

Pain Free ASP.NET 2.0 ObjectDataSource Generation

Peter Kellner <http://PeterKellner.net> San Jose, California

March 7, 2006

Quick Start Guide

**Shows how to generate
ObjectDataSources from
Northwind Database**

Table of Contents

Why	3
Who This is For	3
Who This is Not For	4
Time to Run the Program.....	4
Build a VS2005 Project from the Generated Code.....	11
Conclusions and What's Next.....	28

Why

Microsoft provides a very simple layer between the web interface (asp.net 2.0) and the database. The object they use to perform this is the SqlDataSource. It works very well in many cases. It's fast, easy to use and great for demonstration of technology and to show how fast you can program database type applications. It falls apart (IMHO) when the application starts getting larger and requiring more customization. It's also very hard to maintain because at the end of the day, it puts most of what it does in the aspx page itself.

The ObjectDataSource on the other hand completely abstracts the web interface from the database. This means that you can completely isolate the web interface from directly going against data tables. Because this layer sits between the web and the database, it allows for complex transformations of incoming data to the back end databases. This would be near impossible (and unreadable) with the SqlDataSource.

In general, the bad news about ObjectDataSource's is that they are bulky to program. Lots and lots of repetition, lots of attribute lists and lots of chances to make small mistakes that are hard to find. This being the case, it is just screaming out for some kind of code generator and hence the name, "Pain Free ODS Generator".

Who This is For

The Pain Free ObjectDataSource Generator (PF ODS Gen) is for engineers who just read the Why part above and knew exactly what I was talking about. It's for engineers who know how to build ObjectDataSources from scratch but currently only do so when absolutely necessary because of what a pain they are to build.

PF ODS Gen is for engineers who know c# well. Engineers that feel completely comfortable inserting method calls into existing classes when necessary, understand OO programming and easily debug database type applications. It also assumes that the engineer is comfortable writing low level database code (ADO) and that when things don't work as expected, they can look at the source and figure out why.

And finally, it's for C# programmers because I know almost no VB.

Who This is Not For

The PF ODS Gen is not for engineers that are new to ASP.NET or C#. Best place for them to start is the Microsoft Quick Starts. It's not for engineers that are not very comfortable reading SQL as well as ADO.NET calls. It's not for engineers that do not have a lot of experience debugging ASP.NET applications. It's not for engineers that are not very comfortable in the OO paradigm.

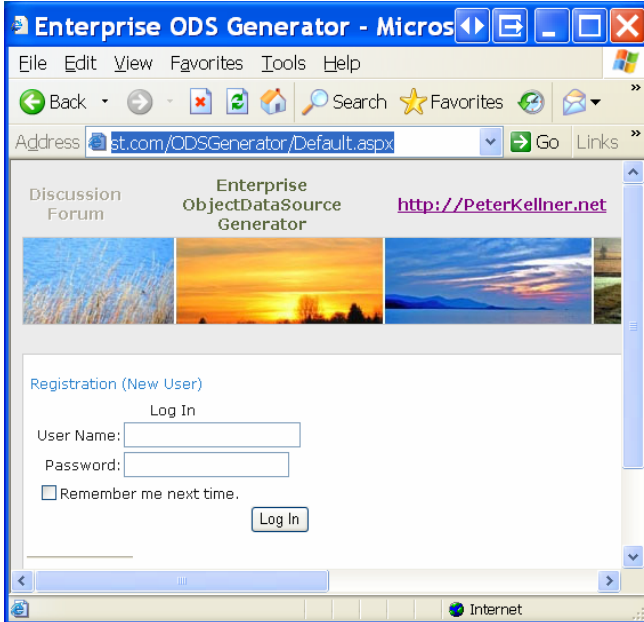
And finally, it is not for engineers that only know VB. You can use this tool with VB but it will always generate c# code that must be placed in a DLL and referenced from the VB Application. (at least that is what they tell me, I have never used VB successfully).

Time to Run the Program

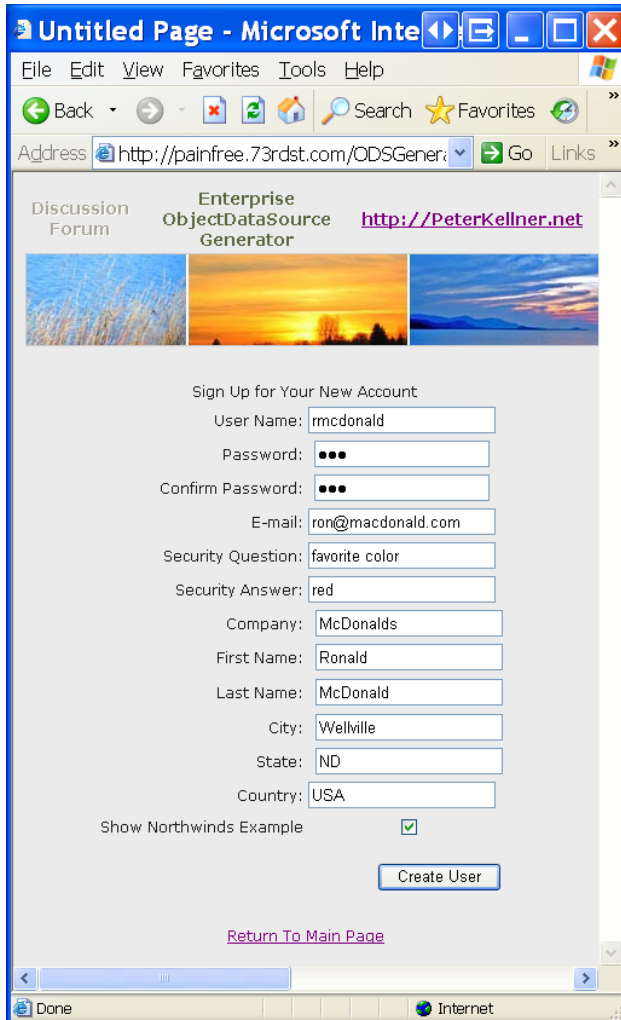
So, I guess it will come as no surprise that we will be generating ObjectDataSource C# code. To make the landing in the program soft for the first time, I've set it up so when you create your account, you will be presented with a bunch of database tables from both the Northwinds and Adventureworks database provided by Microsoft. Since the applications you develop don't use these databases, this is pretty worthless in terms of real work, but it's a good place to start to explain how everything works. In the companion document "Use Your Own Data" we talk about how to get your own data into the web page so you can generate ObjectDataSource's from your own tables.

Finally, a picture.

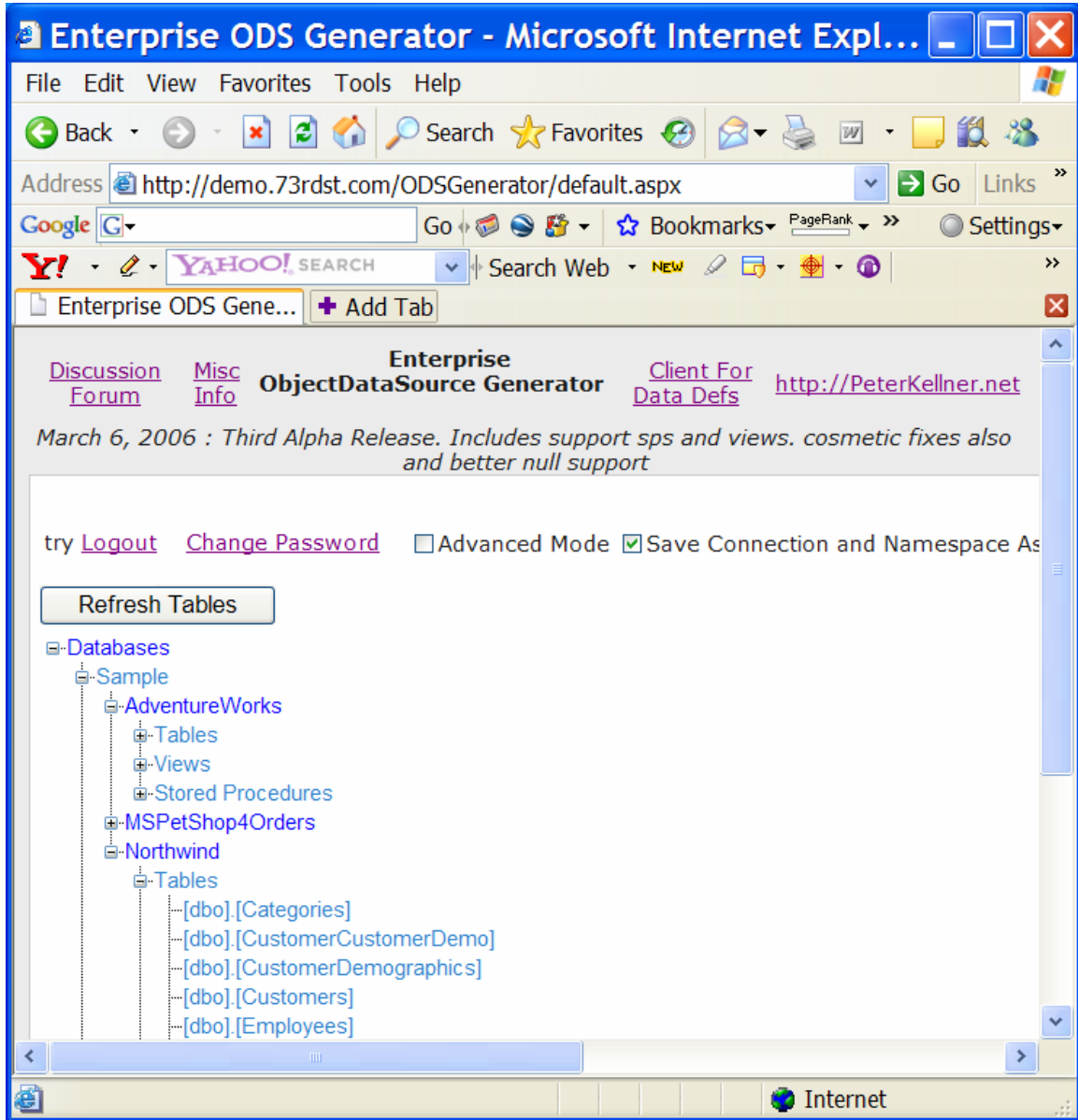
<http://painfree.73rdst.com/ODSGenerator/Default.aspx>



