

ObjectDataSource vs. ControllerDataSource

Data Aquarium Framework features automatic data binding and presentation that are performed by the client-side JavaScript library interpreting the content of server-side data controller definitions to produce interactive grids and forms.

Sometimes your project may require a custom functionality that is not supported by the user interface components of the framework. You can still take advantage of excellent support for filtering, sorting and paging of very large data sets via standard *ObjectDataSource* component available in *.NET Framework* or *ControllerDataSource* component that comes with the premium versions Data Aquarium Framework.

The examples described below are based on an application generated with Data Aquarium premium project from sample *Northwind* database with business objects enabled.

OVERVIEW OF BUSINESS OBJECTS

Business objects generated as a part of your project are needed if you plan to develop custom web forms with *ObjectDataSource* components or if you would like to have a programmatic API on top of your database tables.

Business objects are placed in *MyCompany.Data.Objects* namespace where MyCompany is the namespace of your project. Each object name is matched with the name of the database table and is accompanied by a *Factory* class.

The default naming of CRUD methods will yield the following methods for each business object: *Select, SelectSingle, Insert, Update,* and *Delete.* Here is a sample signature of *Select* method of *Shippers* business object.

C#:

```
public static List<Shippers> Select(
    Nullable<int> shipperID,
    string companyName,
    string phone)
```

VB:

```
Public Overloads Shared Function [Select]( _
ByVal shipperID As Nullable(Of Integer), _
ByVal companyName As String, _
ByVal phone As String) As List(Of Shippers)
```

Here are a few examples that show how the business objects can be used to manipulate database information. Developers can create new records; retrieve records by primary key, by example, or by individual field values specified as parameters. Updating and deleting existing objects is a snap.

C#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MyCompany.Data.Objects;
public partial class Demo : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // create a new shipper
        Shippers s = new Shippers();
        s.CompanyName = "Code OnTime";
        s.Phone = "877-467-6340";
```

```
if (s.Insert() != 1)
    throw new Exception ("Failed to create a shipper");
// find a shipper by ID
Shippers s2 = Shippers.SelectSingle(s.ShipperID);
if (s2 == null)
    throw new Exception("Shipper not found");
// find a list of matching shippers by example
Shippers query = new Shippers();
query.Phone = "877 - 467 - 6340";
List<Shippers> list = Shippers.Select(query);
if (list.Count == 0)
    throw new Exception("Shippers not found");
// find a list of matching shippers by values
List<Shippers> list2 = Shippers.Select(null, "code", null);
if (list2.Count == 0)
    throw new Exception("Shippers not found");
// update shipper
list[0].CompanyName = "My Company";
if (list[0].Update() != 1)
    throw new Exception("Failed to update shipper");
// delete shipper
if (list2[0].Delete() != 1)
    throw new Exception("Failed to delete shipper");
```

```
Imports System.Collections.Generic
Imports MyCompany.Data.Objects
```

```
Partial Class Demo
```

Inherits System.Web.UI.Page

}

}

VB:

```
Protected Sub Page Load(ByVal sender As Object,
                        ByVal e As System. EventArgs) Handles Me. Load
    ' create a new shipper
    Dim s As Shippers = New Shippers()
    s.CompanyName = "Code OnTime"
    s.Phone = "877 - 467 - 6340"
    If s.Insert() <> 1 Then
        Throw New Exception ("Failed to create a shipper")
    End If
    ' find a shipper by ID
    Dim s2 As Shippers = Shippers.SelectSingle(s.ShipperID)
    If s2 Is Nothing Then
        Throw New Exception("Shipper not found")
    End If
    ' find a list of matching shippers by example
    Dim query As Shippers = New Shippers()
    query.Phone = "877 - 467 - 6340"
    Dim list As List(Of Shippers) = Shippers.Select(query)
    If list.Count = 0 Then
        Throw New Exception ("Shippers not found")
   End If
    ' find a list of matching shippers by values
    Dim list2 As List(Of Shippers) =
        Shippers.Select(Nothing, "code", Nothing)
    If list2.Count = 0 Then
        Throw New Exception ("Shippers not found")
    End If
    ' update shipper
    list(0).CompanyName = "My Company"
    If list(0).Update() <> 1 Then
        Throw New Exception ("Failed to update shipper")
   End If
    ' delete shipper
```

