

# LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Over the last few weeks I've been writing a series of blog posts that cover LINQ to SQL. LINQ to SQL is a built-in O/RM (object relational mapper) that ships in the .NET Framework 3.5 release, and which enables you to easily model relational databases using .NET classes. You can use LINQ expressions to query the database with them, as well as update/insert/delete data.

Below are the first four parts of my LINQ to SQL series:

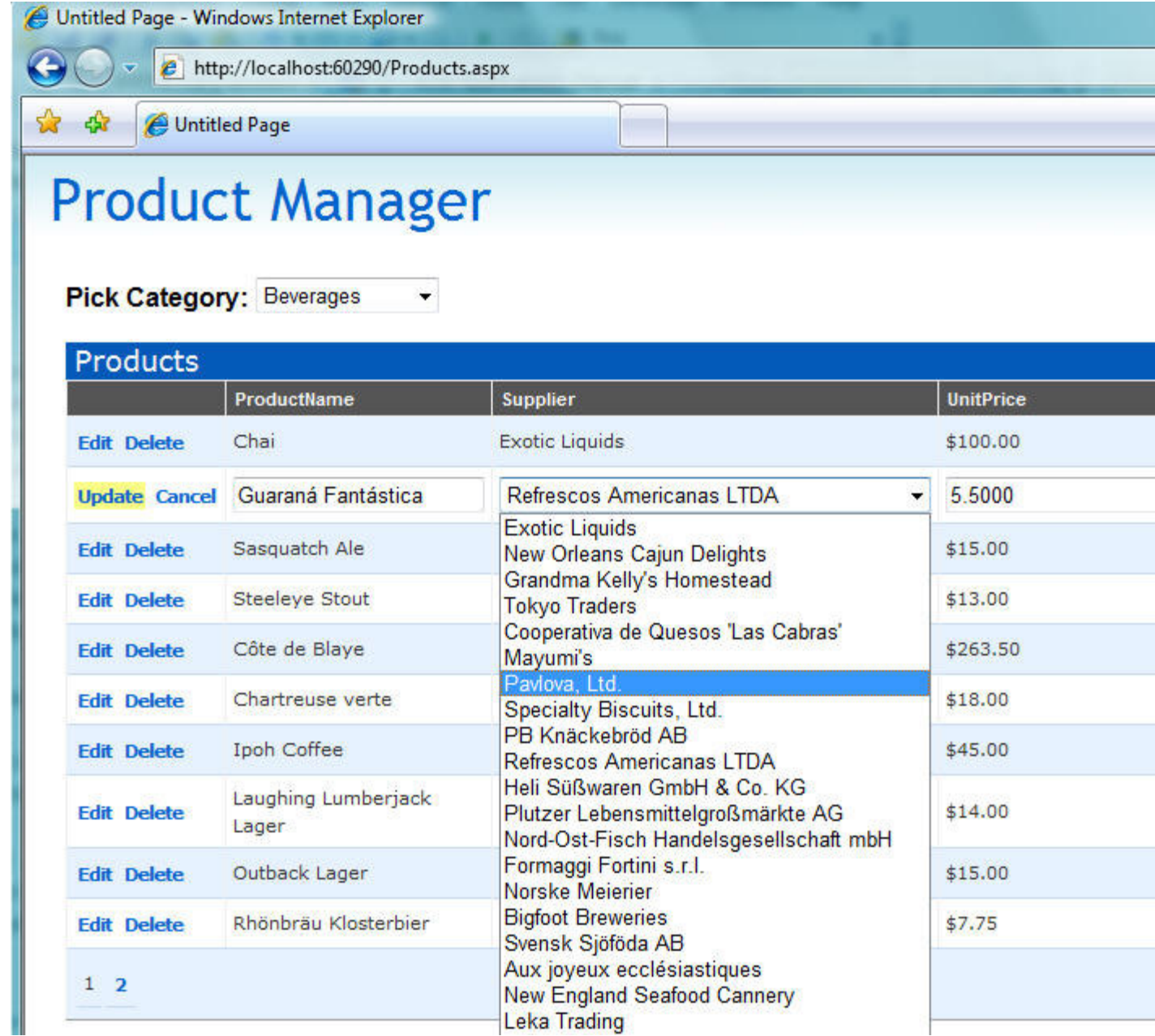
- [Part 1: Introduction to LINQ to SQL](#)
- [Part 2: Defining our Data Model Classes](#)
- [Part 3: Querying our Database](#)
- [Part 4: Updating our Database](#)

In these previous LINQ to SQL blog posts I focused on how you can programmatically use LINQ to SQL to easily query and update data within a database.

In today's blog post I'll cover the new <asp:LinqDataSource> control that is shipping as part of ASP.NET in the upcoming .NET 3.5 release. This control is a new datasource control for ASP.NET (like the ObjectDataSource and SqlDataSource controls that shipped with ASP.NET 2.0) which makes declaratively binding ASP.NET UI controls to LINQ to SQL data models super easy.

## Sample Application We'll be Building

The simple data editing web application I'll walkthrough building in this tutorial is a basic data entry/manipulation front-end for products within a database:



The application will support the following end-user features:

1. Allow users to filter the products by category
2. Allow users to sort the product listing by clicking on a column header (Name, Price, Units In Stock, etc)
3. Allow users to skip/page over multiple product listings (10 products per page)
4. Allow users to edit and update any of the product details in-line on the page
5. Allow users to delete products from the list

The web application will be implemented with a clean object-oriented data model built using the LINQ to SQL ORM.

All of the business rules and business validation logic will be implemented in our data model tier - and not within the UI tier or in any of the UI pages. This will ensure that: 1) a consistent set of business rules are used everywhere within the application, 2) we write less code and don't repeat ourselves, and 3) we can easily modify/adapt our business rules at a later date and not have to update them in dozens of different places across our application.

We will also take advantage of the built-in paging/sorting support within LINQ to SQL to ensure that features like the product listing paging/sorting are performed not in the middle-tier, but rather in the database (meaning only 10 products are retrieved from the database at any given time - we are not retrieving thousands of rows and doing the sorting/paging within the web-server).

## What is the <asp:LinqDataSource> control and how does it help?

The <asp:LinqDataSource> control is an ASP.NET control that implements the [DataSourceControl pattern](#) introduced with ASP.NET 2.0. It is similar to the ObjectDataSource and SqlDataSource controls in that it can be used to declaratively bind other ASP.NET controls on a page to a datasource. Where it differs is that instead of binding directly to a database (like the SqlDataSource) or to a generic class (like the ObjectDataSource), the <asp:linqdatasource> is designed to bind against a LINQ enabled data model.

One of the benefits of using the <asp:linqdatasource> control is that it leverages the flexibility that LINQ based ORMs provide. You don't need to define custom query/insert/update/delete methods for the datasource to call - instead you can point the <asp:linqdatasource> control at your data model, identify what entity table you want it to work against, and then bind any ASP.NET UI control against the <asp:linqdatasource> and have them work with it.

For example, to get a basic product listing UI on my page that works against Product entities within a LINQ to SQL data model, I could simply declare a <asp:linqdatasource> on my page that points to my LINQ to SQL datacontext class, and identify the entities (for example: Products) in the LINQ to SQL data model I want to bind against. I could then point a GridView at it (by settings its DataSourceID property) to get a grid-like view of the Product content:

```

<asp:GridView ID="GridView1" DataSourceID="SupplierDataSource"
              AllowPaging="true"
              AllowSorting="true"
              runat="server" />

<asp:LinqDataSource ID="SupplierDataSource"
                  ContextTypeName="WebApplication12.NorthwindDataContext"
                  TableName="Products"
                  EnableUpdate="true" EnableDelete="true"
                  runat="server" />

```

Without having to do anything else, I can run the page and have a listing of my Product data with built-in support for paging and sorting over the data. I can add a edit/delete button on the Grid and automatically have update support as well. I don't need to add any methods, map any parameters, or write any code for the <asp:LinqDataSource> to handle both these querying and updating scenarios - it can work against the LINQ to SQL data model we point it against and do these operations automatically. When updates are made, the LINQ to SQL ORM will automatically ensure that all business rules and validation logic we've added (as partial methods) to the LINQ to SQL data model pass before persisting anything to the database.

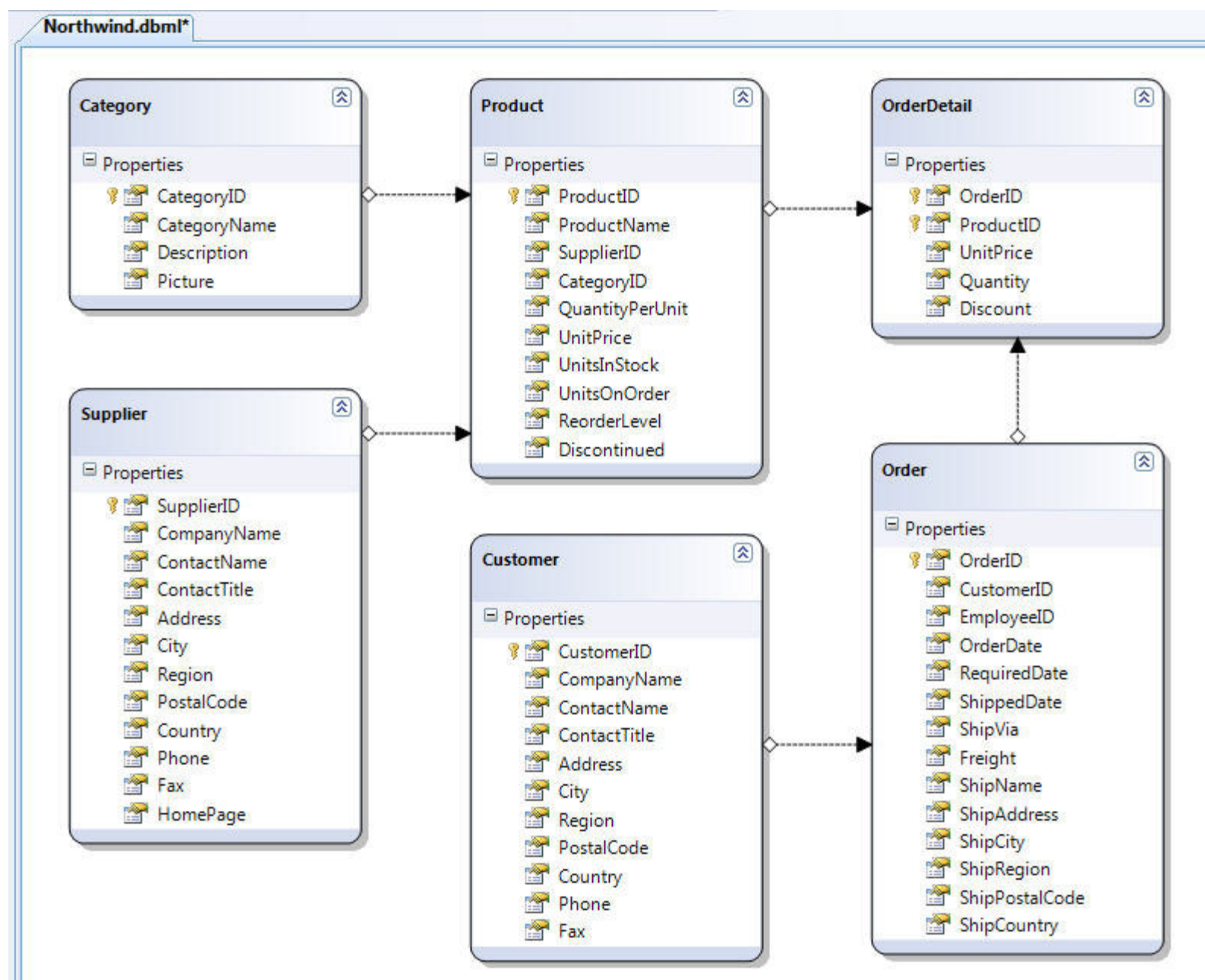
**Important:** The beauty of LINQ and LINQ to SQL is that it obviously isn't tied to being used only in UI scenarios - or with particular UI binding controls like the LinqDataSource. As you've seen in my previous posts in this series, writing code using the LINQ to SQL ORM is extremely clean. You can always write custom UI code to directly work against your LINQ to SQL data model if you prefer, or when you find a UI scenario that isn't particularly suited to using the <asp:linqdatasource>.