When you first go to this page, you must log in. Associated with your login, you will be able to see all the objectdatasource's you've generated along with all your customizations. (at this point, all we have are the sample Northwind and Adventureworks Databases shown). Each users has tons of customization information associated with them. Before logging in though you must register yourself. Click on the hyperlink "Registration (New User) and you will be presented with the following page.



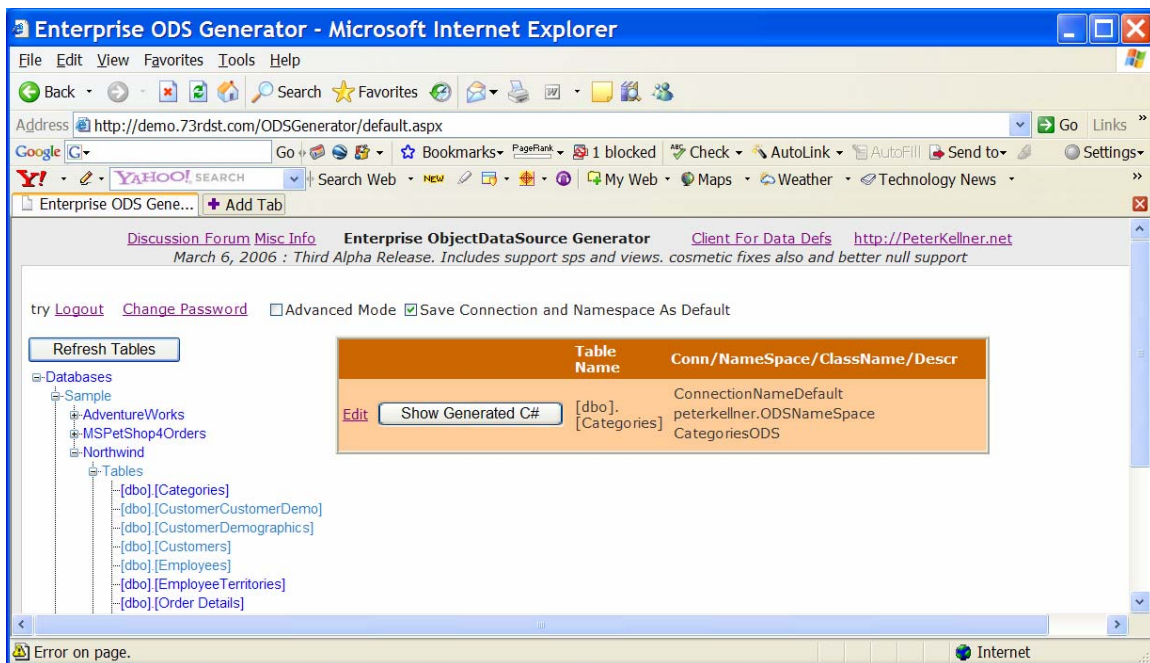
After entering all this information (and leaving the "Show Northwinds Example" box checked), you will be returned to the previous log in page. You will then be presented with the main screen of the PF ODS Gen program. This is where you'll spend all your time when you are generating ODS's. The screen is as follows.



OK, when you got here, NorthWind was not expanded but I figured you could handle one thing I did extra without getting to mad. Things to notice are that there are three databases we can work and each has tables. Those are AdventureWorks, MSPetShop4Orders and Northwind. Only because of magic are they actually there. You never uploaded them, you never asked for them, yet they are existing in your login. Don't worry. You can remove them later once you've uploaded your own database and tables.

Now, notice the checkbox called “Advanced Mode” up near the top. By default this is unchecked. The reason for this is that if you check it and start running the program you will probably run off mad complaining that this is the worst most complicated program you’ve ever seen and never come back. So, please leave it unchecked for the moment. In the companion guide to this (The Advanced Users Guide), I promise to uncheck it with you and provide you with wonderful things you can do well beyond what you can do in the non advanced mode.

OK, so if you’re still reading, let’s create ourselves a Pain Free ObjectDataSource. First thing to do is click on the highlighted hyperlink [dbo].[Categories]. You’ll then see the same screen, but you’ll have an orange line in the middle showing you a couple things as follows.



Basically what this line is showing you is the attributes that will be associated with your generated ObjectDataSource when you push the grey button labeled “Show Generated Code”. Lets discuss the attributes now and what they mean. First though, let me press the underlined edit button (this is actually a gridview of course linked to one of the ObjectDataSource’s created by this program). I’ll show that below zoomed in.

Table Name	Conn/NameSpace/ClassName/Descr
	ConnectionNameDefault
Update	[dbo].
Cancel	[Categories]
	peterkellner.ODSNameSpace
	CategoriesODS

Table Name: This is a non-editable field that really just lets you know that you clicked on [dbo].[Categories].

C# Class Name: This is the class name of the ObjectDataSource that will be generated.

Conn Name / Descr: This is two fields. The top field is the connection name that you will be referencing from your Web.Config file. Since the ObjectDataSource is going after data ultimately, it is the name of the connection you have assigned in your web.Config file. Description is simply some description you give to this particular ODS you are building. You will find in the advanced users guide that you can actually have multiple ODS's for any particular table. In non-advanced mode, you are only allowed to see one so the description is basically meaningless in this mode.

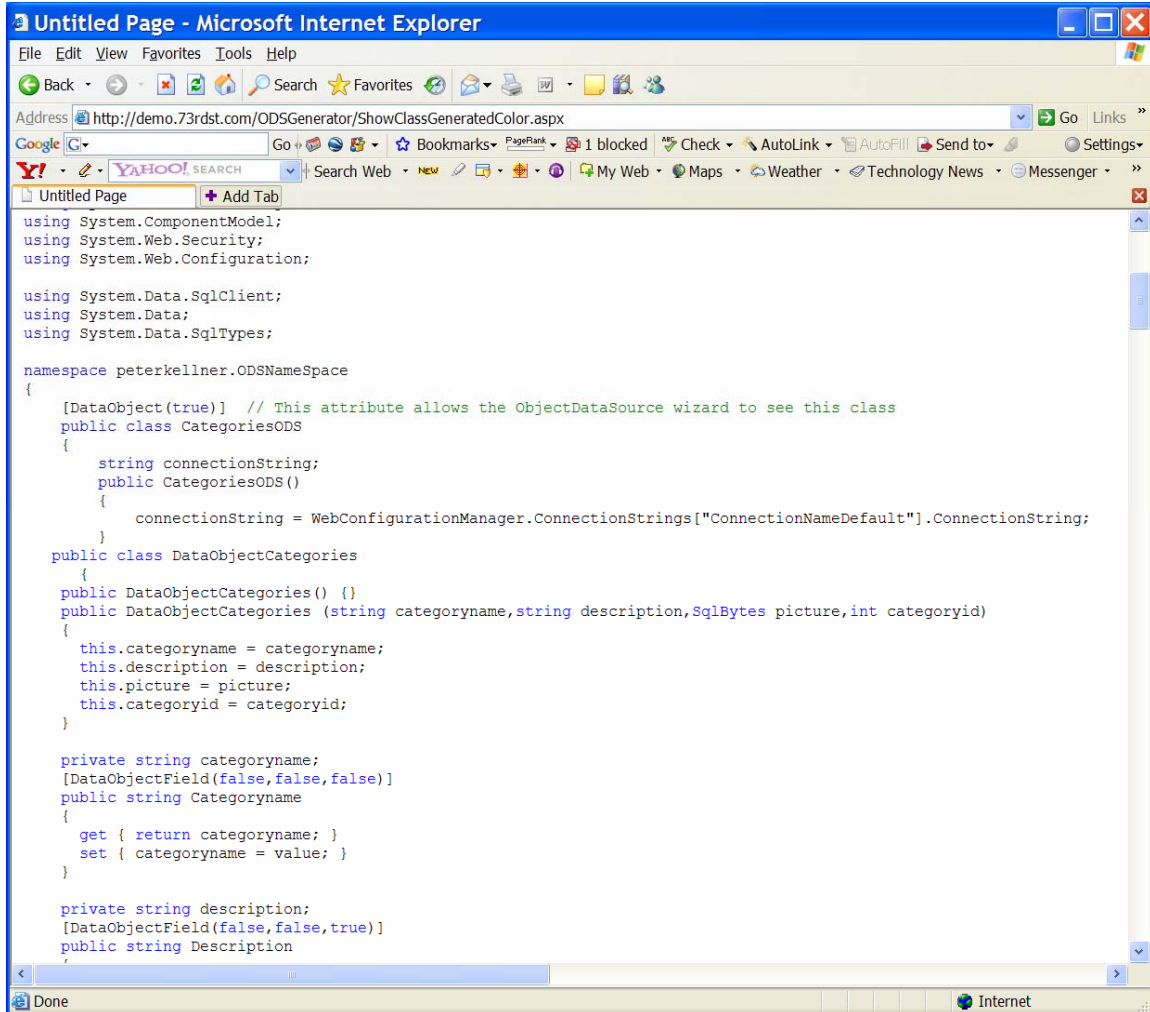
Namespace: The ObjectDataSource is associated with a declare namespace. (at the top of the source you will see lines like the following: `namespace peterkellner.ODSNameSpace { ... }`).