```
If list2(0).Delete() <> 1 Then
        Throw New Exception("Failed to delete shipper")
        End If
   End Sub
End Class
```

Business object is nothing more than a shell that provides placeholders to all fields of underlining data set. Data manipulation methods are simply passing the parameters to the corresponding methods of object factories.

The purpose of factories is to interact with the *Controller* class of Data Aquarium Framework. This very class is executing all operations requested by AJAX scripts that constitute the other half of the framework. The implication of this is that the XML data controller descriptors are driving the behavior of business objects as well.

As a matter of fact, business objects are selecting data by imitating requests of *JavaScript* components of the framework. This allows complete reuse of business logic and rules that are linked to data controllers of your application.

JavaScript client components are always retrieving the exact number of data fields that are declared in the data controller views. Business objects are automatically configured to use the very first view of the corresponding controller. If your business object has a couple of dozen fields and you select data as described in this article then only the fields that are defined in the first view of the controller are retrieved. You can change that by creating a custom view that lists all fields that you do need and making this view first in the data controller definition. You can also add missing fields to the existing first view of the data controller and mark them as "hidden" if you don't want these fields to be displayed in the user interface and only indent to manipulate the fields in your business logic.

This might seem as an overhead but is done to provide maximum efficiency and code reuse. Continue reading to learn how to benefit from the framework capabilities when building custom web forms.

BUSINESS OBJECTS AND DATA SOURCES

If you don't plan to write custom business rules then your only reason to generate business objects is to take advantage of *ObjectDataSource* data binding features of ASP.NET. Business object factories are constructed to fully support filtering, sorting, paging, and editing via *ObjectDataSource*.

You don't need business objects if you take advantage of *ControllerDataSource* component that comes with Data Aquarium Framework. This component implements a *generic factory* and interacts with *Controller* of your application as custom business object factories do. It means that you are taking advantage of data controller descriptors and gain the same great features. You can filter, sort, page, and edit very large data sets without writing any code at all. You will still be required to list all fields that need to be retrieved by modifying view definitions in the corresponding data controller files.

SIMPLE DATA BINDING

Let's take a look at data binding with both data source components.

Here is the markup of a grid bound to *ControllerDataSource*.

