

Features of Master/Detail Presentation

Excellent master/detail support in [Data Aquarium Framework](#) comes with some great premium features, which typically require a lot of custom coding of high complexity.

Simple Markup

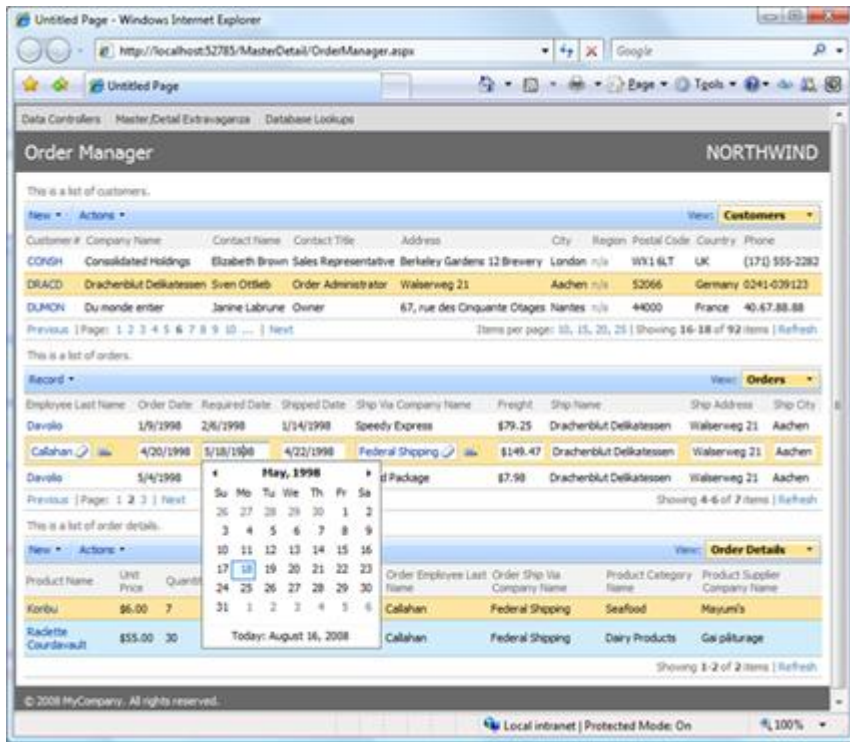
The standout feature is the simplicity of page definitions.

Consider a web form that displays *customers* and their *orders*. A considerable amount of markup is required to define a page like that if you are using standard *ASP.NET* components, such as *GridView*, *DetailsView*, and *ObjectDataSource*.

Here is a page fragment in a [Data Aquarium Framework](#) application that displays *customers* and *orders*:

```
<!-- presentation of customers -->
<div id="Customers" runat="server">
</div>
<aquarium:DataViewExtender ID="CustomersExtender" runat="server"
TargetControlID="Customers"
    Controller="Customers" PageSize="3" />
<!-- presentation of orders -->
<div id="Orders" runat="server">
</div>
<aquarium:DataViewExtender ID="OrdersExtender" runat="server"
TargetControlID="Orders"
    Controller="Orders" PageSize="3" FilterSource="CustomersExtender"
FilterFields="CustomerID" />
```

A couple of *div* elements and a couple of *DataViewExtender* components is all what is needed to render user interface like the one you can see live [here](#).



AJAX and server components of *Data Aquarium Framework* implement a centralized declarative user interface programming model. Your typical *ASP.NET* application is likely based on a page-centric model. Standard *ASP.NET* application pages host components such as *GridView* and *DetailsView* with their columns and fields predefined. The framework is relying on reusable *data controller descriptors* to get the metadata required to rendered user interface presentation. Follow the link to see a [sample data controller](#).