Notice by default there is a check box checked by default on top that says "Save Connection and Namespace As Default". This means, that anything you put into the Conn Name and Namespace will be saved and used again on any new ODS you created. Keep in mind that if you have already visited a table (by clicking on it) the definitions have already been defined so those you will need to either delete (which you can only do in advance mode, sorry) or simply change those values to what you want by using the Edit button.

Finally, time to generate an ObjectDataSource pile of code. To do that, simply press the wide "Show Generated Code Class" button in the orange grid.

dit	<input type="button" value="Show Generated C#"/>	[db [Ca
---------------------	--	------------

When the button is pressed, a complete class will be generated that can be used as an ObjectDataSource. Lets do that now and see what happens and I'll show it below.



The screenshot shows a Microsoft Internet Explorer browser window titled "Untitled Page - Microsoft Internet Explorer". The address bar displays the URL: `http://demo.73rdst.com/ODSGenerator/ShowClassGeneratedColor.aspx`. The browser's status bar at the bottom shows "Done" and "Internet". The main content area of the browser displays the following C# code:

```
using System.ComponentModel;
using System.Web.Security;
using System.Web.Configuration;

using System.Data.SqlClient;
using System.Data;
using System.Data.SqlTypes;

namespace peterkellner.ODSNameSpace
{
    [DataObject(true)] // This attribute allows the ObjectDataSource wizard to see this class
    public class CategoriesODS
    {
        string connectionString;
        public CategoriesODS()
        {
            connectionString = WebConfigurationManager.ConnectionStrings["ConnectionNameDefault"].ConnectionString;
        }
        public class DataObjectCategories
        {
            public DataObjectCategories() {}
            public DataObjectCategories (string categoryname, string description, SqlBytes picture, int categoryid)
            {
                this.categoryname = categoryname;
                this.description = description;
                this.picture = picture;
                this.categoryid = categoryid;
            }

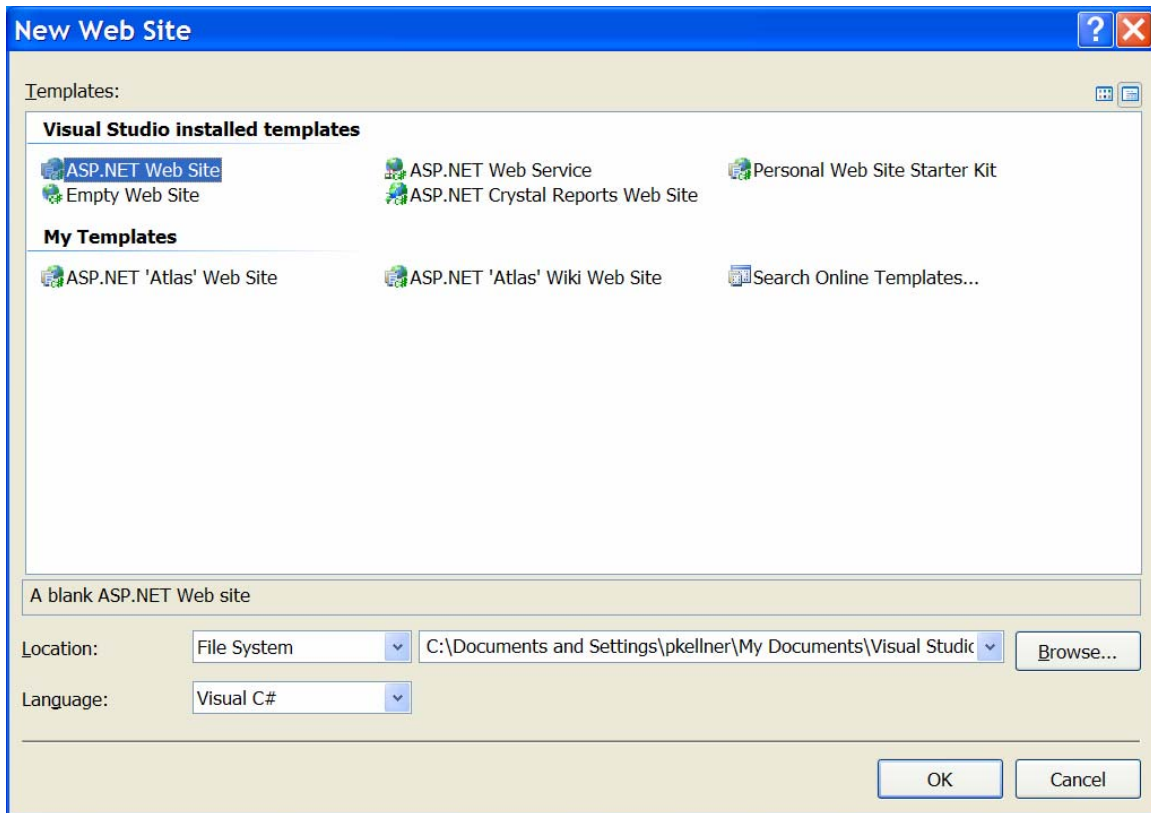
            private string categoryname;
            [DataObjectField(false, false, false)]
            public string Categoryname
            {
                get { return categoryname; }
                set { categoryname = value; }
            }

            private string description;
            [DataObjectField(false, false, true)]
            public string Description
            {

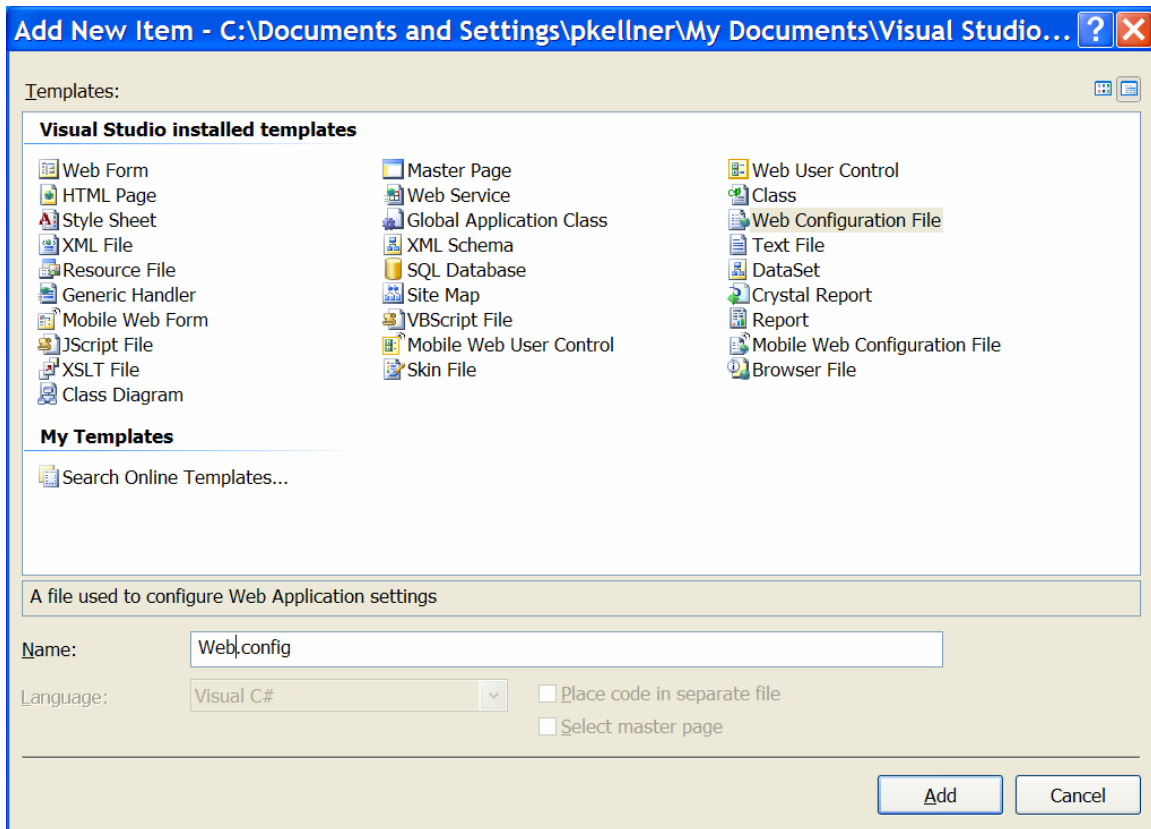
```