```
<asp:GridView ID="GridView1" runat="server" DataSourceID="Cds1" />
<aquarium:ControllerDataSource ID="Cds1" runat="server"
DataController="Products" DataView="grid1" />
```

The following presentation will be rendered.

) () · ()	http://5	ocalhoxt42	K3/DataSourceDemo/	Demolarq	н.		_	_		• fy 🗶 Google		p
	lome									9.0.4	• () Expr • () Tax	A.+.
Leta Controllera	Harter/C	etal Extra o	ganza Datahase Look	ape .								
ObjectData	Sour	ce vs. C	ontrollerDataS	ource							NORTHWIN	D
ProductName	Supple	rID Catego	ryID QuantityPerUnit	UnitPric	e Unitalad	Rock UnitsDe	Order Reorder	Level Discont	mend Produs	tID SupplerCompanylism	r CategoryCategoryNa	-
0w	1	1	bags	18.0000	28		30			Exotic Liquids	Beverages	
pang	1		24 - 12 or bottles	19.0000	57	-40	25		2	Evenculards	Deverages	
www.interest	+	2	12 - 550 mi bottles	10.0000	13	30	25	10	3	ExotoLigada	Condments	
hef Antonia Journ Seasoning	2	2	48-612389	22.0000	53		¢.	12	+	New Orleans Caput Delight	s Condiments	
Dief Antonia Ambo Mix	2	z	36 boxes	21.3900	0		0	12		New Orleans Caput Delight	Condiments	
irandma's byserbeiry izread	2	2	12-8 64 384	25.0000	120		25	0		Grandma Kelly's Homestean	Condments	
inde Bob's Organic Dried	3	2	12 - 1 b pkgs.	30.0000	15		10	61	7	Grandma Kelly's Homestea	f Produce	
kerthwoods Cranberry Sauce	2	2	12 - 12 co jers	40.0000	6		0	(63)		Grandma Katly's Homestea	i Condenents	
Nahi Kabu Niku	4		18 - 500 g piqu.	97.0000	29		¢	- 62		Tukyo Tradera	Heat,Poulty	
kura			12 - 200 mi jara	31.0000	31	8	0		50	Tokys Traders	Seefood	
Suese Cabrales	5	4	140.040	21.0000	22	30	30		11	Cooperativa de Quesos	Dary Products	
Jueso Manchego a Pasitora	\$	4	10 - 500-g okps.	38.0000	-		0		12	Cooperativa de Quesos Las Cabrar	Dary Products	
iontiu	8		2 kg bex	6.0000	24		8		10	Mayunia	Deafood	
who:		7	40 - 100 g pkgs.	23.2900	35		0		14	Mayumla	Produce	
ener Shami		2	24 - 250 mi buttles	15.5000	39	0	8		- 18	Mexuria	Condiments	
whoia	7	3	32 - 500 g buves	17,4500	29	0	80		26	Pavlova, Ltd.	Confections	
Gios Multon	2	4	20 - 1 kg tire	29.0000	0			. (2)	37	Pavlova, Ltd.	HeatPostty	
amarvon Tigers	7		16 kg pkg.	62.5000	40		0		28	Pavlova, Ltd.	Seafood	
eatine Choosele No.its		3	\$0 boxes x 52 pieces	9.3000	25		5	13	- 19	Specialty Becurts, Ltd.	Confections	
ir Rudney's		3	30-pft boxes	81,0000	-		0	(1)	20	Specialty Decute, LM.	Confections	
a Radhen's cones		3	24 pkps. x 4 peces	10.0000	3	40	5		21	Specially Becuita, Ltd.	Confections	
water's mailebrod	¥	5	24 - 500 g pkgs.	21.0000	304	4	25		22	PE Knadednod AB	Grane,Cereals	
umbrid			12 - 250-g pkps.	9.0000	81		28	.63	. 23	PE Knackebroid AB	Grane/Cereals	
iuarané	10		17 - 101 el cara	4.5000	20			101	24	Refrescos Americanes	Security 1	

Here is how you can link an *ObjectDataSource* component to a grid view and take advantage of *ProductsFactory* class generated as a part of business object library of your application. Open a web form in design mode and select *New data source* option when choose source of data.

SiteMa	ntrolers M pDataSourc	ester/Detail E e - SiteMapl	ktravaganza Database I DataSource1	Lookups	Menbership	
Obje	ectDataS	ource v	s. ControllerDat	taSou	rce	
COMMENT	the community	abe	Griuview rasks	-	_	
abc	abc	00.	Auto kormat			
abc abc	abc abc	abc	Auto Format			
abc abc abc	abc abc abc	abx abx	Auto Format Choose Data Sources	(None)		-
abc abc abc abc abc	abc abc abc abc abc	abx abx abx abx	Auto Format Choose Data Source: Edit Columns Add New Column	(None) (None) <new d<="" td=""><td>ata source></td><td></td></new>	ata source>	

A wizard will show up. Select *Object* and click *OK* button.

Data Source Co	nfiguration Wiza	rd					8 ×
Ļ	Choose a Di	ata Source	Туре				
Where will	the application g	get data from	2				
Access Database	0 Database	Contraction Contra	LINQ	Ubject	ite Map	XML File	
Connect to Specify an II	a middle-tier bu	siness object o	or DataSet in t	he Bin or App_	Code directory	for the application.	
ObjectDataS	iource1						
						ОК	Cancel

Select *MyCompany.Data.Objects.ProductsFactory* as a business object.

Configure Data Source - ObjectDataSource1	2 ×
Choose a Business Object	
Select a business object that can be used to retrieve or update data (for example, an object defined in the Bi directory for this application).	n or App_Code
Choose your business object:	
MyCompany.Data.Objects.ProductsFactory	
MyCompany.Data.Objects.CategoriesFactory MyCompany.Data.Objects.CustomerCustomerDemoFactory MyCompany.Data.Objects.CustomersFactory MyCompany.Data.Objects.EmployeesFactory MyCompany.Data.Objects.EmployeesFactory MyCompany.Data.Objects.EmployeesFactory MyCompany.Data.Objects.ProductsFactory MyCompany.Data.Objects.ProductsFactory MyCompany.Data.Objects.ProductsFactory MyCompany.Data.Objects.ScipersFactory MyCompany.Data.Objects.ScipersFactory MyCompany.Data.Objects.ScipersFactory MyCompany.Data.Objects.ScipersFactory MyCompany.Data.Objects.SupplersFactory MyCompany.Data.Objects.SupplersFactory MyCompany.Data.Objects.TerritoriesFactory MyCompany.Data.Objects.TerritoriesFactory	
< <u>Previous</u> Next > Einish	Cancel

Wizard will automatically select appropriate *Select*, *Update*, *Insert*, and *Delete* methods thanks to the data attributes that are applied to the appropriate factory class methods.

SELECT UPDATE INSER	DELETE				
Choose a method of th DataSet, DataReader, or Example: GetProducts()	e business object that i strongly-typed collect nt32 categoryld), retur	returns data to associ ion. ns a DataSet.	ate with the SELECT o	peration. The method	can return a
Select(Nullable <int32></int32>	productID. String prod	luctName_Nullable<	n . •		
Method signature:					
Nullable <int32> catego Nullable<int16> unitsb discontinued), returns</int16></int32>	oryID, String categoryC nStock, Nullable <int16 List<products></products></int16 	ategoryName, String > unitsOnOrder, Null	quantityPerUnit, Null able <inti6> reorderLe</inti6>	able <decimal> unitPi evel, Nullable<boolean< td=""><td>ice, I</td></boolean<></decimal>	ice, I

Finish the remaining wizard steps without making any further changes. This is the markup generated by wizard.

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"</pre>
    DataKeyNames="ProductID" DataSourceID="ObjectDataSource1" >
    <Columns>
        <asp:BoundField DataField="ProductID" HeaderText="ProductID"
            InsertVisible="False" ReadOnly="True" SortExpression="ProductID"
/>
        <asp:BoundField DataField="ProductName" HeaderText="ProductName"</pre>
            SortExpression="ProductName" />
        <asp:BoundField DataField="SupplierID" HeaderText="SupplierID"
            SortExpression="SupplierID" />
        <asp:BoundField DataField="SupplierCompanyName"
            HeaderText="SupplierCompanyName"
SortExpression="SupplierCompanyName" />
        <asp:BoundField DataField="CategoryID" HeaderText="CategoryID"</pre>
            SortExpression="CategoryID" />
        <asp:BoundField DataField="CategoryCategoryName"
```