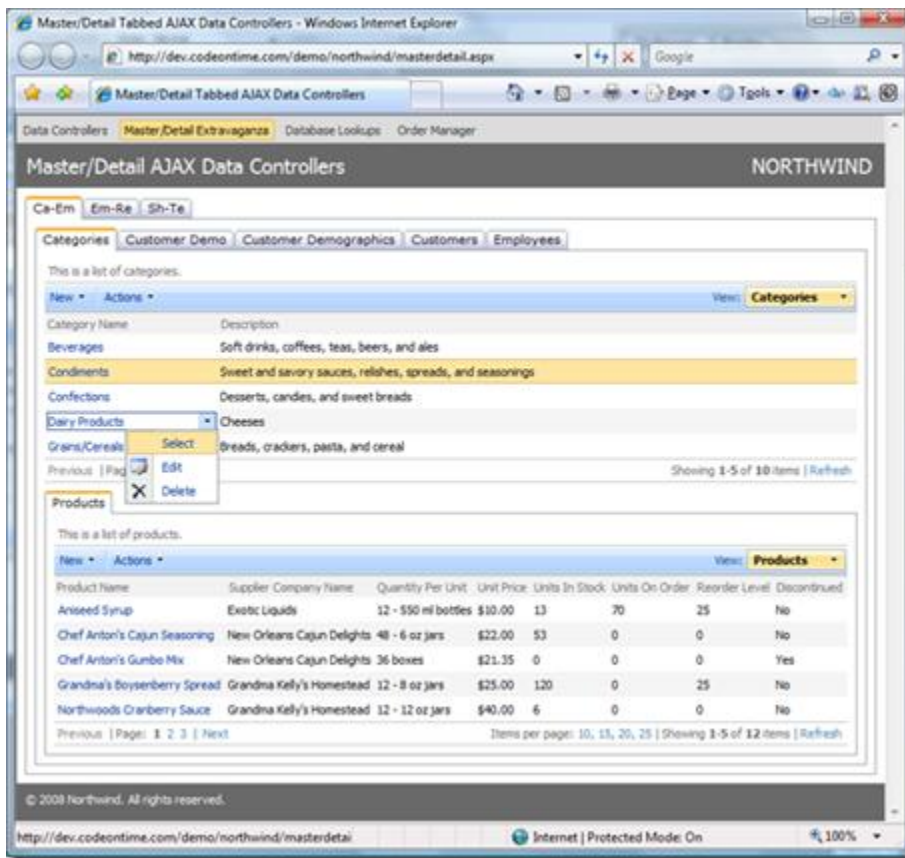
The centralized definition of user interface elements allows unmatched flexibility in user interface development. The same views defined once in a data controller can be referred in dozens of pages. These views are displayed automatically in dynamic lookups without any coding.

You can connect data controller views in all sorts of ways without writing any code. Simply set the *FilterSource* and *FilterFields* properties of detail *DataViewExtender* components to create complex master/detail relationships on a web form. Use the technique shown in the sample markup above.

Properties *FilterSource* and *FilterFields* are used to construct efficient dynamic *SQL* statements that are executed by your database server. *Data Aquarium Framework* packages the results of the queries into arrays of values and delivers them to the client components running on a page. *AJAX* components of the framework will render an *HTML* markup and replace fragments of the page to provide smooth user experience.

Load-on-Demand

Try the [sample](#) tabbed presentation of tables in the *Data Aquarium Framework* application created from the *Northwind* database.

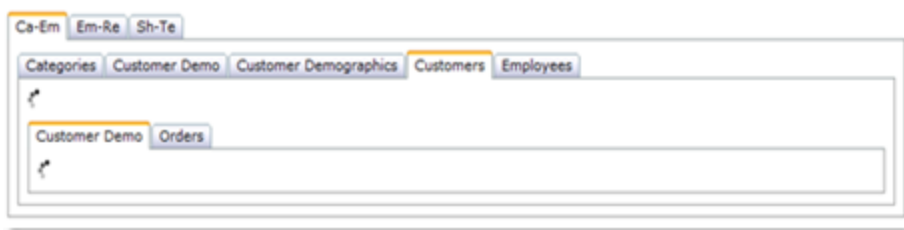


When you [open](#) the sample page for first time *Categories* and *Products* are dynamically loaded from the server and displayed. The act of opening a page is causing the server to render the tabbed user interface with *Ajax Control Toolkit* components *TabContainer* and *TabPanel* that you are seeing on the screen shot. The *Web.DataView AJAX* components of *Data Aquarium Framework* are embedded in the tabs. These

components are making two additional requests to get just enough data to present *customers* and *orders* on the form.

There are many more tabs with *Web.DataView* component instances matched to the rest of the *Northwind* database tables and connected in master/detail fashion. If all of them were executing data retrieval requests at the same time then that would have created dozens of additional server interactions. Many of this requests could be totally without purpose since an application user is likely not to look at all of them.