Build a VS2005 Project from the Generated Code

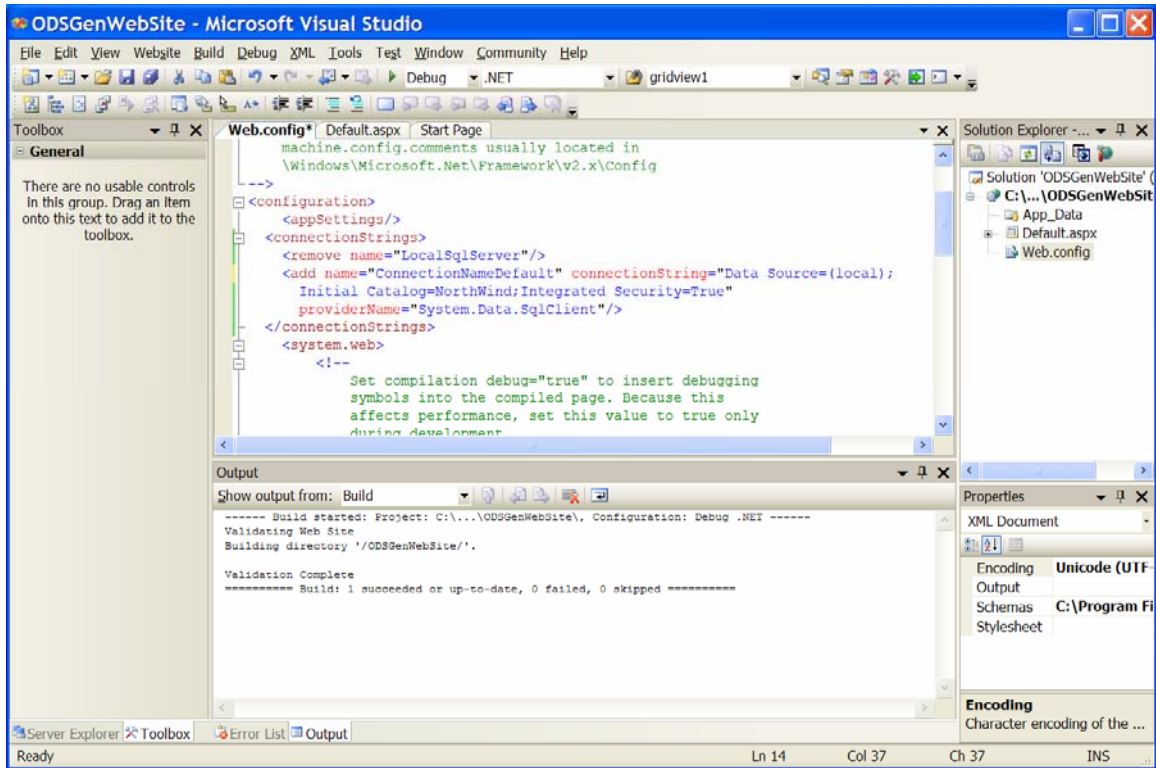
At this point, you need to copy and past this code into VS2005. Just to be extra clear, lets go through creating a VS2005 asp.net project now and build a gridview with this. So, the next section will be mostly pictures with some words of explanation.



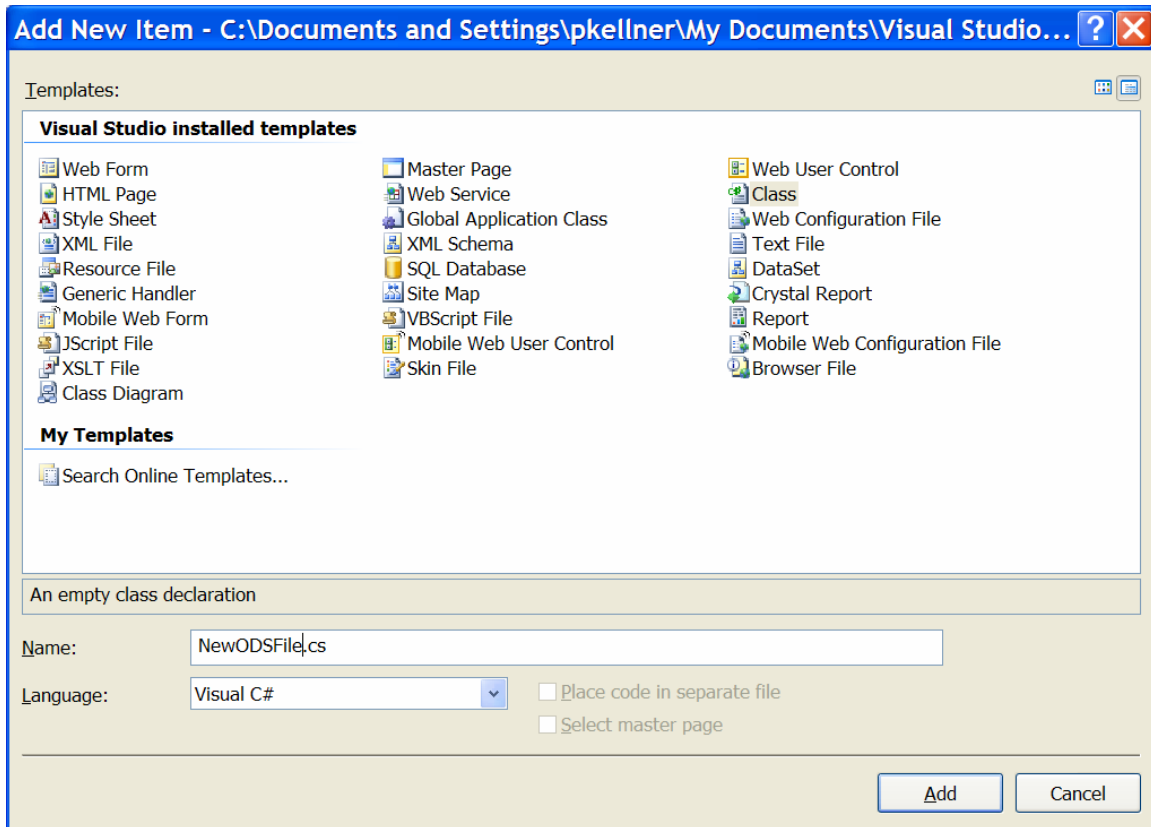
Add a Web.Config File



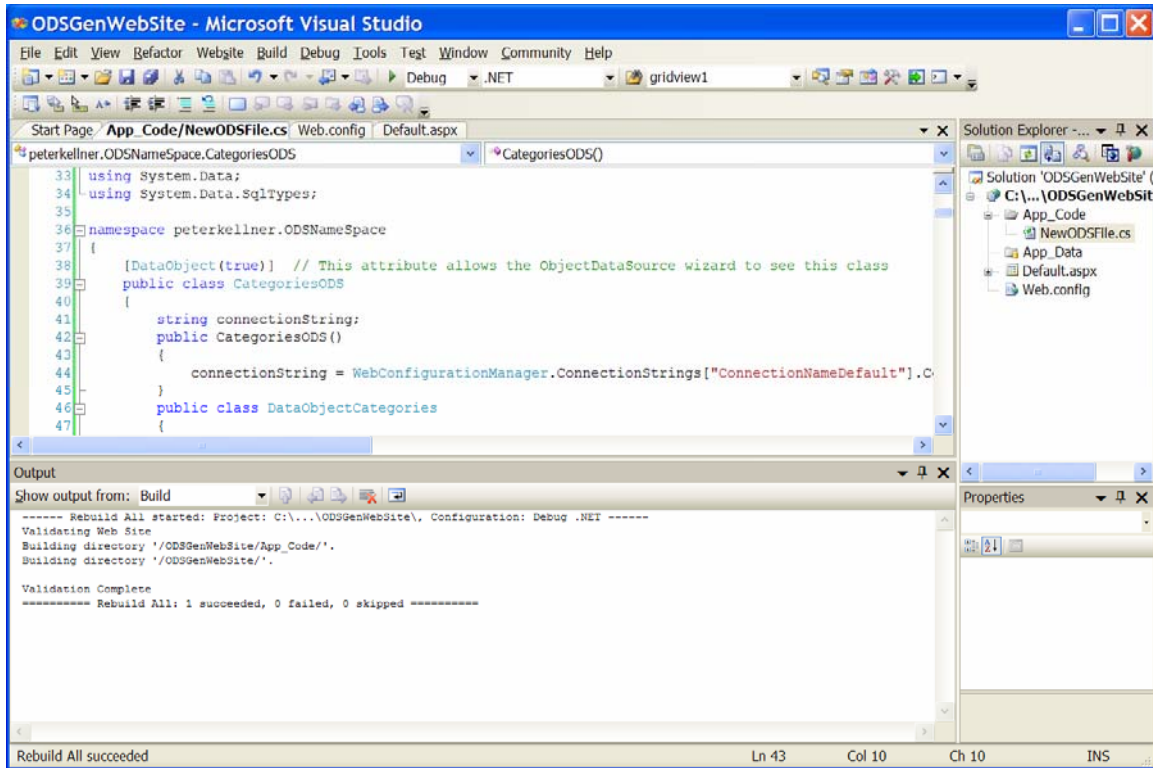
Modify the connection string to point to your NorthWind Database