HeaderText="CategoryCategoryName"

SortExpression="CategoryCategoryName" />

<asp:BoundField DataField="QuantityPerUnit"</pre>

HeaderText="QuantityPerUnit"

SortExpression="QuantityPerUnit" />

<asp:BoundField DataField="UnitPrice" HeaderText="UnitPrice"

SortExpression="UnitPrice" />

<asp:BoundField DataField="UnitsInStock" HeaderText="UnitsInStock" SortExpression="UnitsInStock" />

<asp:BoundField DataField="UnitsOnOrder" HeaderText="UnitsOnOrder" SortExpression="UnitsOnOrder" />

<asp:BoundField DataField="ReorderLevel" HeaderText="ReorderLevel" SortExpression="ReorderLevel" />

<asp:CheckBoxField DataField="Discontinued" HeaderText="Discontinued" SortExpression="Discontinued" />

</Columns>

```
</asp:GridView>
```

<asp:ObjectDataSource ID="ObjectDataSource1" runat="server"</pre>

DataObjectTypeName="MyCompany.Data.Objects.Products"

DeleteMethod="Delete"

InsertMethod="Insert" OldValuesParameterFormatString="original {0}"

SelectMethod="Select" TypeName="MyCompany.Data.Objects.ProductsFactory"

UpdateMethod="Update">

<SelectParameters>

<asp:Parameter Name="productID" Type="Int32" />
<asp:Parameter Name="productName" Type="String" />
<asp:Parameter Name="supplierID" Type="Int32" />
<asp:Parameter Name="supplierCompanyName" Type="String" />
<asp:Parameter Name="categoryID" Type="Int32" />
<asp:Parameter Name="categoryCategoryName" Type="String" />
<asp:Parameter Name="quantityPerUnit" Type="String" />
<asp:Parameter Name="unitPrice" Type="Decimal" />
<asp:Parameter Name="unitSInStock" Type="Int16" /></asp:Parameter Name="unitSInStock" Type

At first glance the markup of *ObjectDataSource* component seems to be more verbose but will be comparable in size if you define grid view fields for *ControllerDataSource* as well.

The web form will render virtually identically in a web browser with the exception of the field order.

```
SORTING AND PAGING
```

All available product records are being retrieved by both data source configurations.

Let's enable sorting and paging. We will start with *ControllerDataSource*.

Select the grid view and enable sorting and paging.

tock	UnitsOnOrder	ReorderLevel	Discontinued	ProductID	٢.	GridView Tasks					
	0	0	5	0		Auto Format					
	1	1	R	1		Choose Data Source: Cds1					
	2	2		2		Refrech Scheme					
	3	3	T	3		Kereso Sciena					
	4	4		4		Add May Columns					
	5	5	2	5	Ŷ	Add New Column					
	6	6	E	6		V Enable Paging					
	7	7	R.	7		V Enable Sorting					
	8	8	D	8		Enable Editing					
	9	9	V	9		Enable Deleting					
						Enable Selection					
					ľ	Edit Templates					

Here is the changed markup.

```
<asp:GridView ID="GridView1" runat="server" DataSourceID="Cds1"
AllowPaging="True" AllowSorting="True" />
<aquarium:ControllerDataSource ID="Cds1" runat="server"
```

