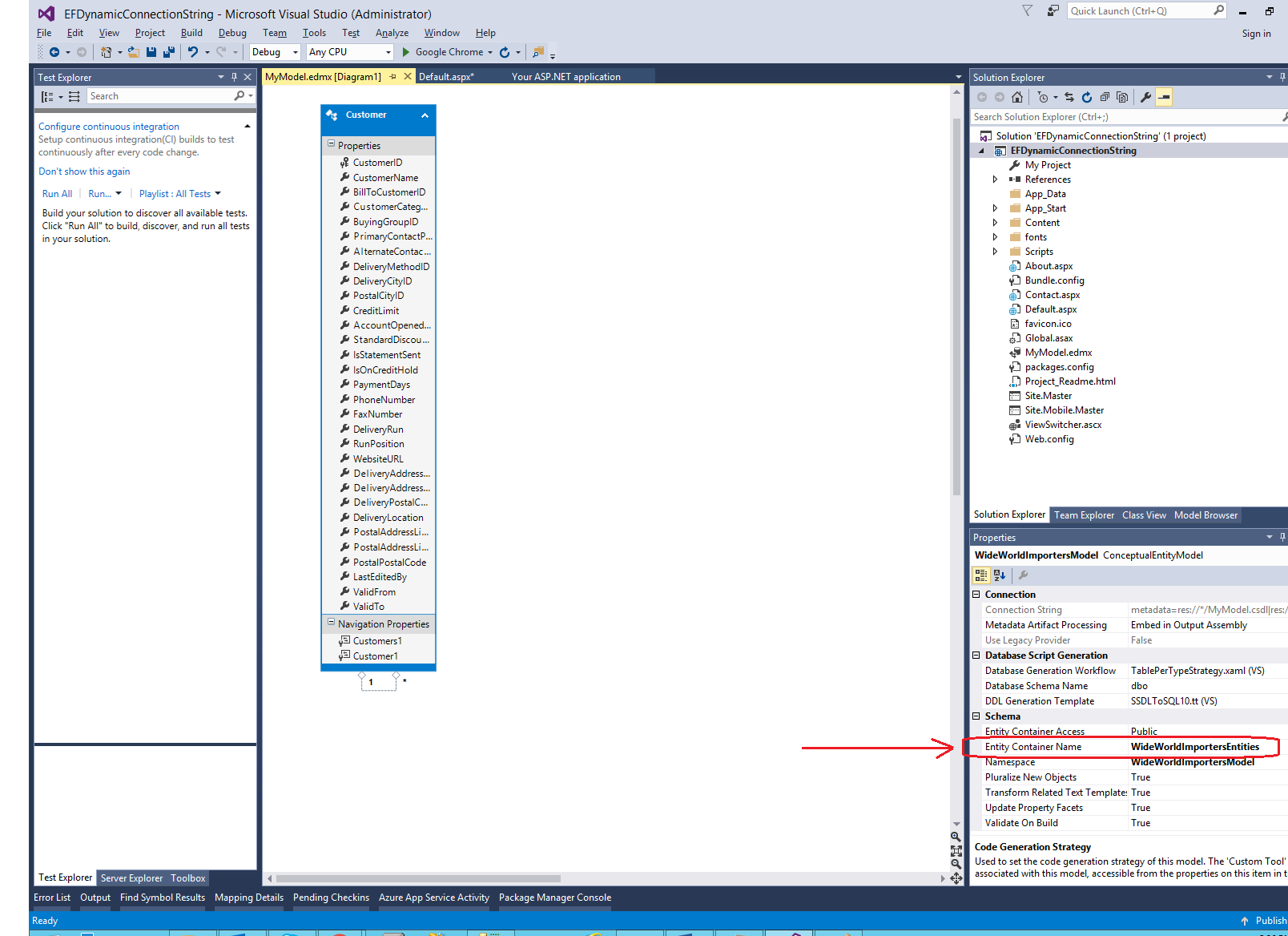
1. Set a name for saving the connection settings in Web.config file. **Make sure** writing down well this name since we will use it later as reference to change the connection string property dynamically. Click “Next”, the following screen will appear:



1. Select the objects from the database you want to be present in your entity framework model. Once you select them all, specify a Model Namespace name and click on “Finish”. Read all dialogs during the creation of the EF model and accept them properly.
2. At this point, you must be able to see at the designer, all the selected objects mapped, except the stored procedures (if selected any). This is fine, stored procedures **won’t** show up here but you must be able to see them by clicking on “Model Browser” tab:



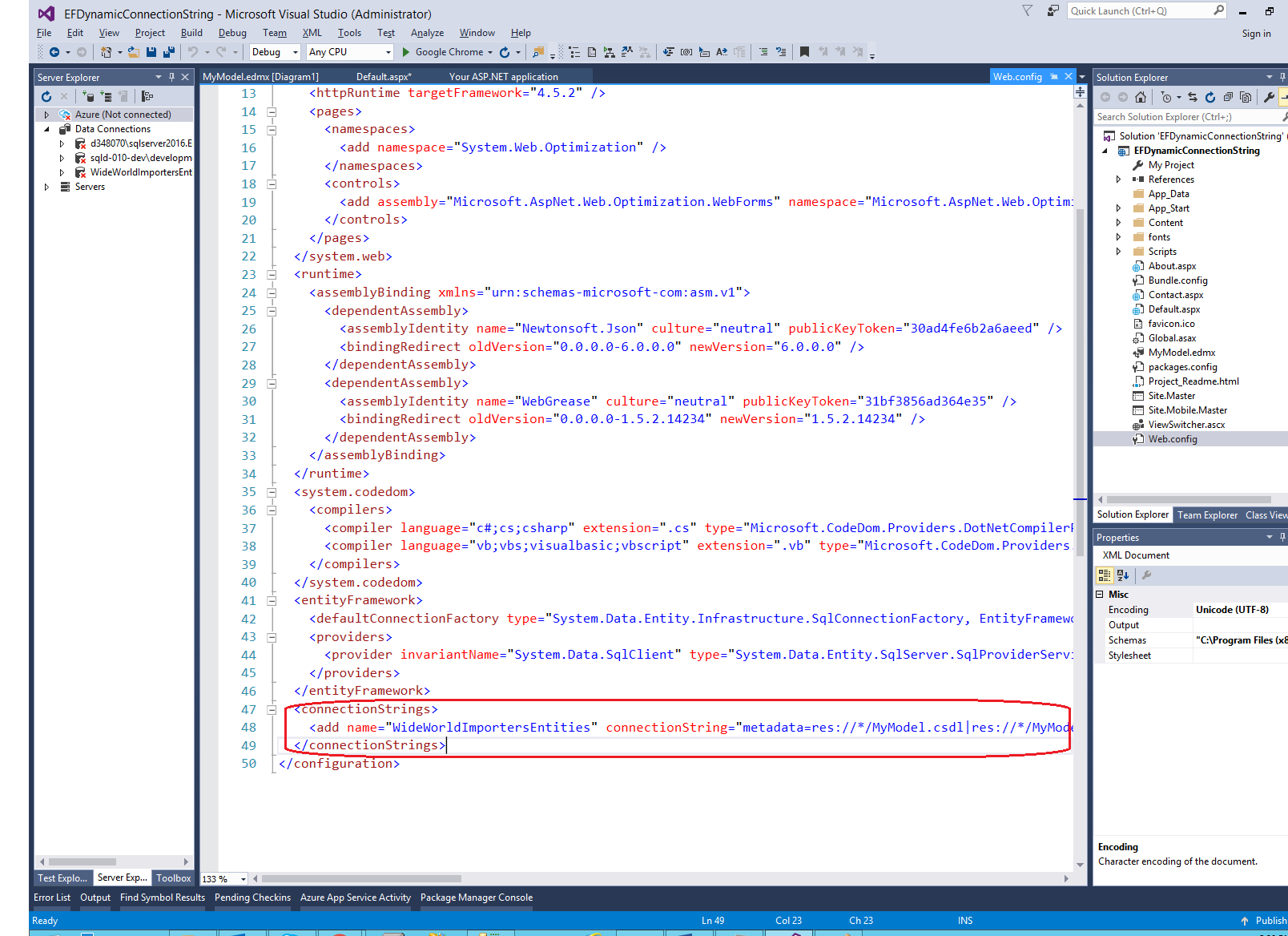
1. If you go back to the Solution Explorer tab and click on any empty space into the created mapped EF designer, you will be able to see the corresponding properties:



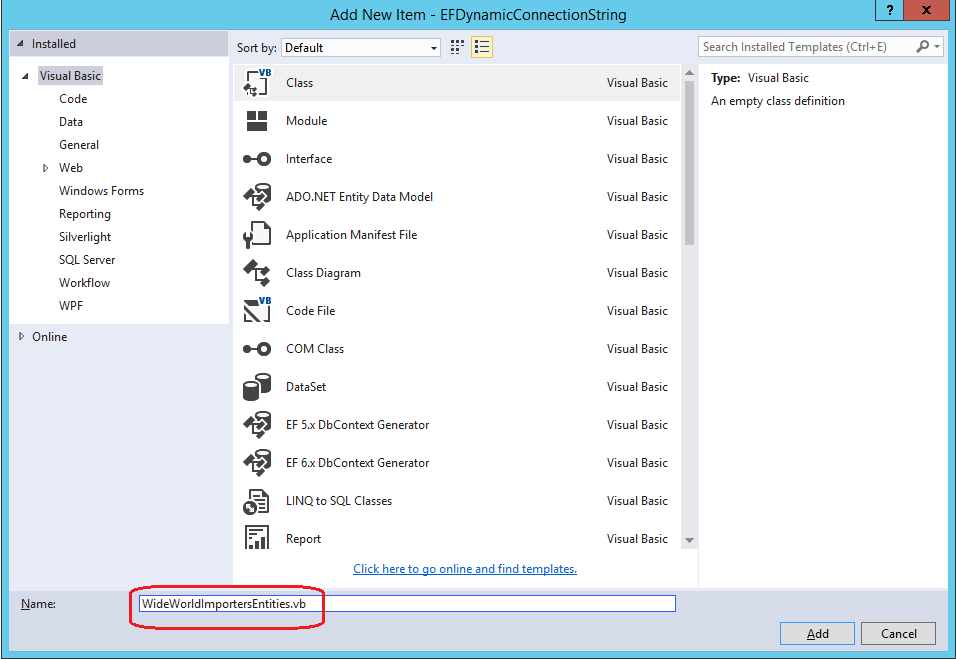
1. The red arrow at the screen above points to the Entity Container\* name. Notice this name is the same one that we used to save the connection settings name in Step 9.

\* The Entity Container is the public class created by Entity Framework to call all the objects mapped into its designer (tables, stored procedures, etc.)

1. Go to Web.config file. You will notice how EF have created a corresponding node for the connection string resulting from the earlier configuration wizard:



1. Since we do not want to read that sensitive data from this file, but from another source dynamically, delete that entry from the Web.config.
2. Right click on the project node, Add-> New Class.



1. Provide for the new class the same name as the one showed for the Entity Container resulting from Step 13.
2. Since it is not advisable making changes to Entity Framework auto-generated classes, we are going to create a new class that extends from DbContext and implements a constructor for passing the connection string as a parameter. For doing that let’s build the new class with the following code:

Imports System.Data.Entity

Partial Public Class WideWorldImportersEntities

Inherits DbContext

Public Sub New(ByVal connString As String)

MyBase.New(connString)

End Sub

End Class

1. Now you can use the EF dynamic connection as shown on the sample project at:

<https://github.com/jlugogarcia/EFDynamicConnectionString>