Create a New csharp class file

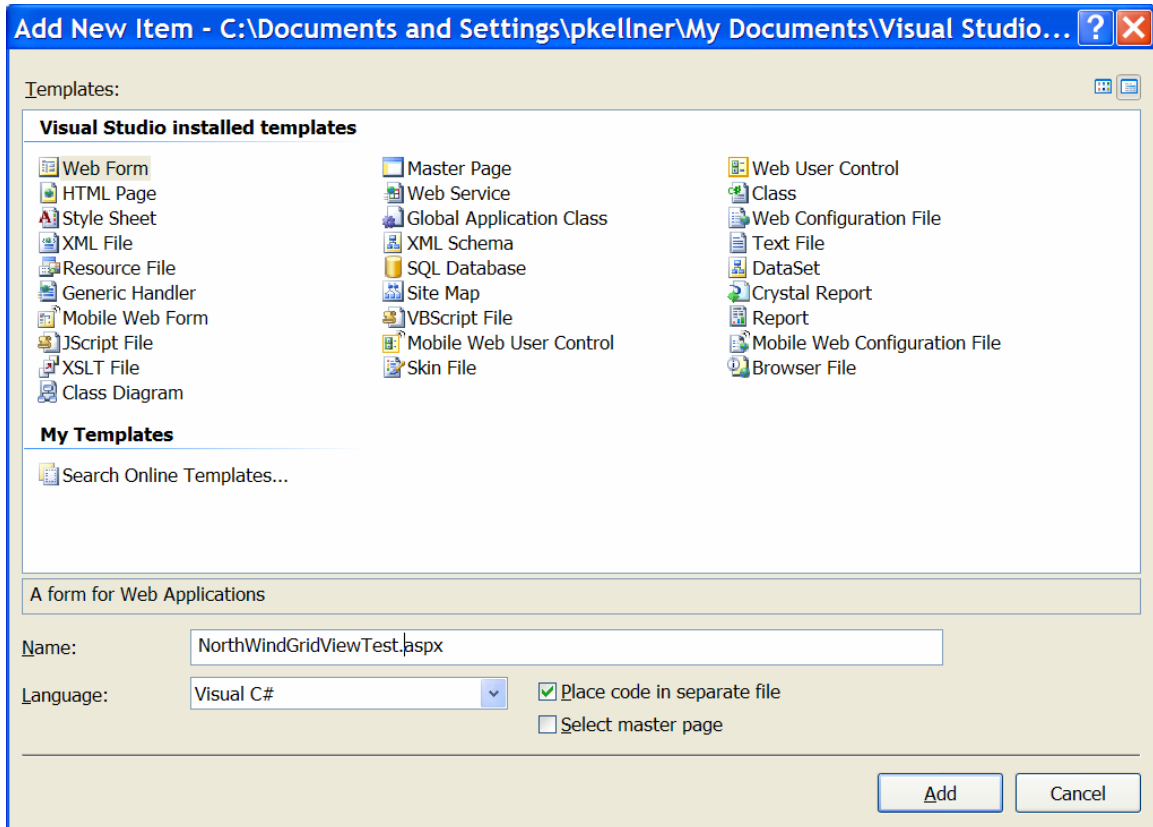


Then, Cut and paste the code from your IE window showing all the code into the newly generated cs file. And you should have something like below.

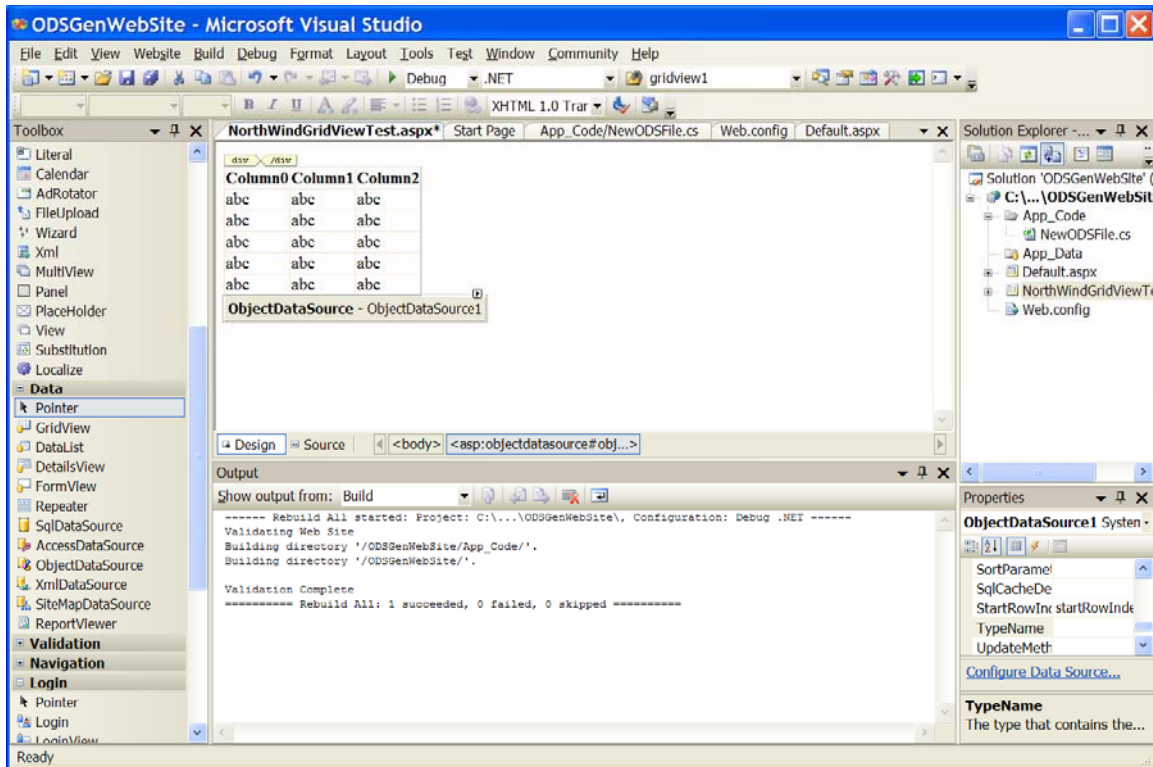


You should be to build this successfully.

Now, Create a new ASP.NET aspx file.



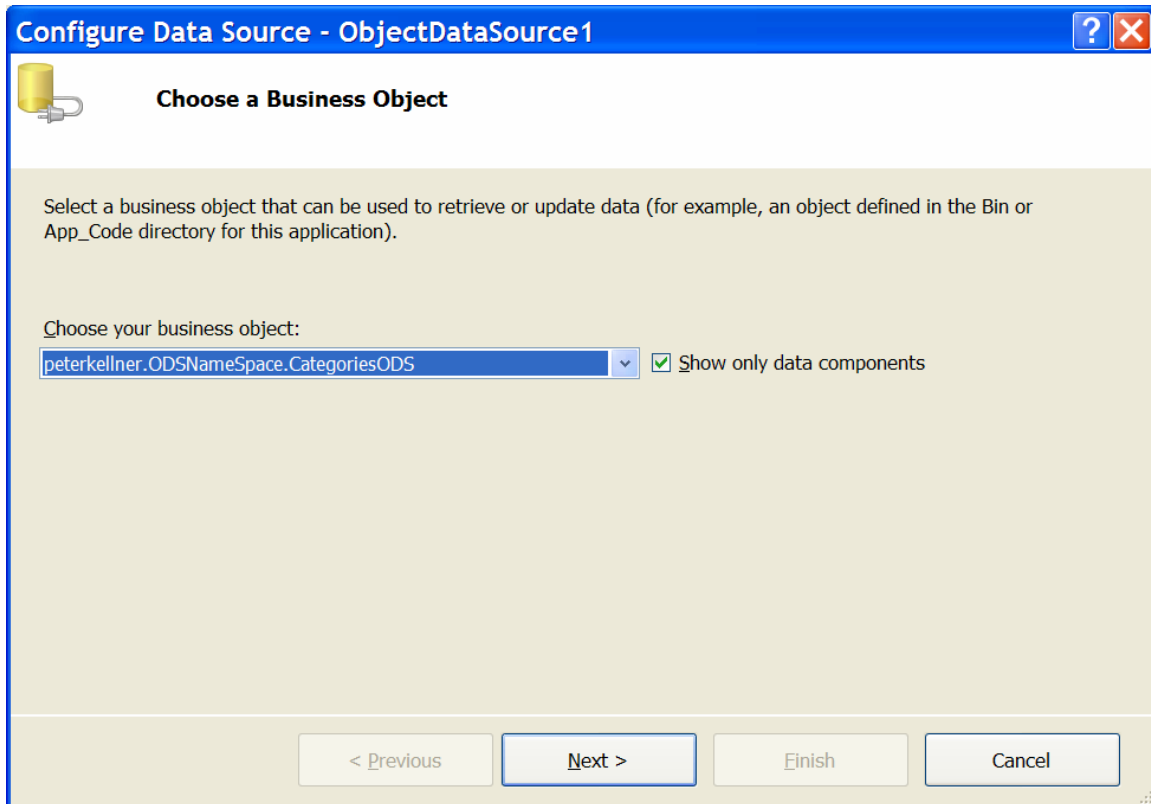
Drop a GridView and ObjectDataSource from the toolbox onto the page