A A	a /betal bet urce vs. and Catego a 1 1	exegence Debbeer to Controller Deb Downlow Ported 2 kg bes 500 ml	okups SOUro 6.0000 18.0000	ce e Unitation 24	Stock UnitsOff	Si •	Level Decemb	• D Bage	t • () Tgols •
tes Controllers Macs DbjectDataSon technitisme Seaplie arbu 6 aklaskson 23 anglmg unbergak 38 appr angle Tolu 4	urce vs. In Catego II 1	evegenze Detabase in ControllerDation DiscontrollerDation 2 lig box 500 mi	okupa aSoturo 6.0000 18.0000	ie e Uwtalat 24	Stock UnitsOn	inder Rearder	Level Documb	nued Eredu	dill Supplied
DisjectDataSo roductitiene Secole antu 6 ekalkoon 23 sugting unbergak 16 spor sogile Tofu 4	urce vs. IIII Catego II 1	ControllerDat 21g box 500 ml		e Uwisła 24	Stock UnitsOnd ©	inder Rearder	Level Disconto	nued Erodu	ct10 SupplierCo
hodiu titiame Supple antu 6 akalkoon 23 aughing unbergack 35 appr sogale Tofu 4	B 1 1	2 kg bes 500 mi	6.0000 18.0000	e Unitalis 24	Stock UnitxOni	Inder Rearder	Level Disconto	nued Frede	ctil) SupplierCa
lanbu 6 akkakison 23 aughing unberjack 35 spor snglife Tofu 4	8 1 1	2 kg box 500 mi	6.0000	24	0	5			
akkalkoon 23 aughing unberjack 35 ager onglife Tofu 4	1	500 mi	18.0000					13	Mayum's
aughing unberjack 35 ager ongife Tofu 4	1			\$7	0	20	10	76	Karklo Cry
onglife Tofu 4		24 - 12 oz bottins	14.0000	52	٥	10		67	Bigfoot Bre
	7	5 kg pkg.	10.0000	4	20	5	13	74	Tokyo Trad
Nisana Piery ot Peoper 2 Not	2	32 - 8 or bottes	21.0500	76	0	٥	13	65	New Orlean Delights
piced Olina 2	2	24 - 8 or jars	17.0000	4	300	20	13	66	New Orlean Delights
ansinup Dried 4 opies	7	50 - 300 g pkgs.	\$3.0000	20	٥	10	13	51	Tokyo Trad
ascarpone 34 abiol	5	24 - 200 g pkgs.	32.0000	5	40	25	0	32	Formagg #
adalu 23	3	24 - 50 g pkgs.	20.0000	30	60	15		49	Karkić Oy
Ishi Kobe Niku -4	.6	18 - 500 g pkps.	97.0000	29	0	0	.58	.9	Tokyo Trad
2245678									

Save and open the page in a web browser.

Paging and sorting is instantly available.

Note that the exact number of visible rows is now retrieved from the database every time you sort or page through the records. This allows you to sort and page through very large data sets. The default page size of *GridView* component is ten. *ControllerDataSource* will never retrieve more than 10 records as configured. Standard *SqlDataSource* component is not able to deliver such performance.

Paging in the *ObjectDataSource* example is configured in a similar fashion. The markup changes are exactly the same. Sorting option is not available though and the entire set of records is automatically retrieved from the database instead of just the records that are rendered on the page.

ObjectDataSource component requires additional instructions to support sorting and perform efficient data retrieval operations.

Select *ObjectDataSource* component and bring up the data source configuration wizard.

ObjectDat	aSource -	ObjectDataSource1	<	ObjectDataSource Tasks
•		Configure Data Source		
Design	Split	Source	<	Refresh Scheppa

Choose the second *Select* method on the second step of the wizard and complete the remaining steps