The below sections walkthrough using LINQ to SQL and the <asp:LinqDataSource> control to build the web application scenario I defined above.

### Step 1: Define our Data Model

We'll begin working on the application by first defining the data model we'll use to represent our database.

I discussed how to create a LINQ to SQL data model using VS 2008's LINQ to SQL designer in [Part 2](#) of this series. Below is a screenshot of the data model classes I can quickly create using the LINQ to SQL designer to model the "Northwind" sample database:



We'll revisit our data model in Step 5 of this tutorial below when we add some business validation rules to it. But to begin with we'll just use the above data model as-is to build our UI.

### Step 2: Creating a Basic Product Listing

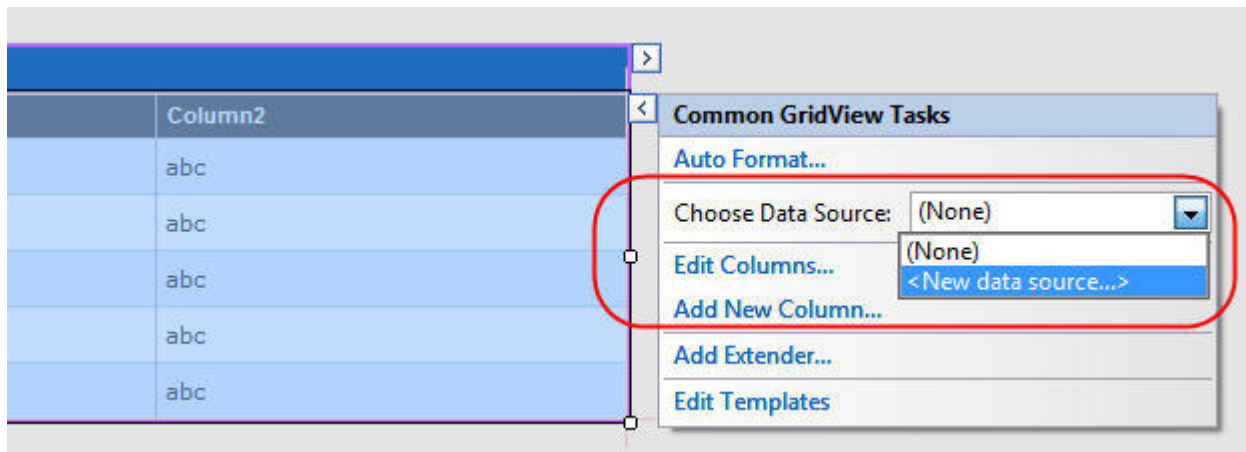
We'll start our UI by creating an ASP.NET page with a <asp:gridview> control on it and use some CSS to style it:



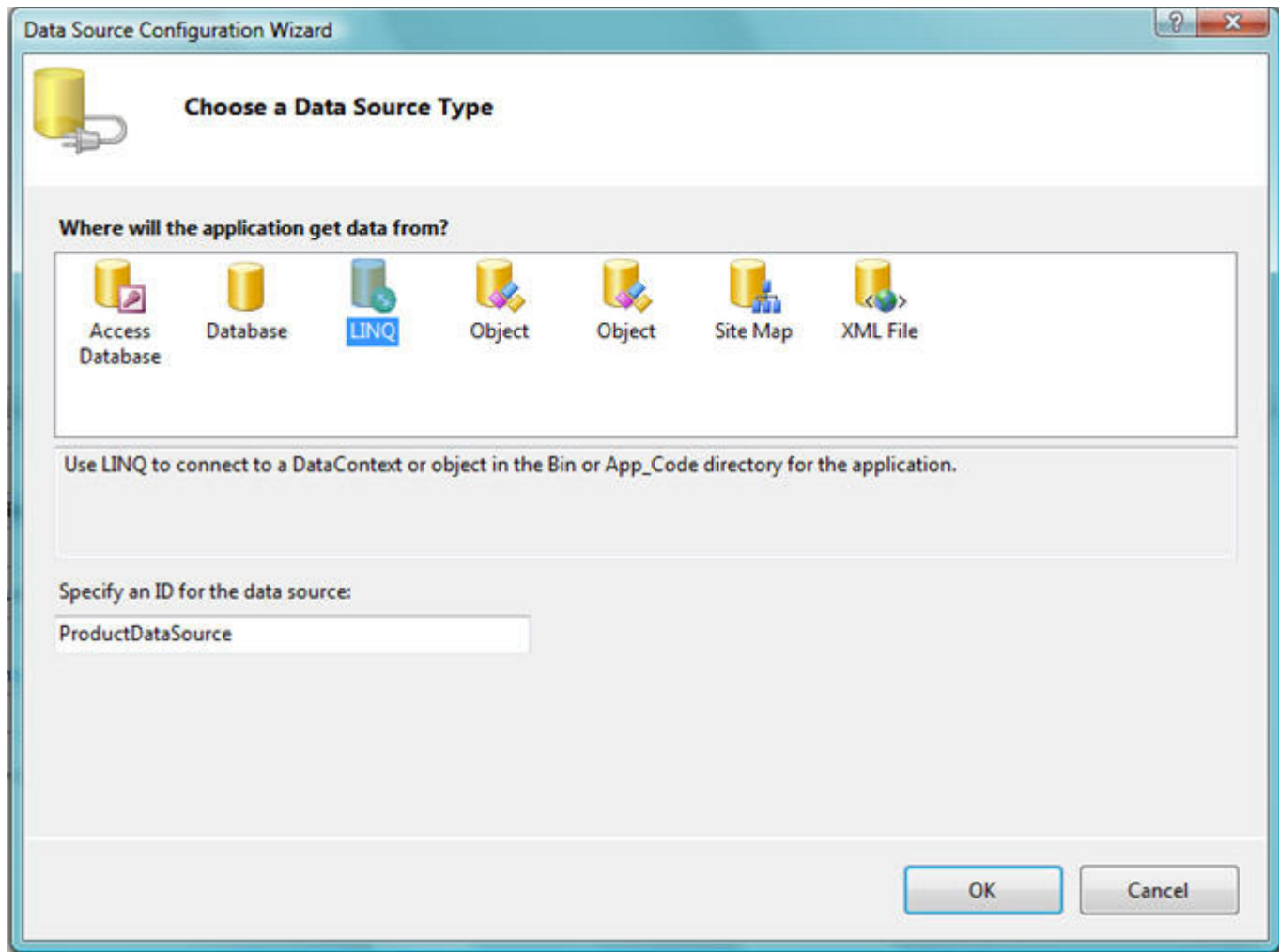


We could write code to programmatically bind our data model to the GridView (like I did in [Part 3](#) of this series), or alternatively I could use the new `<asp:linqdatasource>` control to bind the GridView to our data model.

VS 2008 includes build-in designer support to make it easy to connect up our GridView (or any other ASP.NET server control) to LINQ data. To bind our grid above to the data model we created earlier, we can switch into design-view, select the GridView, and then select the "New Data Source..." option within the "Choose Data Source:" drop-down:



This will bring up a dialog box that lists the available datasource options to create. Select the new "LINQ" option in the dialog box and name the resulting `<asp:linqdatasource>` control you want to create:



The `<asp:linqdatasource>` designer will then display the available LINQ to SQL DataContext classes that your application can use (including those in class libraries that you are referencing):

Configure Data Source - ProductDataSource

Choose a Context Object

Select a context object that can be used to retrieve or update data.

☒ Show only DataContext objects

Choose your context object:

WebApplication12.NorthwindDataContext

< Previous

Next >

Finish

Cancel

We'll want to select the data model we created with the LINQ to SQL designer earlier. We'll then want to select the table within our data model that we want to be the primary entity for the <asp:linqdatasource> to bind against. For this scenario we'll want to select the "Products" entity class we built. We'll also want to select the "Advanced" button and enable updates and deletes for the datasource:

Configure Data Source - ProductDataSource

Configure Data Selection

Table:

Products (Table<Product>)

Group by:

[None]

Select:

☒ \*

☐ ProductID

☐ ProductName

☐ SupplierID

☐ CategoryID

☐ QuantityPerUnit

☐ UnitPrice

☐ UnitsInStock

☐ UnitsOnOrder

☐ ReorderLevel

☐ Discontinued

☐ OrderDetails

☐ Category

Where...

Order By...

Advanced...

< Previous

Next >

Finish

Cancel

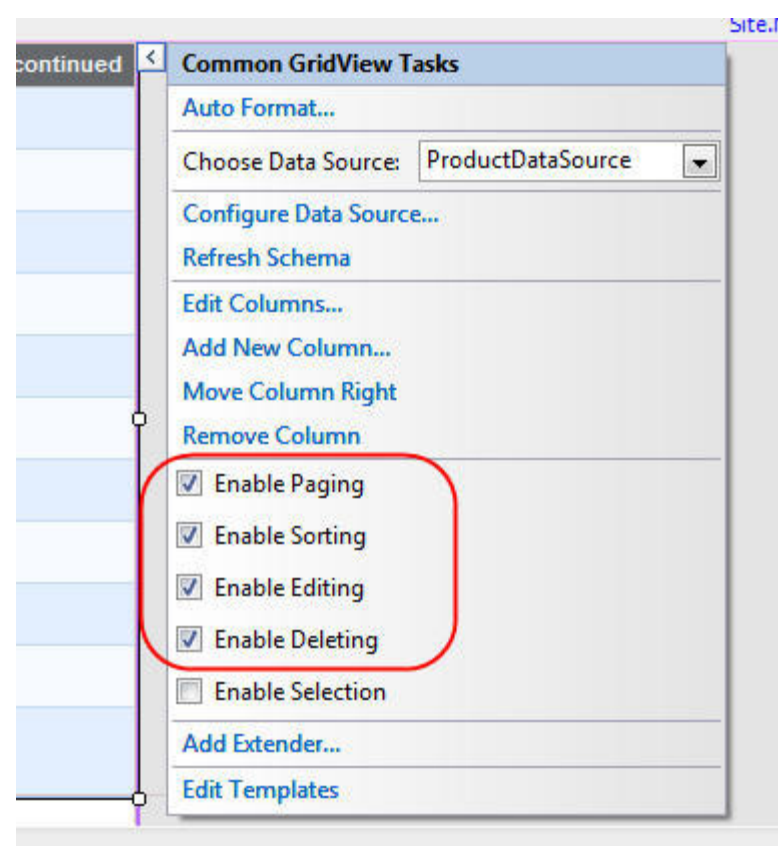
When we click the "Finish" button above, VS 2008 will declare a <asp:linqdatasource> within our .aspx page, and update the <asp:gridview> to point to it (via its DataSourceID property). It will also automatically provide column declarations in the Grid based on the schema of the Product entity we choose to bind against:

ContentPlaceholder1 (Custom)

Products								
ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsIn Stock	UnitsOnOrder	Discontinued
0	abc	0	0	abc	0	0	0	<input type="checkbox"/>
1	abc	1	1	abc	0.1	1	1	<input checked="" type="checkbox"/>
2	abc	2	2	abc	0.2	2	2	<input type="checkbox"/>
3	abc	3	3	abc	0.3	3	3	<input checked="" type="checkbox"/>
4	abc	4	4	abc	0.4	4	4	<input type="checkbox"/>

LinqDataSource - ProductDataSource

We can then pull up the "smart task" context UI of the GridView and indicate that we want to enable paging, sorting, editing and deleting on it:



We can then press F5 to run our application, and have a product listing page with full paging and sorting support (note the paging indexes at the bottom of the grid below):

Products								
	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	Discontinued
Edit Delete	Chai	1	1	10 boxes x 20 bags	100.0000	39	0	<input type="checkbox"/>
Edit Delete	Chang	1	1	24 - 12 oz bottles	22.0000	16	41	<input type="checkbox"/>
Edit Delete	Aniseed Syrup	1	2	12 - 550 ml bottles	10.0000	13	70	<input type="checkbox"/>
Edit Delete	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	23.0000	53	0	<input type="checkbox"/>
Edit Delete	Chef Anton's Gumbo Mix	2	2	36 boxes	21.3500	0	0	<input checked="" type="checkbox"/>
Edit Delete	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25.0000	120	0	<input type="checkbox"/>
Edit Delete	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30.0000	15	0	<input type="checkbox"/>
Edit Delete	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40.0000	6	0	<input type="checkbox"/>
Edit Delete	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97.0000	29	0	<input checked="" type="checkbox"/>
Edit Delete	Ikura	4	8	12 - 200 ml jars	31.0000	31	0	<input type="checkbox"/>
1 2 3 4 5 6 7 8								

We can also select the "edit" or "delete" button on each row to update the data:

Products				
	ProductName	SupplierID	CategoryID	QuantityPerUnit
Edit Delete	Chai	1	1	10 boxes x 20 bags
Update Cancel	<input type="text" value="Chang"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	24 - 12 oz bottles
Edit Delete	Aniseed Syrup	1	2	12 - 550 ml bottles

If we flip into source view on the page, we'll see that the markup of the page contains the content below. The `<asp:linqdatasource>` control points at the LINQ to SQL DataContext we created earlier, as well as the entity table we want to bind against. The GridView then points at the `<asp:linqdatasource>` control (via its `DataSourceID`) and indicates which columns should be included in the grid, what their header text should be, as well as what sort expression to use when the column header is selected.



```

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

    <div class="productsheader">
        <h2>Products</h2>
    </div>

    <asp:GridView ID="GridView1" DataSourceID="ProductDataSource" CssClass="gridview" Alternati
        <columns>
            <asp:commandfield ShowDeleteButton="True" ShowEditButton="True"></asp:commandfield>
            <asp:boundfield DataField="ProductName" HeaderText="ProductName" SortExpression="Pr
            <asp:boundfield DataField="SupplierID" HeaderText="SupplierID" SortExpression="Supp
            <asp:boundfield DataField="CategoryID" HeaderText="CategoryID" SortExpression="Cate
            <asp:boundfield DataField="QuantityPerUnit" HeaderText="QuantityPerUnit" SortExpres
            <asp:boundfield DataField="UnitPrice" HeaderText="UnitPrice" SortExpression="UnitPr
            <asp:boundfield DataField="UnitsInStock" HeaderText="UnitsInStock" SortExpression="
            <asp:boundfield DataField="UnitsOnOrder" HeaderText="UnitsOnOrder" SortExpression="
            <asp:checkboxfield DataField="Discontinued" HeaderText="Discontinued" SortExpressio
        </columns>
    </asp:GridView>

    <asp:LinqDataSource ID="ProductDataSource"
        ContextTypeName="WebApplication12.NorthwindDataContext"
        TableName="Products"
        EnableDelete="True" EnableUpdate="True"
        runat="server" />

</asp:Content>

```

Now that we have the basics of our web UI working against our LINQ to SQL data-model, we can go ahead and further customize the UI and behavior.

### Step 3: Cleaning up our Columns

Our GridView above has a lot of columns defined within it, and two of the column values (the SupplierID and the CategoryID) are currently foreign-key numbers -- which certainly isn't the ideal way to represent them to an end-user.

#### Removing Unnecessary Columns

We can start cleaning up our UI by deleting a few of the columns we don't need. I can do this in source mode (simply nuke the <asp:boundfield> declarations) or in designer mode (just click on the column in the designer and choose the "Remove" task). For example, we could remove the "QuantityPerUnit" column below and re-run our application to get this slightly cleaner UI:

Products							
	ProductName	SupplierID	CategoryID	UnitPrice	UnitsInStock	UnitsOnOrder	Discontinued
Edit Delete	Chai	1	1	100.0000	39	0	<input type="checkbox"/>
Edit Delete	Chang	1	1	22.0000	16	41	<input type="checkbox"/>
Edit Delete	Aniseed Syrup	1	2	10.0000	13	70	<input type="checkbox"/>
Edit Delete	Chef Anton's Cajun Seasoning	2	2	23.0000	53	0	<input type="checkbox"/>
Edit Delete	Chef Anton's Gumbo Mix	2	2	21.3500	0	0	<input checked="" type="checkbox"/>
Edit Delete	Grandma's Boysenberry Spread	3	2	25.0000	120	0	<input type="checkbox"/>
Edit Delete	Uncle Bob's Organic Dried Pears	3	7	30.0000	15	0	<input type="checkbox"/>
Edit Delete	Northwoods Cranberry Sauce	3	2	40.0000	6	0	<input type="checkbox"/>
Edit Delete	Mishi Kobe Niku	4	6	97.0000	29	0	<input checked="" type="checkbox"/>
Edit Delete	Ikura	4	8	31.0000	31	0	<input type="checkbox"/>
1 2 3 4 5 6 7 8							

If you have used the <asp:ObjectDataSource> control before and explicitly passed update parameters to update methods (the default when using DataSet based TableAdapters) one of the things you know can be painful is that you have to change the method signatures of your TableAdapter's update methods when the parameters based by your UI are modified. For example: if we deleted a column in our grid (like above), we'd end up having to modify our TableAdapter to support update methods without that parameter.

One of the really nice things about the <asp:LinqDataSource> control is that you do not need to-do these types of changes. Simply delete (or add) a column from your UI and re-run the application - no other changes are required. This makes changing web UI built using the <asp:LinqDataSource> much easier, and enables much faster scenarios iterations within an application.

#### Cleaning up the SupplierID and CategoryID Columns

Currently we are displaying the foreign-key integer values in our GridView for the Supplier and Category of each Product:

Products							
	ProductName	SupplierID	CategoryID	UnitPrice	UnitsInStock	UnitsOnOrder	Discontinued
Edit Delete	Chai	1	1	100.0000	39	0	<input type="checkbox"/>
Edit Delete	Chang	1	1	22.0000	16	41	<input type="checkbox"/>

While accurate from a data model perspective, it isn't very end-user friendly. What I really want to-do is to display the CategoryName and SupplierName instead, and provide a drop-downlist while in Edit mode to enable end-users to easily associate the SupplierID and CategoryID values.



I can change the GridView to display the Supplier Name and Category Name instead of the ID's by replacing the default <asp:BoundField> in our GridView with an <asp:TemplateField>. Within this TemplateField I can add any content I want to customize the look of the column.

In the source code below I'm going to take advantage of the fact that each Product class in the LINQ to SQL data model we created has a Supplier and Category property on it. What this means is that I can easily databind their Supplier.CompanyName and Category.CategoryName sub-properties within our Grid:

```
<asp:GridView ID="GridView1" DataSourceID="ProductDataSource" CssClass="gridview" Alternati
  <columns>
    <asp:commandfield ShowDeleteButton="True" ShowEditButton="True"></asp:commandfield>
    <asp:boundfield DataField="ProductName" HeaderText="ProductName" SortExpression="Pro

    <asp:TemplateField HeaderText="Supplier" SortExpression="Supplier.CompanyName">
      <ItemTemplate>
        <%#Eval("Supplier.CompanyName")%>
      </ItemTemplate>
    </asp:TemplateField>

    <asp:TemplateField HeaderText="Category" SortExpression="Category.CategoryName">
      <ItemTemplate>
        <%#Eval("Category.CategoryName") %>
      </ItemTemplate>
    </asp:TemplateField>

    <asp:boundfield DataField="UnitPrice" HeaderText="UnitPrice" SortExpression="UnitPri
    <asp:boundfield DataField="UnitsInStock" HeaderText="UnitsInStock" SortExpression="U
    <asp:boundfield DataField="UnitsOnOrder" HeaderText="UnitsOnOrder" SortExpression="U
    <asp:checkboxfield DataField="Discontinued" HeaderText="Discontinued" SortExpression

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

And now when I run the application I get the human readable Category and Supplier name values instead:

Products							
	ProductName	Supplier	Category	UnitPrice	UnitsInStock	UnitsOnOrder	Discontinued
Edit Delete	Côte de Blaye	Aux joyeux ecclésiastiques	Beverages	263.5000	17	0	<input type="checkbox"/>
Edit Delete	Chartreuse verte	Aux joyeux ecclésiastiques	Beverages	18.0000	69	0	<input type="checkbox"/>
Edit Delete	Sasquatch Ale	Bigfoot Breweries	Beverages	15.0000	111	0	<input type="checkbox"/>
Edit Delete	Steeleye Stout	Bigfoot Breweries	Beverages	13.0000	20	0	<input type="checkbox"/>
Edit Delete	Laughing Lumberjack Lager	Bigfoot Breweries	Beverages	14.0000	52	0	<input type="checkbox"/>
Edit Delete	Queso Cabrales	Cooperativa de Quesos 'Las Cabras'	Dairy Products	22.0000	22	30	<input type="checkbox"/>
Edit Delete	Queso Manchego La Pastora	Cooperativa de Quesos 'Las Cabras'	Dairy Products	38.0000	86	0	<input type="checkbox"/>
Edit Delete	Escargots de Bourgogne	Escargots Nouveaux	Seafood	13.2500	62	0	<input type="checkbox"/>
Edit Delete	Aniseed Syrup	Exotic Liquids	Condiments	10.0000	13	70	<input type="checkbox"/>
Edit Delete	Chang	Exotic Liquids	Beverages	22.0000	16	41	<input type="checkbox"/>
1 2 3 4 5 6 7 8							

To get drop-down list UI for the Supplier and Category columns while in Edit-Mode in the Grid, I will first add two additional <asp:LinqDataSource> controls to my page. I will configure these to bind against the Categories and Suppliers within the LINQ to SQL data model we created earlier:

```
<asp:LinqDataSource ID="CategoryDataSource"
  ContextTypeName="WebApplication12.NorthwindDataContext"
  TableName="Categories"
  runat="server" />

<asp:LinqDataSource ID="SupplierDataSource"
  ContextTypeName="WebApplication12.NorthwindDataContext"
  TableName="Suppliers"
  runat="server" />
```

I can then go back to the <asp:TemplateField> columns we added to our GridView earlier and customize their edit appearance (by specifying an EditItemTemplate). We'll customize each column to have a dropdownlist control when in edit mode, where the available values in the dropdownlists are pulled from the categories and suppliers datasource controls above, and where we two-way databind the selected value to the Product's SupplierID and CategoryID foreign keys:



```
<asp:TemplateField HeaderText="Supplier" SortExpression="Supplier.CompanyName">

    <ItemTemplate>
        <%#Eval("Supplier.CompanyName")%>
    </ItemTemplate>

    <EditItemTemplate>

        <asp:DropDownList ID="DropDownList1"
            DataSourceID="SupplierDataSource"
            DataValueField="SupplierID"
            DataTextField="CompanyName"
            SelectedValue='<%#Bind("SupplierID") %>'
            runat="server" />

    </EditItemTemplate>
</asp:TemplateField>
```

And now when end-users click edit in the GridView, they are presented a drop-down list of all valid Supplier's to associate the product with:

Products				
	ProductName	Supplier	Category	UnitP
Edit Delete	Chai	Exotic Liquids	Beverages	100.0
Update Cancel	Chang	Exotic Liquids	Beverages	22.0
Edit Delete	Aniseed Syrup	Exotic Liquids New Orleans Cajun Delights Grandma Kelly's Homestead	Condiments	10.00
Edit Delete	Chef Anton's Cajun Seasoning	Tokyo Traders Cooperativa de Quesos 'Las Cabras'	Condiments	23.00
Edit Delete	Chef Anton's Gumbo Mix	Mayumi's Pavlova, Ltd.	Condiments	21.35

And when they hit save the Product is updated appropriately (the GridView will use the DropDownList's currently selected value to bind the SupplierID).

Step 4: Filtering our Product Listing

Rather than show all products within the database, we can update our UI to include a dropdownlist that allows the user to filter the products by a particular category. Because we already added a <asp:LinqDataSource> control to the page earlier that references our Categories within our LINQ to SQL data model, all I need to-do to create a drop-downlist control at the top of the page that binds against this. For example:

```
<div class="category">

    Pick Category:

    <asp:DropDownList ID="CategoryList"
        DataSourceID="CategoryDataSource"
        DataTextField="CategoryName"
        DataValueField="CategoryID"
        AutoPostBack="True"
        runat="server">

    </asp:DropDownList>

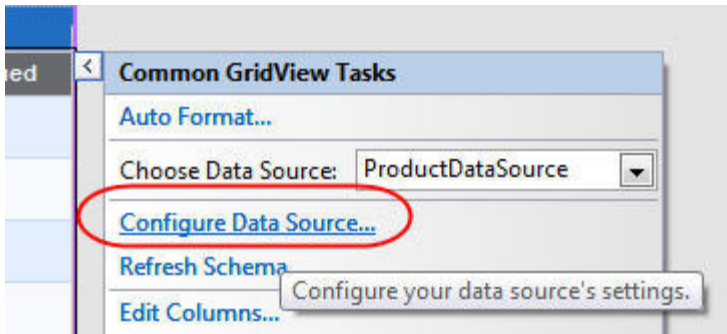
</div>
```

When I run the page I'll now get a filter dropdownlist of all categories at the top of the page:

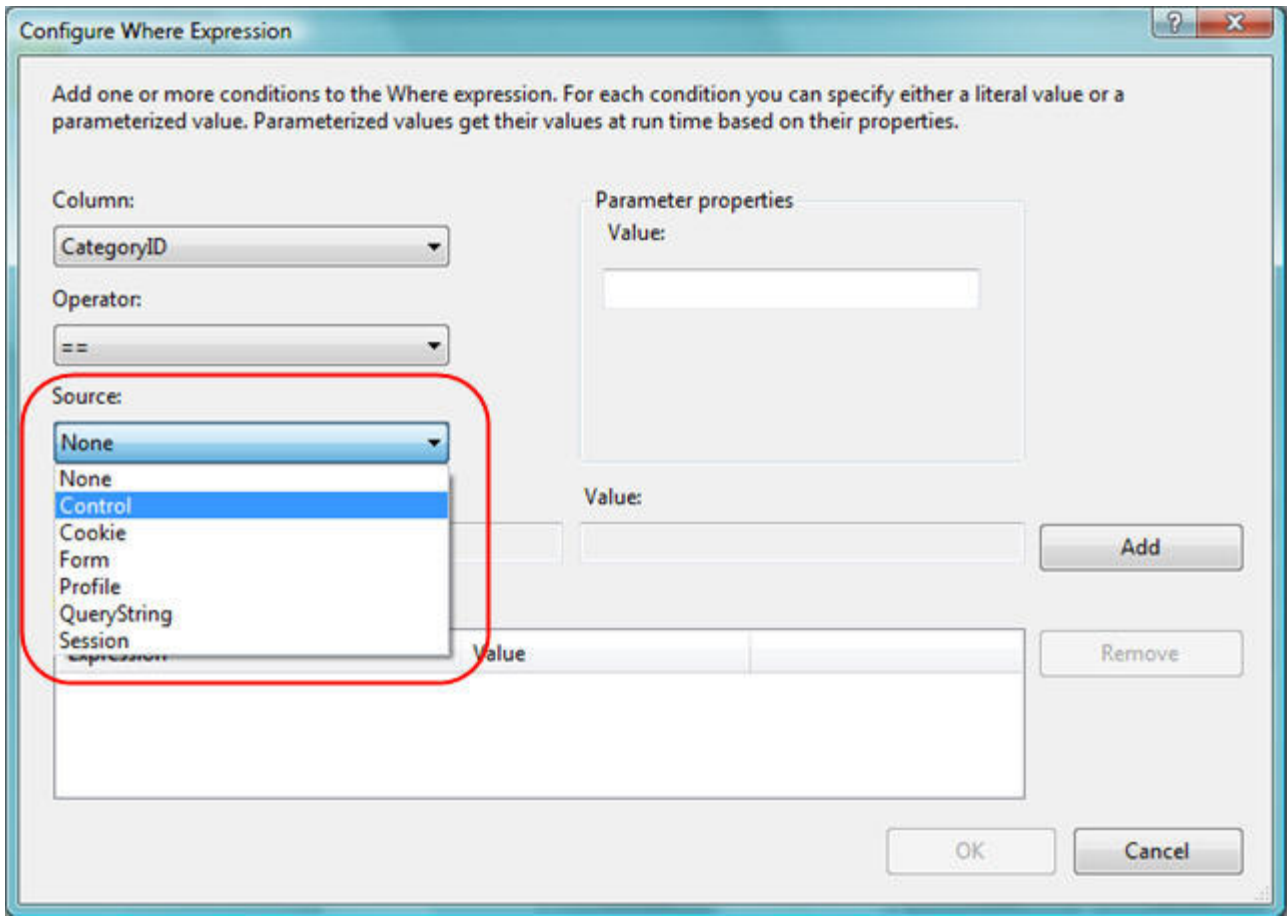


My last step is to configure the GridView to only show those Products in the category the end-user selects from the dropdownlist. The easiest way to-do this is by selecting the "Configure DataSource" option in the GridView smart task:

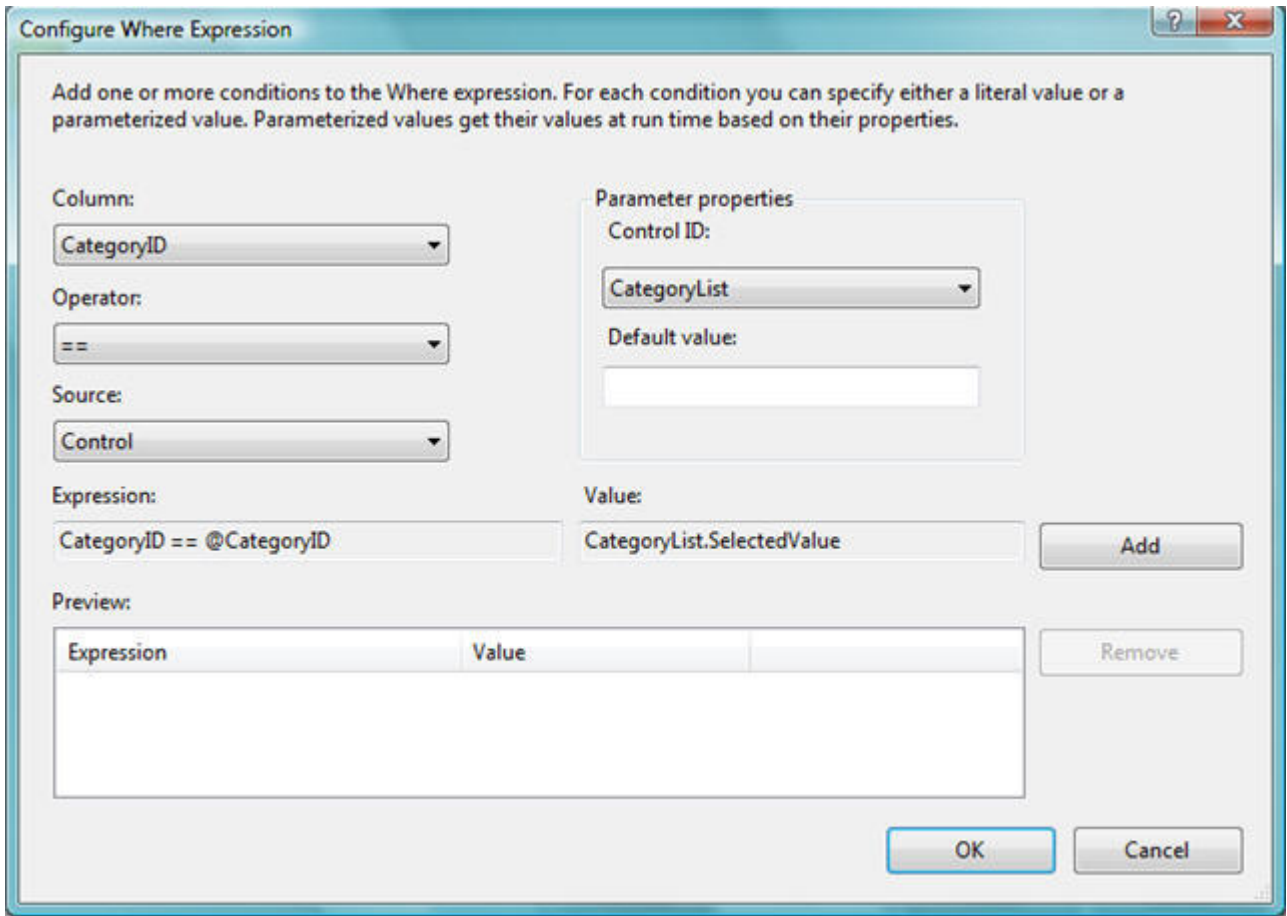




This will bring me back to the `<asp:LinqDataSource>` control's design-time UI that we used at the very beginning of this tutorial. I can select the "Where" button within this to add a binding filter to the datasource control. I can add any number of filter expressions, and declaratively pull the values to filter by from a variety of places (for example: from the querystring, from form-values, from other controls on the page, etc):



Above I'm going to choose to filter by the Products by their CategoryID value, and then retrieve this CategoryID from the DropDownList control we just created on our page:



When we hit finish, the `<asp:linqdatasource>` control in our page will have been updated to reflect this filter clause like so:

```
<asp:LinqDataSource ID="ProductDataSource" ContextTypeName="WebApplication12.NorthwindDataContext" Tail
    <whereparameters>
        <asp:controlparameter ControlID="CategoryList" Name="CategoryID" PropertyName="SelectedValue"
    </whereparameters>
</asp:LinqDataSource>
```

And when we now run the page the end-user will now be able to select from the available Categories in the filter drop-downlist and page, sort, edit and delete just the products in that category:

Pick Category: Confections ▼

Products				
	ProductName	Supplier	Category	UnitPrice
Edit Delete	Valkoinen suklaa	Karkki Oy	Confections	16.25
Update Cancel	<input type="text" value="Tarte au sucre"/>	<input style="border: 1px solid #ccc;" type="text" value="Forêts d'érables"/>	<input style="border: 1px solid #ccc;" type="text" value="Confections"/>	<input style="border: 1px solid #ccc;" type="text" value="49.30"/>
Edit Delete	Scottish Longbreads	Specialty Biscuits, Ltd.	Confections	12.50
1 2				

The <asp:LinqDataSource> control will automatically apply the appropriate LINQ filter expression when working against our LINQ to SQL data model classes to ensure that only the required data is retrieved from the database (for example: in the Grid above only the 3 rows of Product data from the second page of Confection products will be retrieved from the database).

You can optionally handle the Selecting event on the <asp:LinqDataSource> if you want to write a custom LINQ expression in code to completely customize the query instead.

### Step 5: Adding Business Validation Rules

As I discussed in [Part 4](#) of this LINQ to SQL series, when we define LINQ to SQL data models we will automatically have a default set of schema based validation constraints added to our data model classes. This means that if I try and enter a null value for a required column, try and assign a string to an integer, or assign a foreign-key value to a row that doesn't exist, our LINQ to SQL data model will raise an error and ensure that our database integrity is maintained.

Basic schema validation is only a first step, though, and is rarely enough for most real-world applications. Typically we'll want/need to add additional business rules and application-level validation to our data model classes. Thankfully LINQ to SQL makes adding these types of business validation rules easy (for details on the various validation approaches available, please read [Part 4](#) of my LINQ to SQL series).

#### Example Business Validation Rule Scenario

For example, let's consider a basic business logic rule we might want to enforce. Specifically, we want to ensure that a user of our application can't discontinue a product while we still have units on backorder for it:

Category	UnitPrice	UnitsInStock	UnitsOnOrder	Discontinued
Beverages	\$10.00	39	0	<input type="checkbox"/>
Beverages	\$5.50	20	44	<input type="checkbox"/>
Beverages	\$15.00	111	0	<input type="checkbox"/>
<input type="text" value="Beverages"/>	<input type="text" value="13.0000"/>	<input type="text" value="20"/>	<input type="text" value="23"/>	<input checked="" type="checkbox"/>
Beverages	\$263.50	17	0	<input type="checkbox"/>
Beverages	\$18.00	69	0	<input type="checkbox"/>

If a user tries to save the above row, we'll want to prevent this change from being persisted and throw an appropriate error telling the user how to fix it.

#### Adding a Data Model Validation Rule

The *wrong place* to add this type of business validation rule is in the UI layer of our application. Adding it in the UI layer of our application will mean that the rule will be specific to only that one place, and will not be automatically enforced when we add another page to our application that also updates Products. Distributing business rules/logic in our UI layer will also make life extremely painful as our application grows in size - since changes/updates to our business will necessitate making code changes all over the place.

The right place to specify this type of business logic validation is instead in our LINQ to SQL data model classes that we defined earlier. As I discussed in [Part 4](#) of this series, all classes generated by the LINQ to SQL designer are defined as "partial" classes - which means that we can easily add additional methods/events/properties to them. The LINQ to SQL data model classes automatically call validation methods that we can implement to enforce custom validation logic within them.

For example, I could add a partial Product class to my project that implements the OnValidate() partial method that LINQ to SQL calls prior to persisting a Product entity. Within this OnValidate() method I could add the following business rule to enforce that products can't have a Reorder Level if the product is discontinued:

```
Product.vb*
(General)
Partial Public Class Product
    Private Sub OnValidate()
        If (Me.Discontinued = True) And (Me.UnitsOnOrder > 0) Then
            Throw New ArgumentException("Reorder level can't be greater than 0 if Product is Discontinued",
        End If
    End Sub
End Class
```

Once I add the above class into my LINQ to SQL project, the above business rule will be enforced anytime anyone uses my data model to try and modify the database. This is true for both updating existing Products, as well as adding new Products into the database.

Because the <asp:LinqDataSource> that we defined in our pages above works against our LINQ to SQL data model classes, all of its update/insert/delete logic will now have to pass the above validation check prior to the change being persisted. We do not need to-do anything to our UI tier in order for this validation to occur - it will automatically be applied anywhere and everywhere our LINQ to SQL data model is used.

#### Adding Nice Error Handling in our UI Tier

By default if a user now uses our GridView UI to enter a non-valid UnitsOnOrder/Discontinued combination, our LINQ to SQL data model classes will raise an exception. The <asp:LinqDataSource> will in turn catch this error and provides an event that users can use to handle it. If no one handles the event then the GridView (or other) control bound to the <asp:LinqDataSource> will catch the error and provide an event for users to handle it. If no one handles the error there then it will be passed up to the Page to handle, and if not there to the global Application\_Error() event handler in the Global.asax file. Developers can choose any place along this path to insert appropriate error handling logic to provide the right end-user experience.

For the application we defined above, probably the best place to handle any update errors is by handling the RowUpdated event on our GridView. This event will get fired every time an update is attempted on our datasource, and we can access the exception error details if the update event fails. We can add the below code to check if an error occurs, and if so



display an appropriate error message to the end-user:

StyleSheet.cssProducts.aspx.vbProducts.aspxProduct.vb

(General) (Declarations)

```
Public Partial Class Products
    Inherits System.Web.UI.Page

    Private Sub GridView1_RowUpdated(ByVal sender As Object, ByVal e As GridViewUpdatedEventArgs)

        If Not e.Exception Is Nothing Then

            If TypeOf (e.Exception) Is ArgumentException Then
                ErrorMessage.Text = CType(e.Exception, ArgumentException).Message
            Else
                ErrorMessage.Text = "An error occured while trying to update this product"
            End If

            e.ExceptionHandled = True
            e.KeepInEditMode = True

        End If

    End Sub

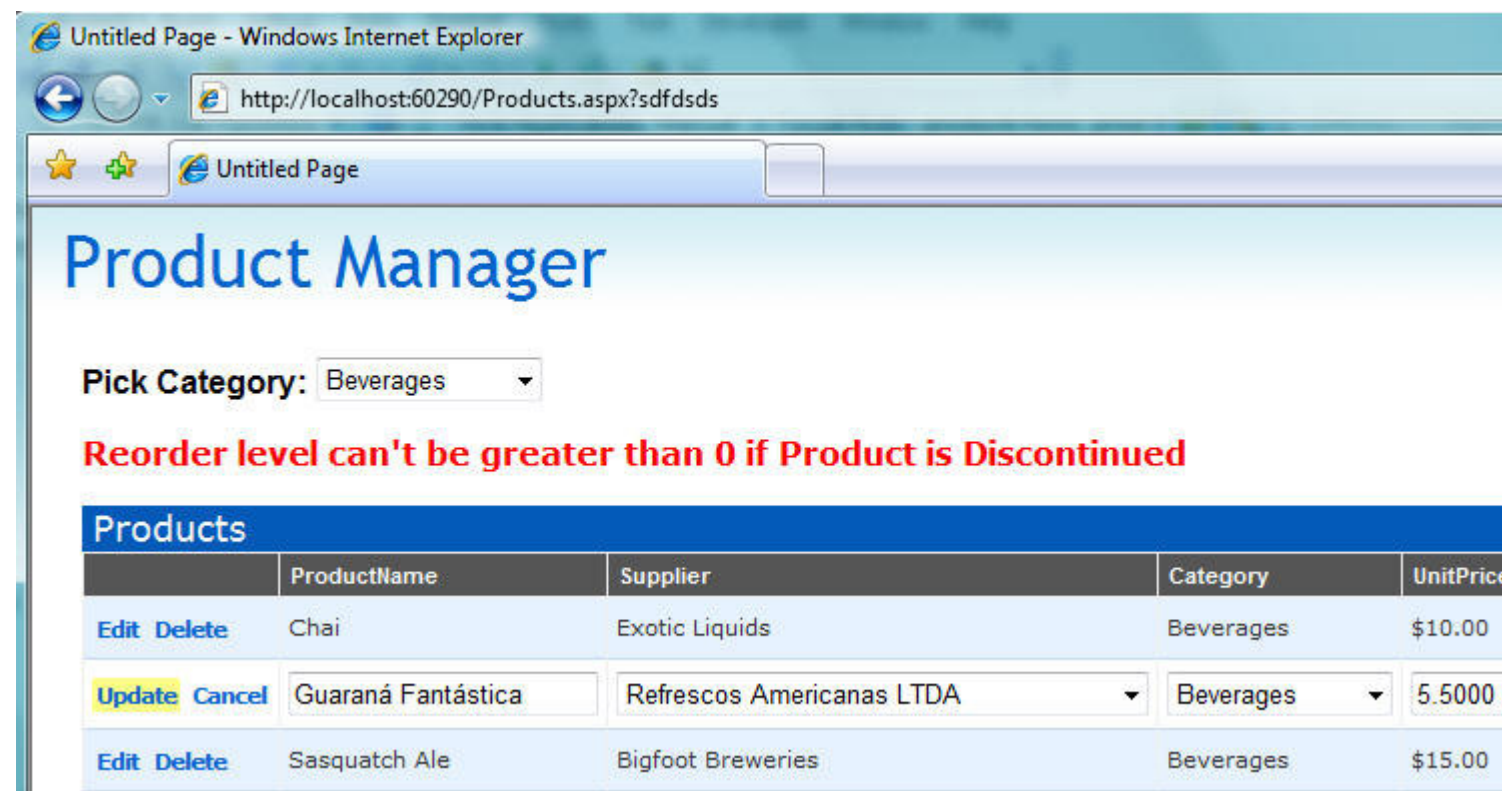
End Class
```

Notice above how we have not added any validation specific logic into our UI. Instead, I am retrieving the validation error message string we raised in our business logic and am using it to display an appropriate message to the end-user (I am then displaying a more generic error message in the event of other failures).

Notice how I'm also indicating above that I want the GridView to stay in Edit mode when an error occurs - that way the user can avoid losing their changes, and can modify the values they entered and click "update" again to try and save them. We can then add a <asp:literal> control with the "ErrorMessage" ID anywhere we want on our page to control where we want the error message to be displayed:

```
<span id="errorMsg">
    <asp:literal ID="ErrorMessage" runat="server" />
</span>
```

And now when we try and update the Product with an invalid value combination we'll see an error message indicating how to fix it:

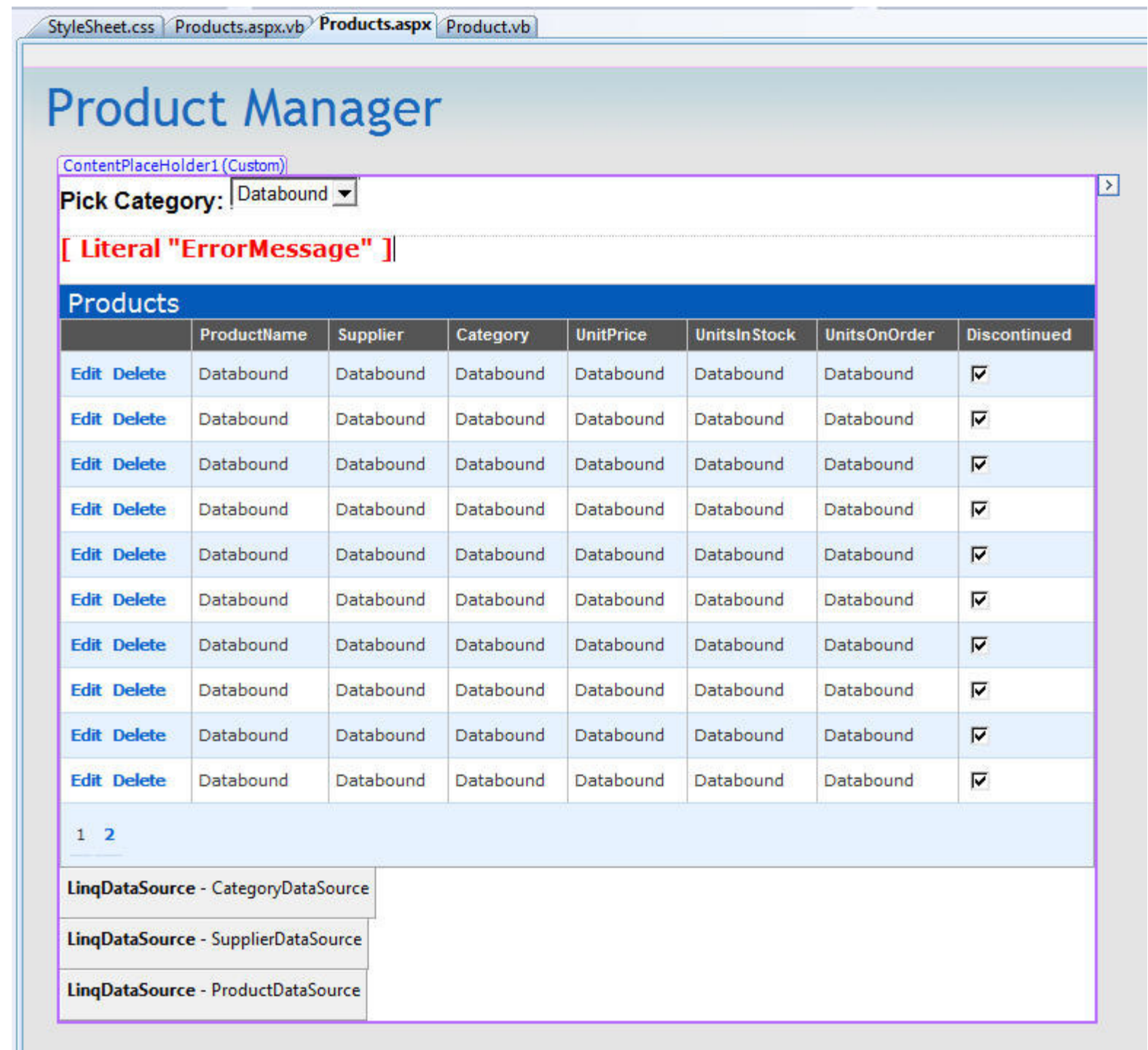


The nice thing about using this approach is that I can now add or change my data model's business rules and not have to modify any of my UI tier's code to have them pick up and honor the changes. The validation rules, and corresponding error messages, can be written and centralized in one place in my data model and automatically applied everywhere.

Summary

The <asp:LinqDataSource> control provides an easy way to bind any ASP.NET UI control to a LINQ to SQL data model. It enables UI controls to both retrieve data from a LINQ to SQL data model, as well as cleanly apply updates/inserts/deletes to it.

In our application above we used the LINQ to SQL ORM designer to create a clean, object oriented data model. We then added three ASP.NET UI controls to our page (a GridView, a DropDownList, and a ErrorMessage Literal), and added three <asp:LinqDataSource> controls to bind Product, Category, and Supplier data from it:



We then wrote 5 lines of business validation logic in our data model, and 11 lines of UI error handling logic.


The end result is a simple web application with custom UI that allows users to dynamically filter product data by category, efficiently sort and page over the product results, inline-edit the product data to save updates (providing our business rules pass), and delete products from the system (also providing our business rules allow it).

In future posts in this series I'll cover more LINQ to SQL scenarios including optimistic concurrency, lazy and eager loading, table mapping inheritance, and custom SQL/SPROC usage.

Next week I also plan to start a new series of blog posts that cover the new <asp:ListView> control - which is a new control that will ship with the ASP.NET release in .NET 3.5. It provides total control over the markup generated for data scenarios (no tables, no spans, no inline styles...), while also delivering built-in support for paging, sorting, editing, and insertion scenarios. For example, we could optionally use it to replace the default Grid layout look of our application above with a completely custom look and feel. Best of all, I could replace it within the above page in my application and not have to change my Data Model, the <asp:linqdatasource> declaration, or my code-behind UI error handling logic at all.

Hope this helps,

Scott

Published Monday, July 16, 2007 11:03 PM by [ScottGu](#)   
Filed under: [ASP.NET](#), [Visual Studio](#), [.NET](#), [LINQ](#), [Data](#), [SQL Server](#)

## Comments

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 3:04 AM by [Majid Shahabfar](#)

Well done Scott,

These are great blog series.

I'm waiting for custom SQL/SPROC usage.

[# LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:29 AM by [DotNetKicks.com](#)

You've been kicked (a good thing) - Trackback from DotNetKicks.com

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:30 AM by Sean Briscoe

Thanks for posting. As usual, I found it only creating more interest for the upcoming .NET 3.5.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:44 AM by Matt

Very nice introduction regarding linqdatasource control



[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:44 AM by [FransBouma](#)

In the part where you display human readable supplier and category names, it's unclear for the reader that these names are read-in using lazy loading, which means 2 queries per row, as that's how ASP.NET works: per row that expression is evaluated, which results in a hit to the property 'Supplier' or 'Category' on the active product, which results in 2 queries per row: one for the supplier, and one for the category. This is really performance intense. Can you specify a span/prefetch path as well?

Furthermore: the linq datasource suggests that any linq enabled O/R mapper can be used with it, as it is a LinqDataSource, not a LinqToSqlDataSource. I find this pretty bad, as I fail to see how it can be used with every Linq enabled O/R mapper out there as it exposes Linq-to-Sql specific features, not generic Linq features in general. This is shown by the 'tablename' attribute in the HTML tag for the linqdatasource control. What if the Linq enabled O/R mapper supports inheritance over multiple tables (as Linq to Sql doesn't support) ? Shouldn't it be 'EntityName' anyway because what if Linq to Sql is going to add entity inheritance over multiple tables in the future, what's the upgrade path then?

One other thing I want to ask is about the group by option: do you display a warning that after selecting group by, the entities aren't updateable anymore? Because grouped data has a totally different meaning than having a list of entity data: grouping that entity data makes the list effectively readonly.

(and I also wonder why on earth people want to mess inside HTML using strings (inside the eval calls) with property references instead of using strongly typed code in the code behind. I mean: if one renames 'Category' on the Product entity into 'MainCategory', this page will fall apart at runtime, not at compile time. )

[# ScottGu's Posts on LINQ to SQL](#)

Tuesday, July 17, 2007 5:03 AM by [Mike Taulty's Blog](#)

I've been reading ScottGu's posts on LINQ to SQL (start here from where it's easy to work backwards)....

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 5:06 AM by [Vlad](#)

Pingback from Code-news

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 5:36 AM by Akaka

Scott,

Which version of Visual Studio you using on this demo? I am using Visual Studio Codename Orcas Version 9.0.20404 Beta1 and cannot see asp:linqdatasource control on toolbox.

Thanks,

Akaka

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 5:49 AM by [Johannes Hansen](#)

Is it possible for you to release the source for the LinqDataSourceControl or better yet the EntityDataSourceControl + the accompanying view(s)? Lately I've been developing a DataSourceControl for the company I work for based on our custom business entity framework. The DataSource control I implemented (using a reflector'ed view of the ObjectDataSource) is working fine but it would still be very interesting to see how you guys are implementing the new DataSource controls in .NET 3.5 especially the EntityDataSource.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 8:05 AM by [Ryan CrawCour](#)

really looking forward to more articles in this series; especially on optimistic concurrency!

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 8:26 AM by Mike

FransBouma makes some excellent points, but I think the group by he refers to is actually a 'where' in the select statement, so edits should work just fine.

Anyway, it's nice to finally see some new web controls. Is the GridView updated as well (perhaps thead/tbody/tfoot and colspec finally)?

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 9:27 AM by Steve

Excellent writeup Scott.

Frans does bring up an interesting question about the lazyloading - will we have 'eagerloading' option with Dlinq to Sql ?

secondary question: If I'm using a SOA where I put my business logic behind webservices that I expose that pass a dlinq object back and forth - I assume since I'm not exposing the datacontext that this won't work?

In my architectures, I would not expose my datacontext to my UI.

Thanks!

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 9:41 AM by [Chad Smith](#)

Hi Scott,

For the non-believers and the curious out there, would it be possible to post a breakdown of the SQL generated by linq when running the example above and explain any optimizations in place and how they relate to the code, or any extra calls to the database we may be seeing that need any explanation. I think this sample is poignant in that it implicitly performs a join where one would expect a dumb system to call many additional queries, can you disprove this?

Cheers

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 9:50 AM by Joe Chung

(This is with regards to Frans's comment.) The reason people use Eval calls instead of code-behind is if they want to use ASP.NET's no-compile pages and still be able to leverage data binding functionality.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 10:19 AM by [Josh Stodola](#)

Oh my gosh, this looks so simple I am going to cry...

Thanks, Scott, for posting an abundance of screenshots, they are \*so\* helpful.

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 10:56 AM by [ScottGu](#)

Hi Frans,

>>>>> In the part where you display human readable supplier and category names, it's unclear for the reader that these names are read-in using lazy loading, which means 2 queries per row, as that's how ASP.NET works: per row that expression is evaluated, which results in a hit to the property 'Supplier' or 'Category' on the active product, which results in 2 queries per row: one for the supplier, and one for the category. This is really performance intense. Can you specify a span/prefetch path as well?

LINQ to SQL allows you to specify whether to lazily or eagerly span/pre-fetch your associations. So for example when you say MyProduct.Supplier.CompanyName in a loop (like I am above with the GridView) you can either end up doing one call per loop iteration, or a join that returns all of the relevant Company name's and assembles them for Products on the client (which is obviously more efficient).

I'm planning on covering how to span/pre-fetch data in my next blog post in the LINQ to SQL series. Basically you can do this one of two ways: 1) either by changing the annotations of your data models (for example: by using the designer), or (more commonly): 2) by specifying the span/pre-fetch pattern to use when you instantiate your DataContext. LINQ to SQL supports using Lambda expressions to define this - which makes it very powerful.

I'll cover how to-do both in my next LINQ to SQL tutorial series - and show both how to express it via code, as well as using the <asp:linqdatasource>.

>>>>>> Furthermore: the linq datasource suggests that any linq enabled O/R mapper can be used with it, as it is a LinqDataSource, not a LinqToSqlDataSource. I find this pretty bad, as I fail to see how it can be used with every Linq enabled O/R mapper out there as it exposes Linq-to-Sql specific features, not generic Linq features in general. This is shown by the 'tablename' attribute in the HTML tag for the linqdatasource control. What if the Linq enabled O/R mapper supports inheritance over multiple tables (as Linq to Sql doesn't support) ? Shouldn't it be 'EntityName' anyway because what if Linq to Sql is going to add entity inheritance over multiple tables in the future, what's the upgrade path then?

Right now our plan has been to add support for additional OR/M implementations to the LINQDataSource in the future (we have a prototype of it running with LINQ to Entities and plan on adding support for it in the future). One thing we will also do is release the source for it. It would obviously be great to get support for it with additional OR/M's like LLBLGen.

>>>>>> One other thing I want to ask is about the group by option: do you display a warning that after selecting group by, the entities aren't updateable anymore? Because grouped data has a totally different meaning than having a list of entity data: grouping that entity data makes the list effectively readonly.

There are operations you can do with the LINQDataSource that make the data read-only. Doing a custom hierarchical grouping of the data is one of them (you can define this using the "Group" button in the dialog above). You can still use all of the declarative binding features with this though (including paging, sorting, and declarative filters) - so even though the data is read-only there are still very interesting things you can do with it. :-)

Note that if you using Grouping or a custom projection to shape the data in a read-only way, the LINQDataSource at design-time will grey out update semantics so you know those have been disabled by your action.

>>>>>> (and I also wonder why on earth people want to mess inside HTML using strings (inside the eval calls) with property references instead of using strongly typed code in the code behind. I mean: if one renames 'Category' on the Product entity into 'MainCategory', this page will fall apart at runtime, not at compile time. )

The Eval() statement is a feature people either love or hate. :-) You could alternatively call out to an event in your code-behind to set this value instead.

Hope this helps,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 10:58 AM by [ScottGu](#)

Hi Akaka,

>>>>>> Which version of Visual Studio you using on this demo? I am using Visual Studio Codename Orcas Version 9.0.20404 Beta1 and cannot see asp:linqdatasource control on toolbox.

Sorry - I should have been more clear. I am using VS 2008 Beta2 for this tutorial (the asp:LinqDataSource was not yet implemented in Beta1). VS 2008 and .NET 3.5 Beta2 should be available for download in a little over a week.

Thanks,

Scott

**# LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control) - ScottGu's Blog**

Tuesday, July 17, 2007 11:34 AM by [LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\) - ScottGu's Blog](#)

Pingback from LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control) - ScottGu's Blog

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 11:39 AM by Leo

How compatible with Vista is Beta 2? What's the performance like on Vista? Will Beta 2 be released as an ISO?

Thanks Scott!

-Leo

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 11:53 AM by Jeff

Very nice. It requires much less work than the SqlDataSource. I will be using the heck out of it, but only in my own work. I find too often that many people are afraid of new things and look for any excuse to not use it. Too bad for them.

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:01 PM by Ben Hayat

Hi Scott;

As I was reading this new post, the following sentence brought a question to mind:

>>VS 2008 includes build-in designer support to make it easy to connect up our GridView (or any other ASP.NET server control) to LINQ data.<<

Does LinqDataSource work with controls that are currently being used in VS2005 or all the third party databound controls that were working previously with other data sources? Another words, do I have to wait for third party products to be changed to work with LinqDataSource or can they work as if nothing has changed from their point of view? Can you use the grid from VS2005 with this new datasource?

Thanks Scott!

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:04 PM by Eric Newton

does the LinqDatasource control expose some events so that we could specify a linq query in the codebehind?

I look at it and it seems to fall way short of what I was expecting from a LinqDatasource control... It would be like SqlDatasource having Database/TableName property instead of SelectCommand.



But since none of us have the ability to play with it, I'm really looking forward to trying it out and understanding the design concept behind LinqDatasource

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:14 PM by [ScottGu](#)

Hi Leo,

>>>>> How compatible with Vista is Beta 2? What's the performance like on Vista? Will Beta 2 be released as an ISO?

VS 2008 and .NET 3.5 are definitely compatible with Vista. That is the OS I'm using with it right now and I haven't had any issues. I believe both a standalone setup as well as an ISO image is available.