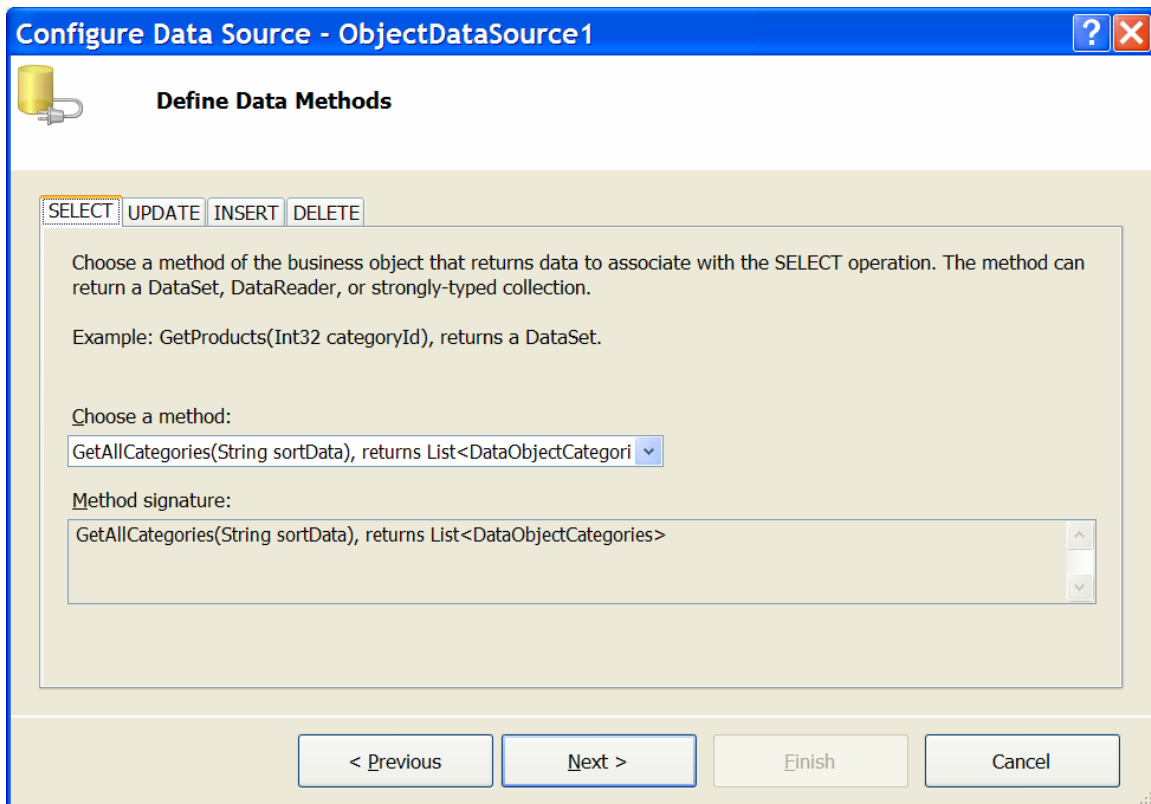
Choose from the little wizard / chevron button Configure Data Source.



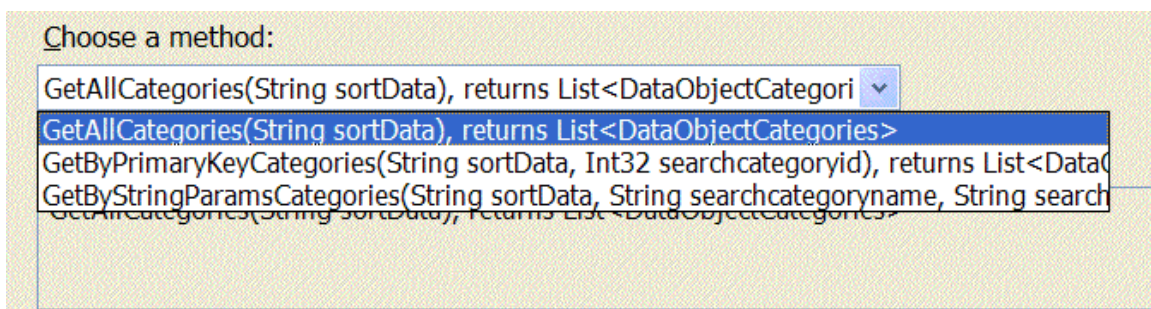
Press Configure Data Source and leave checked “Show only data components”. Choose the only object listed in the dropdown. (this is the class we just built).



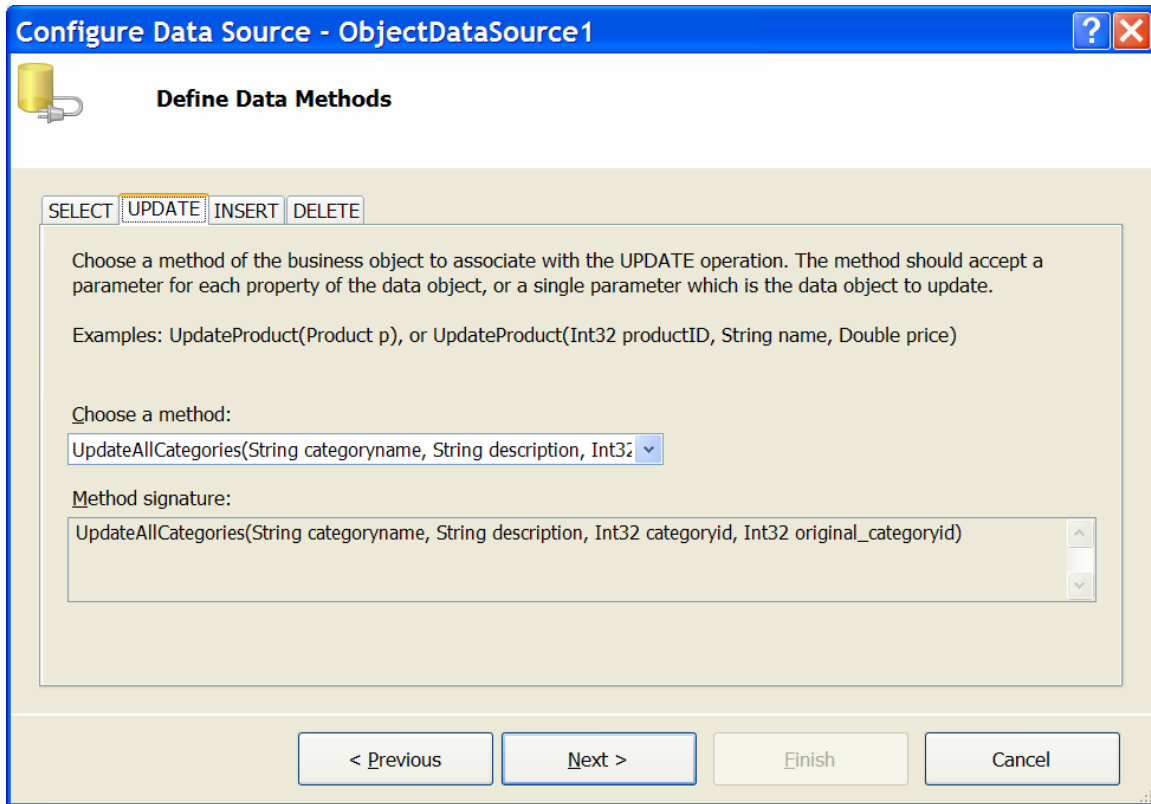
Choose Next



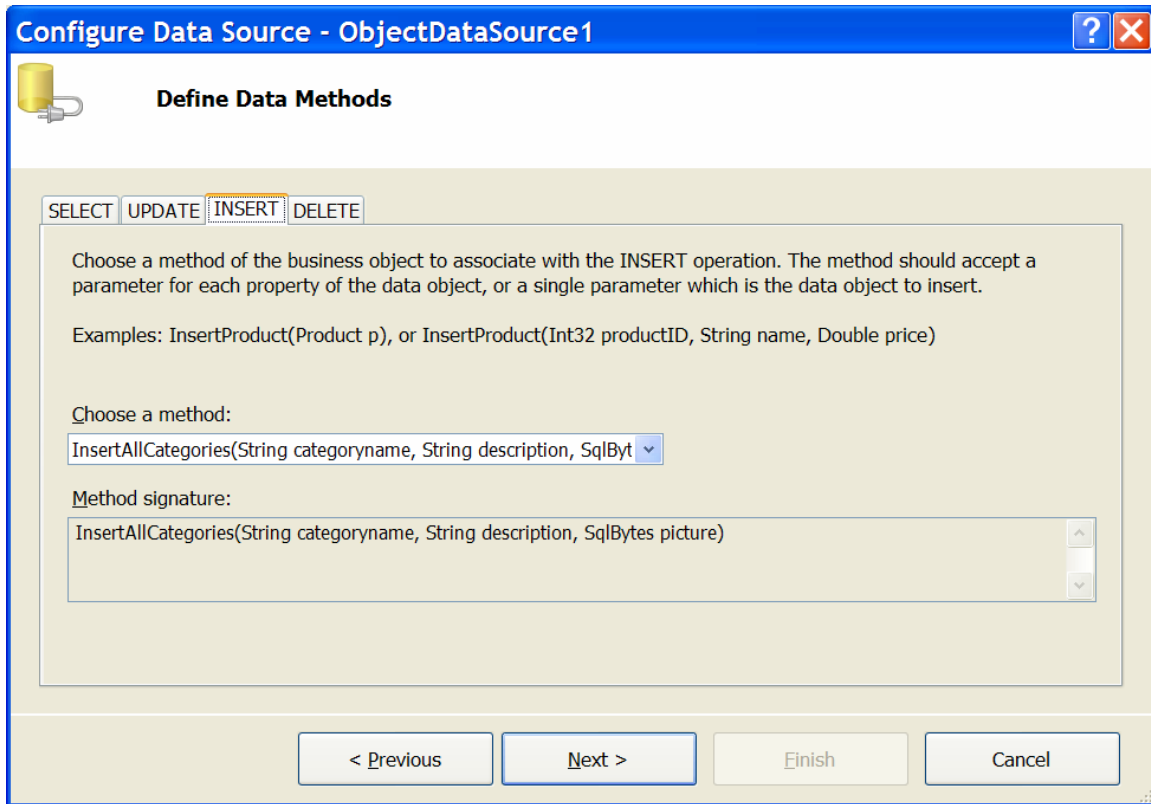
Notice in the drop down you could choose from multiple get methods.