	ctDataSource1
Define Da	ta Methods
SELECT UPDATE INSER	RT DELETE
Choose a method of th	te business object that returns data to associate with the SELECT operation. The method can return a retronoly-typed collection.
Entered and Cat Dead water	1422 adverse 10. adverse Data Sat
Example: GetProducts()	int32 categoryid), returns a DataSet.
Choose a method:	
Select(Nullable <int32></int32>	productID, String productName, Nullable <in td="" ·<=""></in>
Select(Nullable <int32></int32>	productID, String productName, Nullable <int32> supplierID, String supplierCompanyName, I</int32>
	productiv, string productivame, reliable strick2> suppliesu, string supplier, ompanyvame, r
Select(Products gbe), r	eturns List <products></products>
Select(Products gbe), n Nullable <int16> units1</int16>	eturns List <products> InStock, Nullable<intl6> unitsOnOrder, Nullable<intl6> reorderLevel, Nullable<boolean> Instock, Nullable<intl6> unitsOnOrder, Nullable<intl6> reorderLevel, Nullable<boolean> Instock, Nullable<intl6> unitsOnOrder, Nullable<intl6> reorderLevel, Nullable<in< td=""></in<></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></intl6></boolean></intl6></intl6></boolean></intl6></intl6></products>
Select(Products gbe), r Nullable <int16> units1 discontinued, String so</int16>	eturns List <products> InStock, Nullable<int16> unitsOnOrder, Nullable<int16> reorderLevel, Nullable<boolean> ort, Int32 maximumRows, Int32 startRowIndex, String dataView), returns List<products></products></boolean></int16></int16></products>
Select(Products gbe), r Nullable <int16> units1 discontinued, String se</int16>	eturns List <products> InStock, Nullable<int16> unitsOnOrder, Nullable<int16> reorderLevel, Nullable<boolean> ort, Int32 maximumRows, Int32 startRowIndex, String dataView), returns List<products></products></boolean></int16></int16></products>
Select(Products gbe), r Nullable <int16> unitsl discontinued, String se</int16>	eturns List <products> InStock, Nullable<int16> unitsOnOrder, Nullable<int16> reorderLevel, Nullable<boolean> ort, Int32 maximumRows, Int32 startRowIndex, String dataView), returns List<products></products></boolean></int16></int16></products>
Select(Products gbe), r Nullable <intl6> unitsi discontinued, String se</intl6>	eturns List <products> InStock, Nullable<int16> unitsOnOrder, Nullable<int16> reorderLevel, Nullable<boolean> ort, Int32 maximumRows, Int32 startRowIndex, String dataView), returns List<products></products></boolean></int16></int16></products>
Select(Products gle), r Nullable <intis> units discontinued, String se</intis>	eturns List <products> InStock, Nullable<int16> unitsOnOrder, Nullable<int16> reorderLevel, Nullable<boolean> ort, Int32 maximumRows, Int32 startRowIndex, String dataView), returns List<products></products></boolean></int16></int16></products>
Select(Products gle), r Nullable <intl6> units discontinued, String se</intl6>	eturns List <products> InStock, Nullable<int16> reorderLevel, Nullable<boolean> ort, Int32 maximumRows, Int32 startRowIndex, String dataView), returns List<products> </products></boolean></int16></products>

The second method is similar to the first one but features additional parameters *sort*, *maximumRows*, *startRowIndex* and *dataView*. The three of these parameters are needed to support sorting and efficient record retrieval as dictated by *ObjectDataSource* requirements for high performance business objects. The last parameter allows you to choose the data controller view of Data Aquarium Framework application that must be used as a source of data.

Continue making change to the *ObjectDataSource*. Use Properties Window of Visual Studio to change *SortParameterName*, *EnablePaging*, and *SelectCountMethod* of the component. Change them to *sort*, *True*, and *SelectCount* accordingly.



The sort parameter has been added to the data source markup by wizard. Paging is supported by the extended *Select* method of *ProductsFactory* via *maximumRows* and *startRowIndex* parameters. Method *SelectCount* is available in *ProductsFactory*.

Select the grid and enable sorting.

GridView Tasks	
Auto Format	
Choose Data Source:	ObjectDataSource1
Configure Data Source Refresh Schema	e
Edit Columns Add New Column	
Enable Paging Enable Sorting	
Enable Enable sor	ting of rows on the GridView
Enable Deleting	
Enable Selection	
Edit Templates	

Now paging and sorting in the grid linked to *ObjectDataSource* are efficient and will never read more records from the database than are needed for presentation.

FILTERING

Data filtering is an important element of any application. From the developer's prospective filtering must translate into SQL statements with WHERE clause to be considered efficient. Filtering supported in the standard *SqlDataSource* is performed only by retrieving all records from the database, which does not match this criterion of efficiency.

Both, *ControllerDataSource* and *ObjectDataSource* filtering implementations in Data Aquarium Framework applications are efficient.