Click on the *Customers* tab and notice that the *customer* data has been requested on-demand and displayed shortly after you have click on the tab.



Data Aquarium Framework components automatically determine the exact amount of data that is needed to present in the user interface views that are actually visible to a user. Physical data retrieval happens when you select a tab and bring invisible views in focus.

Automatic Hiding of Filter Fields

If you change the markup of *DataViewExtender* components by removing *FilterSource* and *FilterFields* properties then the runtime screen will look similar to the one below if you select the first *customer* and sort orders by *customer company name*.

This is a list of customers.

Customer #	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	n/a	12209	Germany	030-0074321
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	n/a	05021	Mexico	(5) 555-4729
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	n/a	05023	Mexico	(5) 555-3932

Previous | Page: 1 2 3 4 5 6 7 8 9 10 ... | Next Items per page: 10, 15, 20, 25 | Showing 1-3 of 92 items | Refresh

This is a list of orders.

Customer Company Name	Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
Alfreds Futterkiste	Suyama	8/25/1997	9/22/1997	9/2/1997	Speedy Express	\$29.46	Alfreds Futterkiste	Obere Str. 57	Berlin
Alfreds Futterkiste	Peacock	10/3/1997	10/31/1997	10/13/1997	United Package	\$61.02	Alfred's Futterkiste	Obere Str. 57	Berlin
Alfreds Futterkiste	Peacock	10/13/1997	11/24/1997	10/21/1997	Speedy Express	\$23.94	Alfred's Futterkiste	Obere Str. 57	Berlin

Previous | Page: 1 2 3 4 5 6 7 8 9 10 ... | Next Items per page: 10, 15, 20, 25 | Showing 1-3 of 838 items | Refresh

Notice that the *orders* section is displaying a *customer company name* for each *order*. This is great if both views are completely independent but is not desirable if you have a master / detail presentation. All displayed *orders* are related to the same selected *customer* in that case. A repeated *customer name* on each *order* line takes up the valuable real estate of the page and should be eliminated.

Restore the *FilterSource* and *FilterFields* properties on *OrdersExtender* and notice that *orders* view automatically hides the *customer company name* without any extra coding.

This is a list of customers.

Customer #	Company Name	Contact Name	Contact Title	Address	City	Region	Postal Code	Country	Phone
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	n/a	12209	Germany	030-0074321
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	n/a	05021	Mexico	(5) 555-4729
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	n/a	05023	Mexico	(5) 555-3932

Previous | Page: 1 2 3 4 5 6 7 8 9 10 ... | Next Items per page: 10, 15, 20, 25 | Showing 1-3 of 92 items | Refresh

This is a list of orders.