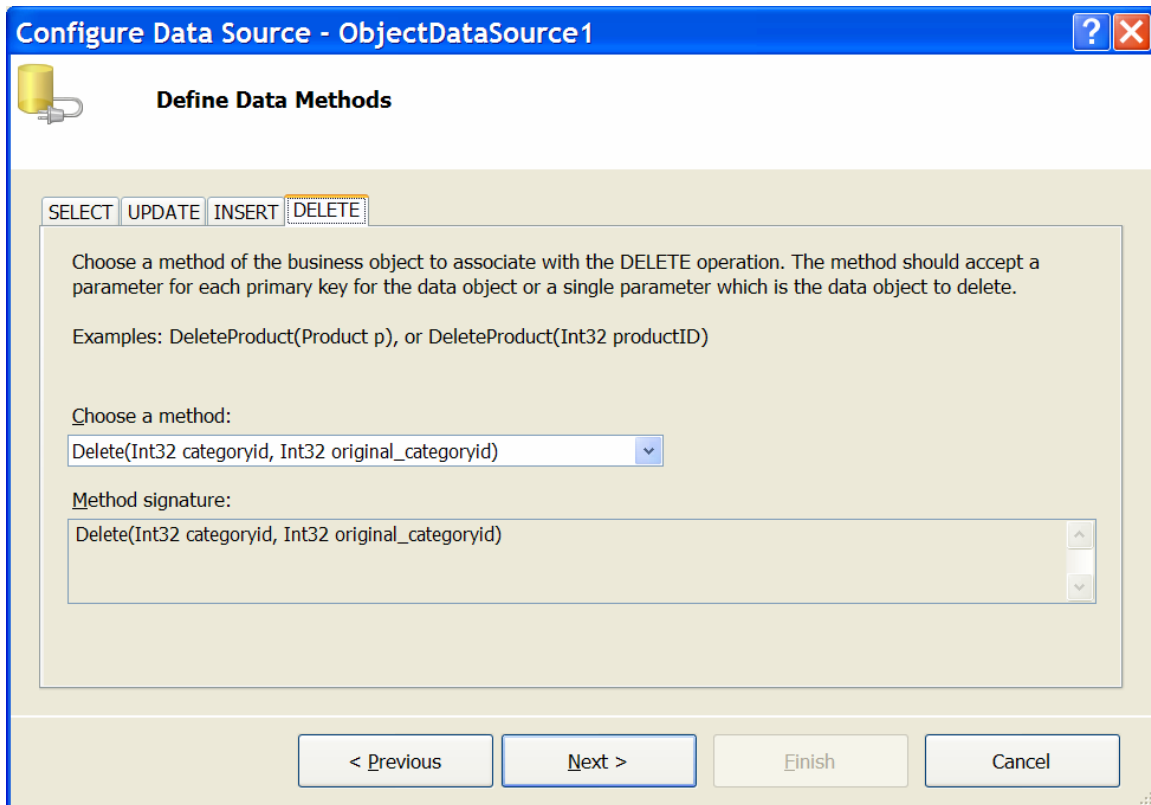
Let it take the default which simply returns all data and lets you sort by any sortable column. Next, click through all the tabs and you will notice there are methods populated for update, insert and delete. (this is a gridview so insert doesn't do us any good but it is here anyhow).



(update method)




(Insert Method)



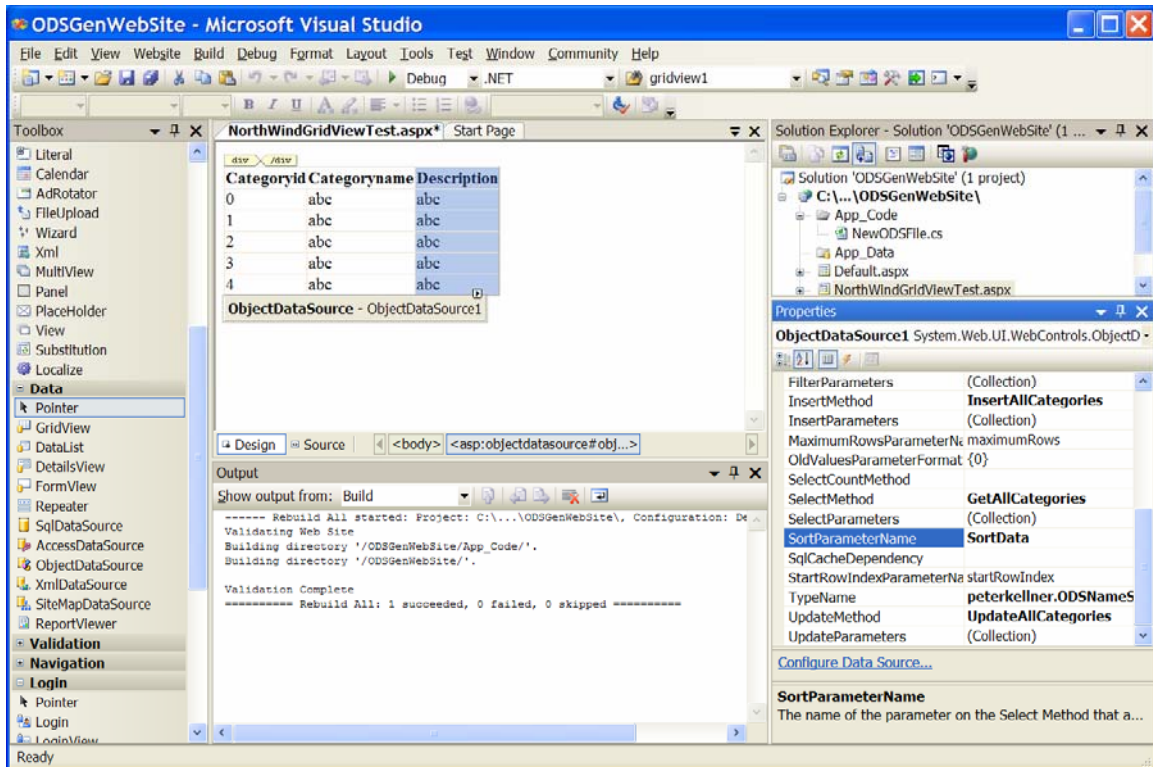
(Delete Method)

Finally, press next and you get the screen that lets you define retrieval parameters. (which we will define none for this example). Press finish.


Now, associate this ObjectDataSource with the gridview by press the chevron on the GridView and saying choose datasource and then choose ObjectDataSource1. (the one we just configured)

GridView Tasks
Auto Format...
Choose Data Source: <input type="text" value="ObjectDataSource"/> 
Configure Data Source...
Refresh Schema
Edit Columns...
Add New Column...
<input type="checkbox"/> Enable Paging
<input type="checkbox"/> Enable Editing
<input type="checkbox"/> Enable Deleting
<input type="checkbox"/> Enable Selection
Edit Templates