Change the markup of *ControllerDataSource* sample web form as shown below.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"</pre>
   AutoEventWireup="true" CodeFile="Demo.aspx.cs" Inherits="Demo" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="Header1Placeholder"
    runat="Server">
    ObjectDataSource vs. ControllerDataSource
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="Header2Placeholder"
    runat="Server">
   Northwind
</asp:Content>
<asp:Content ID="Content4" ContentPlaceHolderID="BodyPlaceholder"
runat="Server">
    Product:<br />
                <asp:TextBox ID="ProductName" runat="server" />
            \langle t.d \rangle
                Supplier:<br />
```

```
<aquarium:DataViewLookup ID="SupplierLookup" runat="server"
                    DataController="Suppliers" />
            <br />
                <asp:Button ID="Button1" runat="server" Text="Go" />
            <asp:GridView ID="GridView1" runat="server" DataSourceID="Cds1"</pre>
AllowPaging="True"
       AllowSorting="True" AutoGenerateColumns="False"
DataKeyNames="ProductID">
        <Columns>
            <asp:BoundField DataField="ProductName" HeaderText="Product Name"
                SortExpression="ProductName" />
            <asp:BoundField DataField="SupplierCompanyName"
                HeaderText="Supplier Company Name"
                ReadOnly="True" SortExpression="SupplierCompanyName" />
            <asp:BoundField DataField="CategoryCategoryName"</pre>
                HeaderText="Category Category Name"
                ReadOnly="True" SortExpression="CategoryCategoryName" />
            <asp:BoundField DataField="QuantityPerUnit" HeaderText="Quantity"</pre>
Per Unit"
                SortExpression="QuantityPerUnit" />
            <asp:BoundField DataField="UnitPrice" HeaderText="Unit Price"</pre>
                SortExpression="UnitPrice" />
            <asp:BoundField DataField="UnitsInStock" HeaderText="Units In</pre>
Stock"
                SortExpression="UnitsInStock" />
            <asp:BoundField DataField="UnitsOnOrder" HeaderText="Units On
Order"
                SortExpression="UnitsOnOrder" />
```

```
<asp:BoundField DataField="ReorderLevel" HeaderText="Reorder</pre>
```

Level"

```
SortExpression="ReorderLevel" />
```

```
<asp:CheckBoxField DataField="Discontinued"
```

HeaderText="Discontinued"

```
SortExpression="Discontinued" />
```

</Columns>

</asp:GridView>

<aquarium:ControllerDataSource ID="Cds1" runat="server"

```
DataController="Products"
```

DataView="grid1">

<FilterParameters>

<asp:ControlParameter Name="ProductName" ControlID="ProductName"</pre>

/>

```
<asp:ControlParameter ControlID="SupplierLookup"</pre>
```

Name="SupplierID"