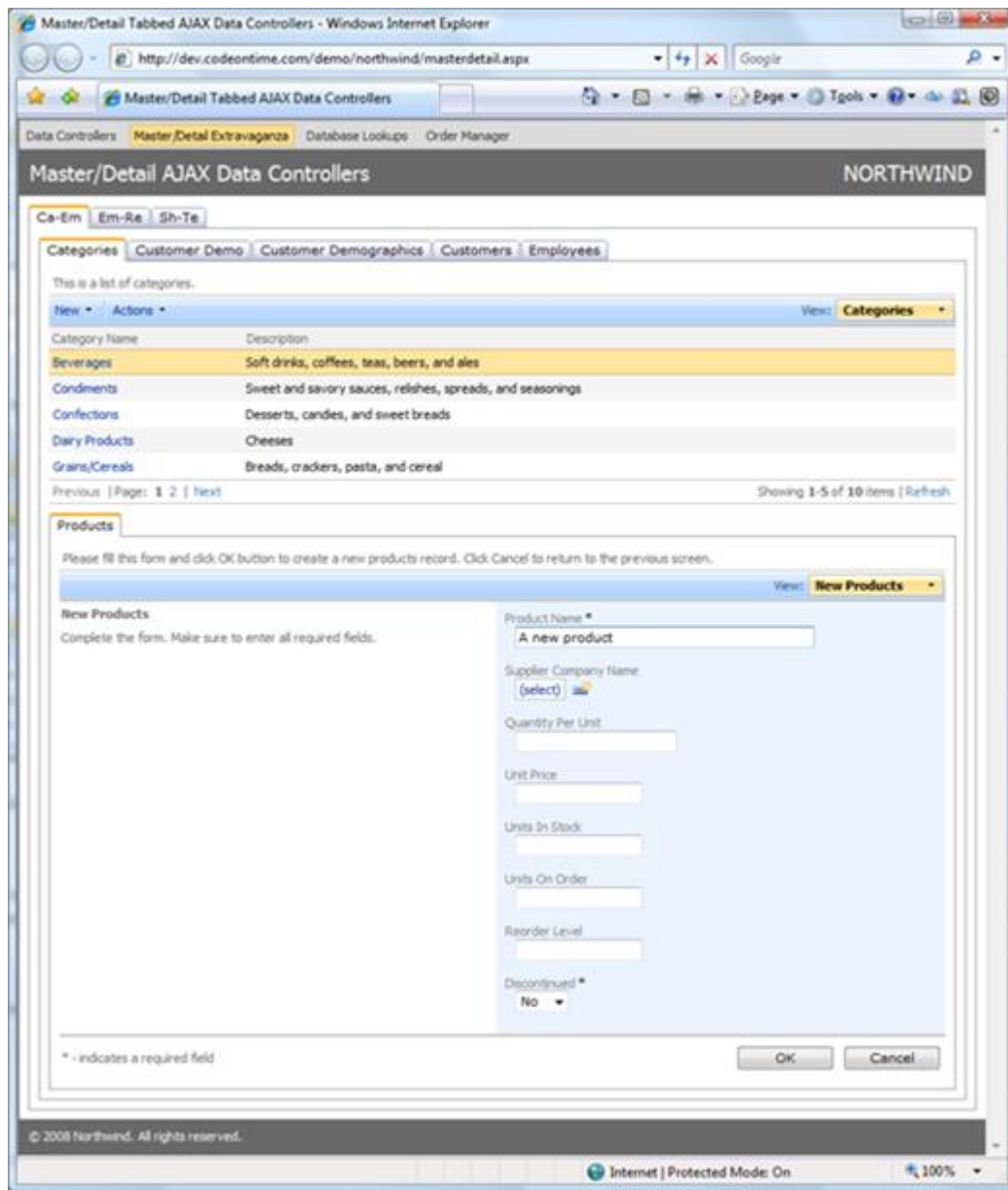
Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City
Suyama	8/25/1997	9/22/1997	9/2/1997	Speedy Express	\$29.46	Alfreds Futterkiste	Obere Str. 57	Berlin
Peacock	10/3/1997	10/31/1997	10/13/1997	United Package	\$61.02	Alfred's Futterkiste	Obere Str. 57	Berlin
Peacock	10/13/1997	11/24/1997	10/21/1997	Speedy Express	\$23.94	Alfred's Futterkiste	Obere Str. 57	Berlin