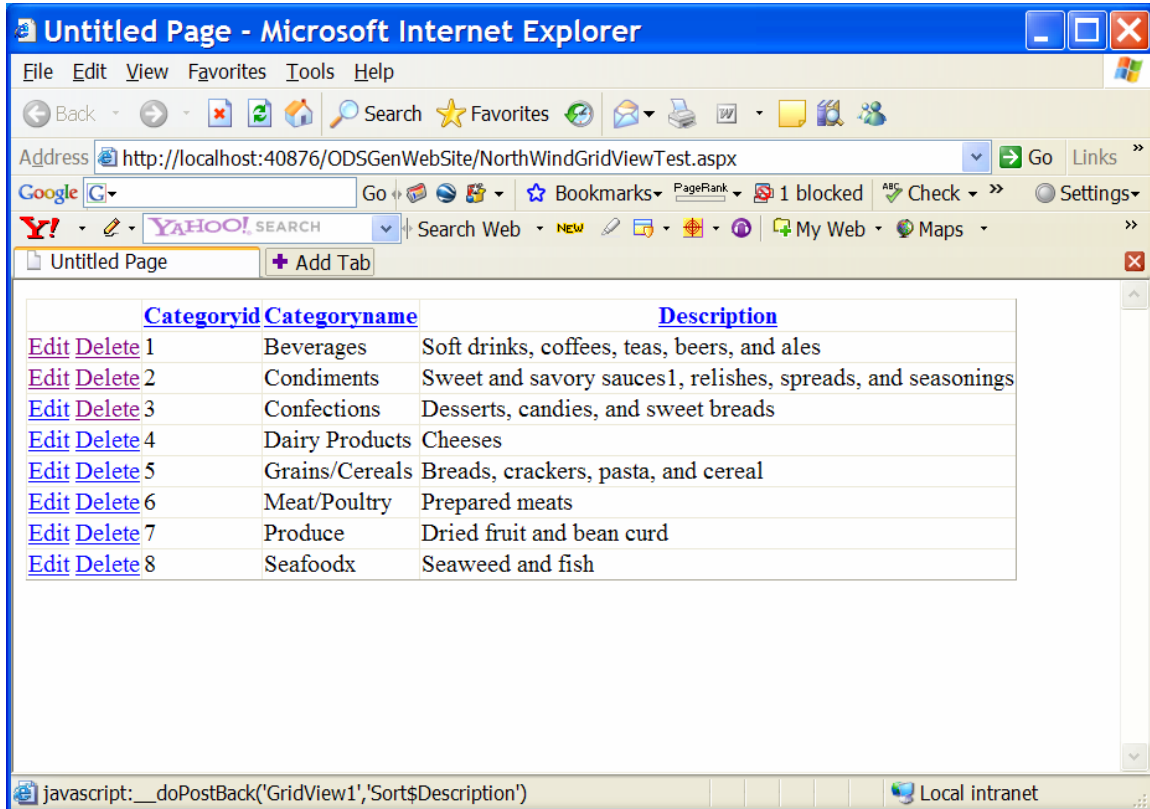
Notice that there is no “Enable Sorting”. Darn. Wish we could have automated that. You need to go into the property editor of the ObjectDataSource and set SortParameterName to “SortData”.



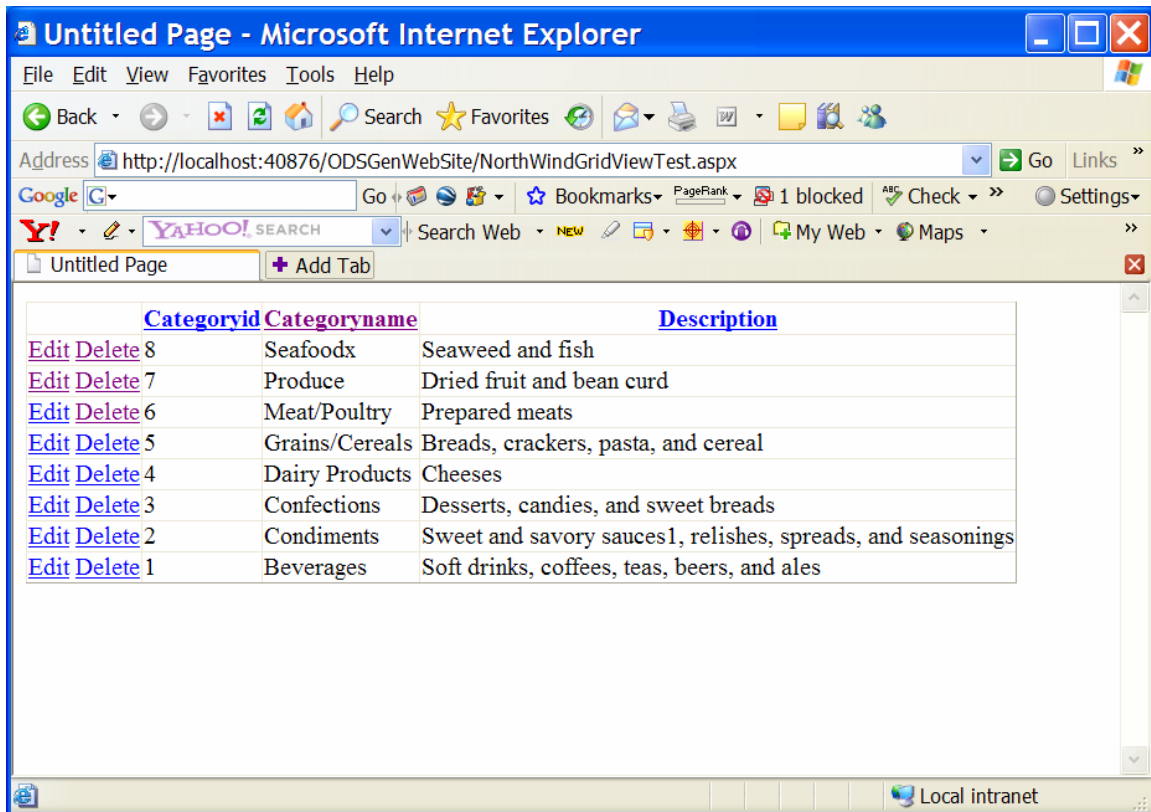
Now, you can go back to the gridview chevron and set Paging, sorting, editing and deleting. (if this were a detailsview you could set inserting also).

GridView Tasks
Auto Format...
Choose Data Source: ObjectDataSource 
Configure Data Source...
Refresh Schema
Edit Columns...
Add New Column...
Move Column Left
Move Column Right
Remove Column
<input checked="" type="checkbox"/> Enable Paging
<input checked="" type="checkbox"/> Enable Sorting
<input checked="" type="checkbox"/> Enable Editing
<input checked="" type="checkbox"/> Enable Deleting
<input type="checkbox"/> Enable Selection
Edit Templates

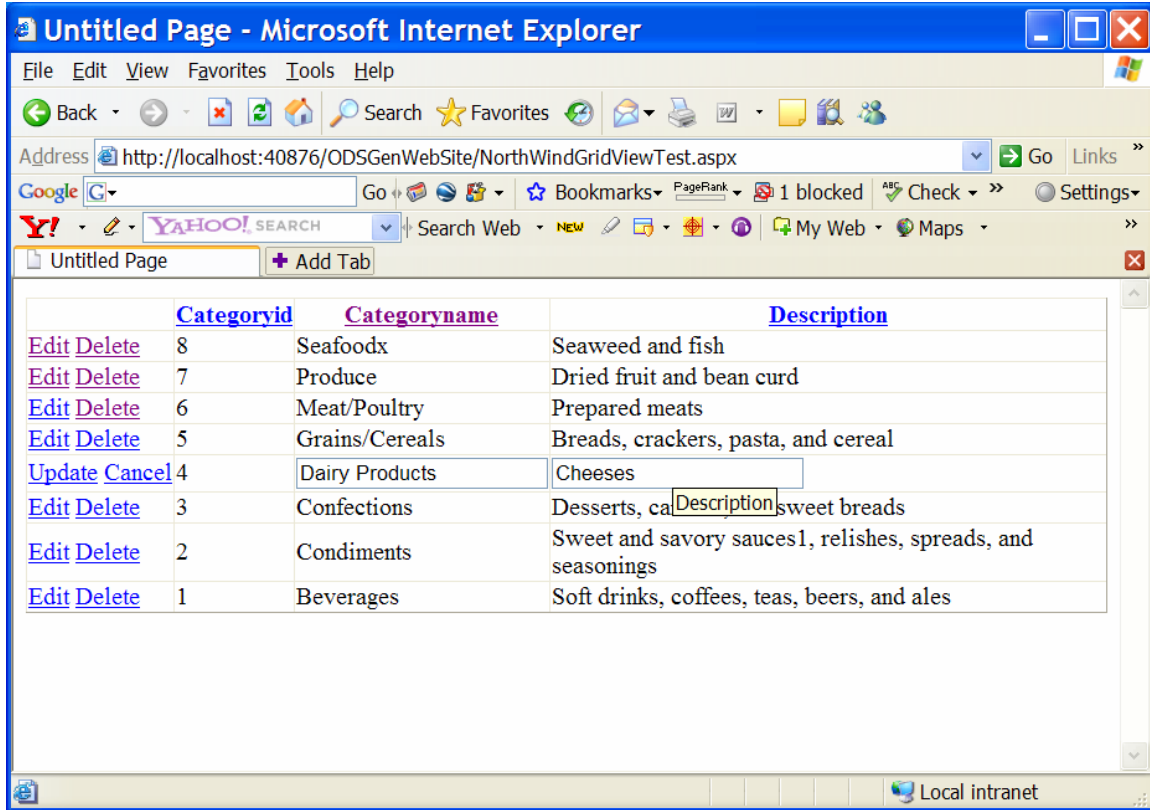
Now, set the default starting page to NorthWindGridViewTest.aspx and run it.



Presto! Click on the CategoryName column header. And see the reverse sort.



Press an edit button.



And you could even try deleting but probably you will get a database error since Northwinds has constraints on categories.

Conclusions and What's Next

That's it for now. We've made an ObjectDataSource from scratch (485 lines of csharp code). Put it to use in an asp.net web page and ran it. There is a lot more you can do with the ObjectDataSource creator but we were running in non-advanced mode and you saw how easy it was to generate.

In other documents, you'll see how to add your own methods. Customize selection and also upload your own database schema so you can make ODS's out of your own data.