```
PropertyName="SelectedValue" />
```

</FilterParameters>

</aquarium:ControllerDataSource>

</asp:Content>

You can modify filter parameters visually in *Properties Window* of *Visual Studio* if you select the *ControllerDataSource* component and edit *FilterParameters* properties.

Param <u>e</u> ters:		Parameter source:	
Name ProductName SupplierID	Value ProductName.Text SupplierLookup.SelectedValue	Control Contr	•
	Add Parameter		
		OK Cancel	

Run application and try data filtering in actions.

🏉 н	ome - Wir	ndows Internet Ex	plorer		_	124				x
0	0.	http://localh	ost:42663/DataSour	ceDemo/Den	nc 🕶 😽	× Ga	ogle		م	•
*	* 👔	9 Home			1	• • 🔊	- 🖶	▼ 🔂 <u>P</u> age	 ▼	>>
Data	Data Controllers Master/Detail Extravaganza Database Lookups ^									
Ob	ObjectDataSource vs. ControllerDataSource NORTHWIND									
Pro	Product: Supplier: m Formaggi Fortini s.r.l.									
	Product Name	Supplier Company Name	Category Category Name	Quantity Per Unit	Unit Price	Units In Stock	<u>Units On</u> <u>Order</u>	Reorder Level	Discontinued	
Mas Fabi	carpone ioli	Formaggi Fortini s.r.l.	Grains/Cereals	24 - 200 g pkgs.	32.0000	9	40	25		
Moz Giov	zarella di /anni	Formaggi Fortini s.r.l.	Dairy Products	24 - 200 g pkgs.	34.8000	14	0	0		
© 2	009 MyCom	pany. All rights res	erved.							-
Done				😌 Inter	rnet Prot	ected Mo	de: On		🔍 100% 🛛 🗸	

You can bind filter parameters to any ASP.NET components available to you. This sample is using standard *TextBox* and *DataViewLookup* component found in Data Aquarium Framework.

If you copy the markup for the table of filtering parameters to the *ObjectDataSource* sample and set parameter binding in the object data source configuration wizard then you will achieve exactly the same capability. There is a slight difference in configuring parameters. All available parameters of *ProductsFactory.Select* method are listed in parameter configuration step of the wizard.

The wizard has detec	ted one or more parameters in	your SELECT m	ethod. For each paramete	r in the SELECT me	thod, choose a
source for the param Paramgters:	ieter's value.		Parameter gource:		
Name	Value		Control	-	
productID	ProductName Text		⊆ontroIID:		
supplierID	SupplierLookup.Select	edValu	SupplierLookup		
supplierCompanyN categoryID	ame	-	DefaultValue:		
			Show advanced prop	erties	
Method signature:					
Select(Nullable <int3 Nullable<int32> cat Nullable<int16> uni</int16></int32></int3 	2> productID, String productA egoryID, String categoryCateg tsInStock, Nullable <int16> uni</int16>	lame, Nullable< oryName, String tsOnOrder, Null	Int32> supplierID, String s quantityPerUnit, Nullable able <int16> reorderLevel,</int16>	upplierCompanyN «Oecimal» unitPri Nullable <boolean< td=""><td>ame, • ce, > discontinued, •</td></boolean<>	ame, • ce, > discontinued, •

UPDATE, INSERT, DELETE

Update and *Delete* operations are automatically enabled for both data source components if you enable support for these features in the grid view. Here is the *ControllerDataSource* sample application with *Classic* auto-formatting applied to it.

									or route to start
и фи (@н	lome					G • D	- (#)	• Dege	• () Tools •
ata Controllers	Master/Detail Extravagana	a Database	Lóokups						
ObjectData	Source vs. Cont	trollerDa	taSource					NOF	THWIND
roduct:	Supplier: (select) 🛋 Go								
	Product Name	Supplier Company Name	Category Category Name	Quantity Per Unit	Unit Price	Units In Stock	Units On Order	Reander Level	Discontinued
dit Delete Select	Oia	Exotic Liquids	Beverages	10 boxes x 20 bags	18.0000	39	0	10	
idit Delete Select	Chang	Exotic Liquids	Beverages	24 - 12 oz böttles	19.0000	17	40	25	
Indate Lanual	Aniseed Syrup	Emolic Logaria	Continuents	12 - 550 mi bottles	10.0000	13	70	25	
idit Delete Select	Chef Anton's Cajun Seasoning	New Orleans Catun Delights	Condiments	48 - 6 oz jars	22.0000	53	0	0	0
idit Delete Select	Chef Anton's Gumbo Mix	New Orleans Calun Delights	Condiments	36 boxes	21.3500	0	0	0	92
Edit Delete Select	Grandma's Boysenberry Spread	Grandma Kelly's Homestead	Condiments	12 + 8 oz jars	25.0000	120	0	25	0
idit Delete Select	Unde Bob's Organic Dried Pears	Grandma Kelly's Homestead	Produce	12 - 1 lb pkgs.	30.0000	15	0	10	٥
idit Delete Select	Northwoods Cranberry Sauce	Grandma Kelly's Homestead	Condiments	12 - 12 oz jars	40.0000	6	0	0	
dit Delete Select	Mishi Kobe Niku	Tokyo Traders	Meat/Poultry	18 - 500 g pkgs.	97.0000	29	0	0	(2)
dit Delete Select	Dura	Tokyo Traders	Seafood	12 - 200 ml jars	31.0000	31	0	0	
				12355628					

Use standard *DetailsView* component if you want to be able to insert new records. Here is the *ObjectDataSource* sample with *DetailsView* component that has *Classic* autoformatting applied to it.

Home - Windows Interne	et Explorer						
🕒 🗸 🖉 http://localhost:42927/DataSourceDe 🗸 47 🗙 Google							
👷 🏟 🏉 Home 🏠 🔹 🗟 👻 🖶 Page 💌 🍈 Tgols 👻 🦄							
Data Controllers Master/De	tail Extravaganza	Database Lookups	^				
ObjectDataSource vs. ControllerDataSource NORTHWIND							
Product: Su	pplier: okyo Traders 🖉						
ProductID	10						
ProductName	Ikura						
SupplierID	4						
SupplierCompanyName	Tokyo Traders						
CategoryID	8						
CategoryCategoryName	Seafood						
QuantityPerUnit	12 - 200 ml jars						
UnitPrice	31.0000						
UnitsInStock	31						
UnitsOnOrder	0						
ReorderLevel	0						
Discontinued							
Edit Delete New							
1234							
© 2009 MyCompany. All rights reserved.							
		Internet Protected Mode: On	€ 100% -				

CONCLUSION

Data Aquarium Framework does not stop with AJAX-enabled user interfaces. Any ASP.NET components supporting the *data source* architecture of Microsoft.NET will benefit from paging, sorting, and filtering of data sets of any size available in Data Aquarium application.

Business objects generated as a part of application are not mandatory and can be replaced with any external data access engine or library available to you.

Code OnTime LLC

http://www.codeontime.com