Previous | Page: 1 2 | Next Showing 1-3 of 6 items | Refresh

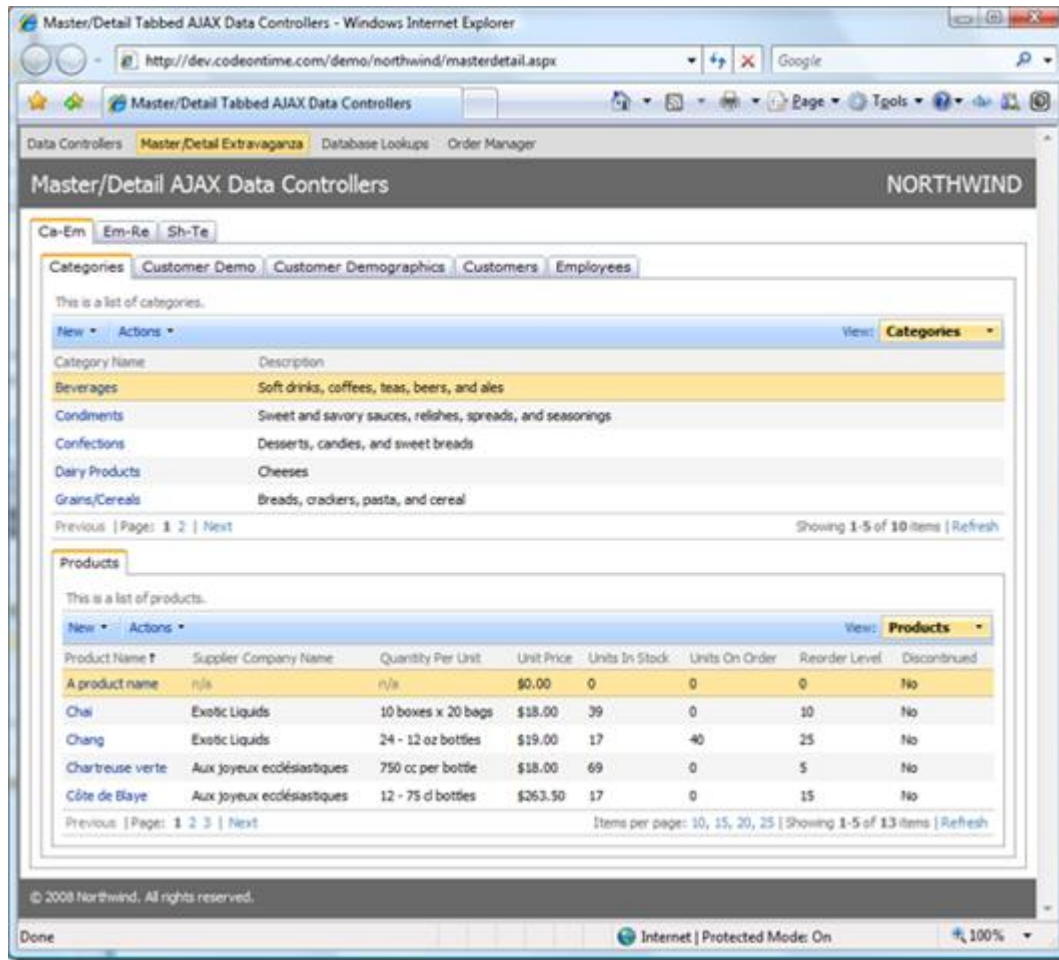
If you were to link *orders* to *employees* then the *employee last name* column would disappear.

Primary Key Inheritance

Navigate to the [sample application](#) and try to add a new product in any category. First select a *category*. Sort *products* by name and then click on *New* menu option on the action bar of the *products* view. Click on *New Products* menu item. Enter *A new product* and press OK button.



The following view will be presented next.



The new product is automatically linked to selected master category without any coding. The primary key field values of selected master record are automatically copied to the detail record foreign key fields when the detail record is inserted into database.

This feature is automatically enabled when you set a master/detail connection via *FilterSource* and *FilterFields* properties of *DataViewExtender*. You can create [custom action handlers](#) to provide additional data processing before and after the *SQL* command.

Support For Standard Data Components

There will be times when you need to write some highly custom master/detail web forms and would rather rely on standard *ASP.NET* components to do so. [Data Aquarium](#)

Framework provides *ControllerDataSource* data source component that works great with standard components or any other commercial library that you own.

Generate an application based on *Data Aquarium* project , open the generated code in *Visual Studio 2008* or *Visual Web Developer Express 2008* and and create a new page *StandardMD.aspx* based on the *MasterPage.master* in the web site root. Enter the following markup:

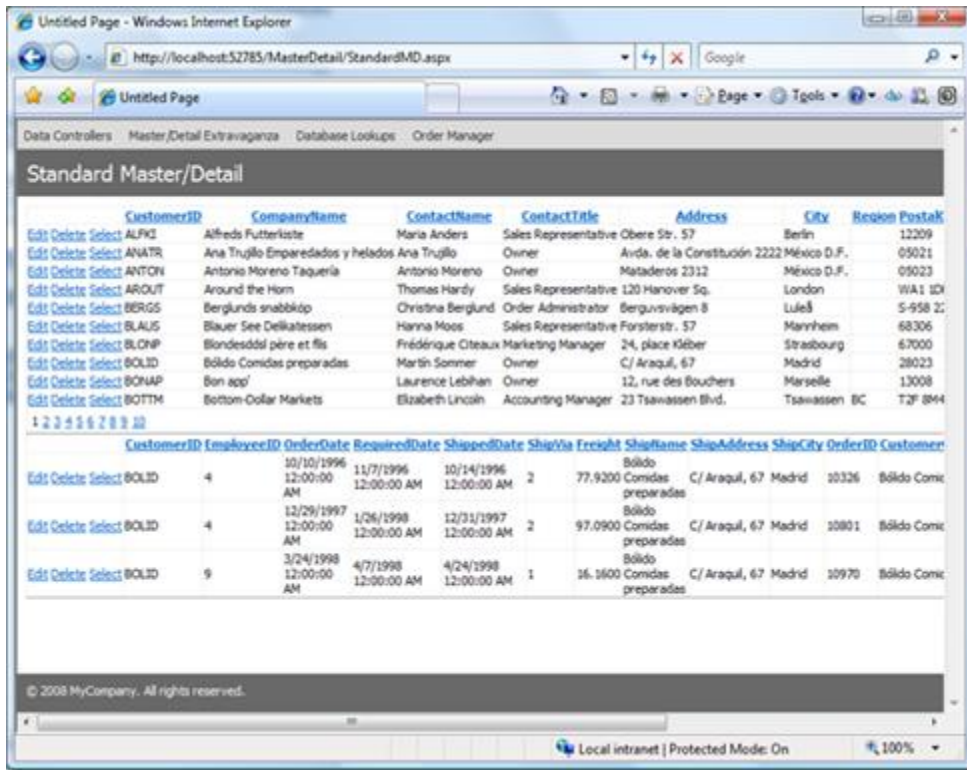
```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true"
    CodeFile="StandardMD.aspx.cs" Inherits="StandardMD" Title="Untitled Page"
%>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="Header1Placeholder"
runat="Server">
    Standard Master/Detail
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="Header2Placeholder"
runat="Server">
    Northwind
</asp:Content>
<asp:Content ID="Content4" ContentPlaceHolderID="BodyPlaceholder"
runat="Server">
    <aquarium:ControllerDataSource ID="CustomersCDS" runat="server"
DataController="Customers">
        </aquarium:ControllerDataSource>
        <asp:GridView ID="Customers" runat="server" DataSourceID="CustomersCDS"
AllowPaging="true"
            AllowSorting="true" AutoGenerateDeleteButton="true"
AutoGenerateEditButton="true"
            AutoGenerateSelectButton="true" DataKeyNames="CustomerID">
        </asp:GridView>
    </asp:Content>
</pre>
```



```
<aquarium:ControllerDataSource ID="OrdersCDS" runat="server"
DataController="Orders">
  <FilterParameters>
    <asp:ControlParameter Name="CustomerID" ControlID="Customers"
PropertyName="SelectedValue" />
  </FilterParameters>
</aquarium:ControllerDataSource>
<asp:GridView ID="Orders" runat="server" DataSourceID="OrdersCDS"
AllowPaging="true"
  AllowSorting="true" AutoGenerateDeleteButton="true"
AutoGenerateEditButton="true"
  AutoGenerateSelectButton="true" DataKeyNames="OrderID">
</asp:GridView>
</asp:Content>
```

The only unusual code here is the presence of *ControllerDataSource* instances. The standard *GridView* components on the page are bound to the data sources via *DataSourceID* property. Data source *OrdersCDS* is set to behave as a detail of the *Customers* grid view and will filter data whenever a *customer* is selected on the page. If you run the page and select a *customer* then the following user interface will be presented.



Nothing really fancy. But if you do look closer then you will notice that you can page and sort your records. You can edit, and delete anything on the page. The *orders* grid is automatically refreshed when a *customer* is selected.

In fact, you can have thousands of records in both grids and they will work extremely fast thanks to the on-demand data retrieval built into [Data Aquarium Framework](#). Only the data that needs to be presented on the page is actually retrieved from the database. The exact same code is being executed on the sever in response to requests by client-side *AJAX* components when they need to displays data. That sort of functionality is hard to accomplish unless you resort to use *ObjectDataSource* components and write custom business objects to support them.

With a little bit of editing you can have this page look like the one below.

The screenshot shows a web browser window displaying a Master/Detail web form. The browser title is "Untitled Page - Windows Internet Explorer" and the address bar shows "http://localhost:52785/MasterDetail/StandardMD.aspx". The page title is "Standard Master/Detail" and the database name is "NORTHWIND".

The top table, titled "Customer", has the following columns: Customer#, Company Name, Contact Name, Title, Address, City, Region, Postal Code, Country, and Phone. The rows are as follows:

	Customer#	Company Name	Contact Name	Title	Address	City	Region	Postal Code	Country	Phone
Edit Delete Select	ALFKJ	Alfreds Futterkiste	Maria Anders	Sales Representative	Oberstr. 57	Berlin		12209	Germany	030-0074321
Edit Delete Select	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.		05021	Mexico	(5) 555-4729
Edit Delete Select	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.		05023	Mexico	(5) 555-3932
Edit Delete Select	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	WA1	1DP	UK	(171) 555-7788
Edit Delete Select	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvavägen 8	Luleå		S-958 22	Sweden	0921-12 34 65
Edit Delete Select	BLAUS	Blauser See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Hannheim		68306	Germany	0621-08450
Edit Delete Select	BONAP	Bonaparte et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg		67000	France	88.60.15.31
Edit Delete Select	BOLID	Bólido Comidas preparadas	Marín Sommer	Owner	C/ Araquil, 67	Madrid		28023	Spain	(91) 555 22 82
Edit Delete Select	BONAP	Bon app'	Laurence Labihan	Owner	12, rue des Bouchers	Marseille		13008	France	91.24.45.40
Edit Delete Select	BOTTM	Bottom-ollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen BC	T2F 0M4		Canada	(604) 555-4729

Below the first table is a pagination control: "1 2 3 5 6 7 8 9 10".

The second table, titled "Order", has the following columns: Employee Last Name, Order Date, Required Date, Shipped Date, Ship Via Company Name, Freight, Ship Name, Ship Address, Ship City, and Order ID. The rows are as follows:

	Employee Last Name	Order Date	Required Date	Shipped Date	Ship Via Company Name	Freight	Ship Name	Ship Address	Ship City	Order ID
Edit Delete Select	Oodivorth	4/9/1997	5/7/1997	4/16/1997	Blauser See Delikatessen	\$8.85	Blauser See Delikatessen	Forsterstr. 57	Mannheim	10501
Edit Delete Select	Peacock	4/17/1997	5/15/1997	4/29/1997	Blauser See Delikatessen	\$0.15	Blauser See Delikatessen	Forsterstr. 57	Mannheim	10509
Edit Delete Select	Leverling	6/27/1997	7/25/1997	7/14/1997	Blauser See Delikatessen	\$27.71	Blauser See Delikatessen	Forsterstr. 57	Mannheim	10582
Edit Delete Select	Callahan	7/29/1997	8/26/1997	8/1/1997	Blauser See Delikatessen	\$1.93	Blauser See Delikatessen	Forsterstr. 57	Mannheim	10614
Edit Delete Select	Oodivorth	1/27/1998	2/24/1998	2/3/1998	Blauser See Delikatessen	\$53.83	Blauser See Delikatessen	Forsterstr. 57	Mannheim	10853
Edit Delete Select	Suyama	3/17/1998	4/28/1998	3/20/1998	Blauser See Delikatessen	\$44.65	Blauser See Delikatessen	Forsterstr. 57	Mannheim	10956
Edit Delete Select	Oodivorth	4/29/1998	5/27/1998		Blauser See Delikatessen	\$31.14	Blauser See Delikatessen	Forsterstr. 57	Mannheim	11058
Edit Delete Select	Buchanan				Blauser See Delikatessen	\$0.00				11099

At the bottom of the page, there is a copyright notice: "© 2008 MyCompany. All rights reserved." and a status bar: "Local intranet | Protected Mode: On | 100%".

To speed up customization select the data source instances of this sample page and execute *Refresh Schema* command in the smart tag options of each data source. This will automatically create fields for your grid views.

Remember that *ControllerDataSource* is working with any standard or custom components that are compatible with ASP.NET data binding architecture.

Conclusion

Data Aquarium Framework provides significant productivity features to developers building master / detail web forms. Modern AJAX-based user interface components of the framework will automatically handle many complicated issues that are commonly encountered by application developers. Standard ASP.NET web forms are supported as well via *ControllerDataSource* component.