Hope this helps,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:17 PM by [ScottGu](#)

Hi Ben,

>>>>>> Does LinqDataSource work with controls that are currently being used in VS2005 or all the third party databound controls that were working previously with other data sources? Another words, do I have to wait for third party products to be changed to work with LinqDataSource or can they work as if nothing has changed from their point of view? Can you use the grid from VS2005 with this new datasource?

The LinqDataSource uses the standard DataSource model that we introduced in ASP.NET 2.0. What this means is that any control that today supports an ObjectDataSource or SqlDataSource will work against the LinqDataSource - so you do not need to wait for your third party controls to be updated (they should just work).

Hope this helps,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:22 PM by [ScottGu](#)

Hi Eric,

>>>>>> does the LinqDatasource control expose some events so that we could specify a linq query in the codebehind?

There is a "Selecting" event that you can handle in your code-behind to provide a custom LINQ query for the LINQDataSource to use. This should give you the ability to customize the query however you want.

Hope this helps,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:25 PM by Ben Hayat

>>The LinqDataSource uses the standard DataSource model that we introduced in ASP.NET 2.0. What this means is that any control that today supports an ObjectDataSource or SqlDataSource will work against the LinqDataSource - so you do not need to wait for your third party controls to be updated (they should just work).<<

GOD bless you guys!

Just bought a new Core 2 Quad Desktop yesterday for VS2008 ;-)

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:33 PM by Ben Hayat

Scott, I'm sorry if this question is unrelated to your post, but your answer might save me a lot of hassle. As mentioned in my previous post, just bought a new PC, but it comes with Home Vista Premium. Can I install VS2008 on Home Vista or do I need to reformat my hard disk and use my Vista from my MSDN subscription?

Thank you very much for taking the time and answering all my questions!

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:40 PM by [ScottGu](#)

Hi Ben,

>>>>>>> Scott, I'm sorry if this question is unrelated to your post, but your answer might save me a lot of hassle. As mentioned in my previous post, just bought a new PC, but it comes with Home Vista Premium. Can I install VS2008 on Home Vista or do I need to reformat my hard disk and use my Vista from my MSDN subscription?

You can definitley install VS 2008 on Home Vista. The only feature I know of that you might care about that isn't in Home Premium is that IIS7 doesn't support Windows Authentication on it (although you can use ASP.NET's Forms Authentication). If you want to use that feature you might want to upgrade - but otherwise it should work for you.

Hope this helps,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:46 PM by Leo

Thanks for the response above. One final question: How feature complete will Beta 2 be? Or put another way, is there anything not in Beta 2 that will be in the release? ADO.NET Entity Framework, for example, comes to mind.

Thanks again Scott!

-Leo

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 17, 2007 12:51 PM by [ScottGu](#)

Hi Leo,

>>>>>>> Thanks for the response above. One final question: How feature complete will Beta 2 be? Or put another way, is there anything not in Beta 2 that will be in the release? ADO.NET Entity Framework, for example, comes to mind.

Beta2 will be mostly feature complete. There are a few additional small tweaks that will be made based on feedback, but overall the feature set will be 99% there.

The ADO.NET Entity Framework won't be in the VS 2008 or .NET 3.5 release when it ships. It will instead ship later next year.

Hope this helps,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 1:57 PM by Jim

Scott,

Your examples in this post seem to allow for very rapid, but simple development. I have a hard time actually using the designers for production code and tend always use code to do databinding. Do you find yourself using the designers more often than typing by hand?

Great post none the less and I'm very excited for VS 2008.

Jim

[# Lives writer &laquo; chakrit](#)

Tuesday, July 17, 2007 2:10 PM by [Lives writer &laquo; chakrit](#)

Pingback from Lives writer &laquo; chakrit

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 2:18 PM by elixir

can't wait till this goes rtm!

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 2:30 PM by JasonBSteele

Hi Scott,

Looks great - and I'm glad you're saying it'll be with us in a couple of weeks as i had heard horror stories that it wouldn't be until "late summer" (summer hasn't even started in the UK)!

However, I think I share Eric's and Franks concerns about the use of TableName to specify the query being a bit restrictive. And trapping it in an event to change it into suomething else just doesn't seem very elegant.

Are there plans to add other ptopertie(s) that would allow us to specify a query instead? Or perhaps we'll have to wait until the EF version before we get this sort of thing. In which case do you know how that might specy the query? (Perhaps an eSQL string?)

Thanks again for sharing,

Jason

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 3:04 PM by [Roger Hendriks](#)

Hi Scott,

I notice Microsoft is supporting more and more backend logic in the presentation layer. From a 3-tier pont of view this looks dirty (to say the least:). Is this to please the old asp and php hackers to finally start using asp.net?

Grtz from Roger

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 3:14 PM by [billie](#)

All looks great, can't wait to use it. What sort of plan do you guys have in place to convince large companies that LINQ to SQL is going to be a good, safe choice? Lots of large companies have the database divide where developers do not write sql/store procs dbms do. What is in place to push linq adoption in groups such as these?

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 3:56 PM by [kevindente](#)

Great info. Thanks, Scott.

It's a drag, though, that the developer has to do so much manual work to resolve foreign key references. That seems like a primary use case, and it should just work out of the box (as it does in Subsonic, for example).

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:08 PM by Chris Moseley

Hi Scott,

Great post, I'm really impressed with most of LINQ To Sql but I have a few questions.

Frans said:

>>>>>>>>> In the part where you display human readable supplier and category names, it's unclear for the reader that these names are read-in using lazy loading, which means 2 queries per row, as that's how ASP.NET works: per row that expression is evaluated, which results in a hit to the property 'Supplier' or 'Category' on the active product, which results in 2 queries per row: one for the supplier, and one for the category. This is really performance intense. Can you specify a span/prefetch path as well?

>>>>> LINQ to SQL allows you to specify whether to lazily or eagerly span/pre-fetch your associations. So for example when you say MyProduct.Supplier.CompanyName in a loop (like I am above with the GridView) you can either end up doing one call per loop iteration, or a join that returns all of the relevant Company name's and assembles them for Products on the client (which is obviously more efficient).

span/prefetch path as well?"

I've looked into how to do this before (using a DataShape.LoadWith) and noticed that it's possible to join two tables together resulting in a single query (for say, customers and orders). This is fine, however, is it possible to avoid lazy loading on more two tables? for example, can I retrieve Products, Suppliers and Categories all in one go, without any extra queries for each product?

If I wanted to use LINQ To Sql to bind to a GridView without using a LinqDataSource, how would I do the column sorting, which would have to use a dynamic field name in the query?

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:15 PM by [ScottGu](#)

Hi Jim,

>>>>>>> Your examples in this post seem to allow for very rapid, but simple development. I have a hard time actually using the designers for production code and tend always use code to do databinding. Do you find yourself using the designers more often than typing by hand?

I usually use a combination of the designers + writing the code by hand. I find some of the databinding picker dialogs useful (since they use reflection to look up things for me) and also a good way to learn for things I'm not super familiar with. Once I know what I am doing I then usually just drop into source mode and type it. Thankfully the <asp:linqdatasource> is fairly easy to type - since it only



really requires you to specify the class name of your content and the original entity table to use to get started.

Hope this helps,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:18 PM by [ScottGu](#)

Hi Jason,

>>>>> However, I think I share Eric's and Franks concerns about the use of TableName to specify the query being a bit restrictive. And trapping it in an event to change it into suomething else just doesn't seem very elegant. Are there plans to add other ptopertie(s) that would allow us to specify a query instead? Or perhaps we'll have to wait until the EF version before we get this sort of thing. In which case do you know how that might specy the query? (Perhaps an eSQL string?)

LinqDataSource allows you to specify a declarative select, where and orderby statement that allows you to-do custom shapeing, filtering and ordering for common scenarios. We thought of providing more declarative support for more advanced queries - but ultimately decided that since LINQ is a terse way to represent queries (and has the benefit of intellisense, debugging and compile-time checking) it didn't make sense for us to invent yet a new language to express these more advanced scenarios. Instead you can just add a method to your code-behind class and use LINQ within it to express any query you want.

Hope this helps,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:19 PM by [ScottGu](#)

Hi Roger,

>>>>>> I notice Microsoft is supporting more and more backend logic in the presentation layer. From a 3-tier pont of view this looks dirty (to say the least:). Is this to please the old asp and php hackers to finally start using asp.net?

In my sample above I don't have any backend logic in the presentation layer. All of the actual logic on how to bind against the database (column/schema mappings) and business logic rules are kept entirely out of the presentation tier (and instead in the data model). I think capabilities like LINQ to SQL hopefully encourage even more people to cleanly build their applications this way in the future.

Thanks,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:22 PM by [ScottGu](#)

Hi Billie,

>>>>> All looks great, can't wait to use it. What sort of plan do you guys have in place to convince large companies that LINQ to SQL is going to be a good, safe choice? Lots of large companies have the database divide where developers do not write sql/store procs dbms do. What is in place to push linq adoption in groups such as these?

Different companies and teams usually have different sets of requirements. So rather than force people to only do things one way, we typically want to provide good options and guidance and allows teams to customize where necessary based on their specific needs.

I do think LINQ to SQL will nail 90%+ of application scenarios, though, and will provide an awesome experience for building applications that scale both from simple requirements to very demanding ones. I think the productivity benefits it brings, as well as the integrated language and tool support, will hopefully convince teams of its value.

Hope this helps,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:25 PM by [ScottGu](#)

Hi Kevin,

>>>>>> It's a drag, though, that the developer has to do so much manual work to resolve foreign key references. That seems like a primary use case, and it should just work out of the box (as it does in Subsonic, for example).

We actually have some more dynamic UI control support (which is based on the <asp:linqdatasource>) that will be coming out in a few months that allows you to get dynamic column support (including visualizing things like drop-downlists for foreign-keys automatically). I used the vanilla GridView control above for the sample mainly since it is already in the box, and people know it today. But we'll definitely add to it in the future to make this scenario even more integrated.

Hope this helps,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 4:48 PM by Mike

You're doing a great job answering all questions, thanks!

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 5:01 PM by [ScottGu](#)

Hi Chris,

>>>>> I've looked into how to do this before (using a DataShape.LoadWith) and noticed that it's possible to join two tables together resulting in a single query (for say, customers and orders). This is fine, however, is it possible to avoid lazy loading on more two tables? for example, can I retrieve Products, Suppliers and Categories all in one go, without any extra queries for each product?

I know you can use the DataShape feature in LINQ to SQL to support more than two tables. Whether you can retrieve all of the data in a single query, though, probably depends on the structure of your data. I need to play around with it a little to learn more. One thing to do in VS 2008 Beta2 is to use the debug visualizers I talked about in Part 3 of this blog post series to look at the actual SQL generated when you execute a query. That should provide a good way to see what SQL it is executing under the covers.

>>>>>>> If I wanted to use LINQ To Sql to bind to a GridView without using a LinqDataSource, how would I do the column sorting, which would have to use a dynamic field name in the query?

The easiest way to have a UI control do column sorting is to use the LinqDataSource. You could then handle the "selecting" event off of the LindDataSource if you want to use your own logic to retrieve the data.

LINQ and LINQ to SQL queries support deferred execution - which means they don't run until you try and access the data from them. Once thing thing this enables is the ability to compose queries out of other queries - allowing you to build up things like conditional filtering and ordering. You could do this in response to a column header be clicked, and add an orderby clause to the LINQ expression based on what column is clicked.

Hope this helps,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, July 17, 2007 5:26 PM by Aykut Ucar

Will Beta2 be with go live license?

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Tuesday, July 17, 2007 5:32 PM by ScottGu

Hi Aykut,

>>>>>> Will Beta2 be with go live license?

Yes - Beta2 will have a go-live license.

Hope this helps,

Scott

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Tuesday, July 17, 2007 5:40 PM by Chris Moseley

Hi Scott,

Thanks for replying; I agree with Mike, you're doing a great job answering all these questions.

>>>>>I know you can use the DataShape feature in LINQ to SQL to support more than two tables.

I've looked quite hard and I can't figure out how... for example:

```
NorthwindDataContext ctx = new NorthwindDataContext();
```

```
    DataShape shape = new DataShape();
```

```
    shape.LoadWith<Customer>(c => c.Orders);
```

```
    ctx.Shape = shape;
```

It looks as if I can only apply a single Shape object to each context, and that the LoadWith<T> method only supports a single expression so how would I 'pre-load' the data for more than two tables?

>>>>>The easiest way to have a UI control do column sorting is to use the LinqDataSource

That's fair enough I'm sure I can bring myself to use declarative code if I must ;-)

>>>>>You could do this in response to a column header be clicked, and add an orderby clause to the LINQ expression based on what column is clicked.

But, when you click a column, you'll get a string representation of a field name - how would you use that in an orderby clause without the aid of a potentially enormous switch/select statement? Correct me if i'm wrong but the following is not currently possible:

```
from c in ctx.Customers
```

```
orderby "name"
```

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Tuesday, July 17, 2007 6:31 PM by ScottGu

Hi Chris,

>>>>>>> It looks as if I can only apply a single Shape object to each context, and that the LoadWith<T> method only supports a single expression so how would I 'pre-load' the data for more than two tables?

I think you can call the LoadWith() statement multiple times. So to setup relationships across multiple tables just call it multiple times with each one.

>>>>>>> But, when you click a column, you'll get a string representation of a field name - how would you use that in an orderby clause without the aid of a potentially enormous switch/select statement?

You can use a switch statement today to construct the LINQ queries dynamically in a strongly typed way. There is also a utility library that we'll ship the source to that allows you to easily construct LINQ expressions dynamically via strings. That is actually what the LinqDataSource does internally - and which you can optionally use instead to make the orderby values completely dynamic.

Hope this helps,

Scott

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Tuesday, July 17, 2007 11:41 PM by kevindente

>>>>>>>>We actually have some more dynamic UI control support (which is based on the <asp:linqdatasource>) that will be coming out in a few months that allows you to get dynamic column support

Scott,

Will that be part of Orcas proper? ASP.NET Futures? Some open source project? Or a future release of the .NET framework?

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Wednesday, July 18, 2007 12:24 AM by Vikram

wow, I really love the concept of Drop down list in Place of Related Data. Way to go. Cant wait for the beta 2 release. Please get it as fast as possible.

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Wednesday, July 18, 2007 12:35 AM by Ngoc Luu

Hi Scott,

In VS2008 Beta1, when I place some WebPartZone in an UpdatePane, each WebPartZone has some WebParts in it, when I switch webparts to Design mode, I can't drag WebParts to other zones.

Will WebPart and UpdatePanel in VS2008 Beta2 be compatible together?

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Wednesday, July 18, 2007 2:22 AM by ScottGu

Hi Kevin,

>>>>> Will that (the dynamic data controls) be part of Orcas proper? ASP.NET Futures? Some open source project? Or a future release of the .NET framework?

We are looking to make this initially available as an ASP.NET extensions release (similar to the ASP.NET AJAX release). It would then be built-in to the next release of the .NET Framework.

Hope this helps,



Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 2:23 AM by [ScottGu](#)

Hi Ngoc,

>>>>>>>In VS2008 Beta1, when I place some WebPartZone in an UpdatePane, each WebPartZone has some WebParts in it, when I switch webparts to Design mode, I can't drag WebParts to other zones.

Will WebPart and UpdatePanel in VS2008 Beta2 be compatible together?

In Beta2 you'll be able to put your webpartzone's within an UpdatePanel, and then have each WebPart optionally use its own UpdatePanel as well. This should give nice integration between ASP.NET AJAX and WebPart scenarios.

Hope this helps,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 3:00 AM by Vishal Doshi

Well done Scott ! I'm waiting for more to come in 3.5 from your side .... Thanks a lot.

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 3:41 AM by [Keith Patton](#)

for paging, is the default behaviour to pull back all results and then just show a page, or is the LinqDataSource clever enough to use 2 queries internally and RowNumber() with SQL 2005 to implement performant querying. If not, you would have objects instantiated for every single row in the query results resulting in presumably poorer performance than filling and returning dataset. We use NHibernate a lot and although it does not support ROWNUmber() with SQL 2005 as yet, you can tell it a start and end result and it will at least resolve the page of results at teh database end of things and only instantiate objects for the number of results you are using.

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 3:58 AM by [FransBouma](#)

"Right now our plan has been to add support for additional OR/M implementations to the LinqDataSource in the future (we have a prototype of it running with LINQ to Entities and plan on adding support for it in the future). One thing we will also do is release the source for it. It would obviously be great to get support for it with additional OR/M's like LLBLGen."

We already have our own datasourcecontrol, so I think we'll keep that for some time, but it's good to hear MS sees the LinqDataSource as a general datasource for all Linq enabled O/R mappers. :)

The only thing I'm a bit bothered with is that it's not that generic when Orcas hits the streets. I fail to see why, as it's a new control, so you could have designed it as generic as possible from the ground up, but I have the feeling it wasn't meant to be in the first place. If you ship this as a non-generic datasource control, it will be a true sign MS doesn't understand that .NET is a framework others build upon, not an end-user product. Sorry to say it Scott, but 'in the future' means not before 2009, so effectively a missed chance.

Will that sourcecode be shipped with orcas or later?

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 4:24 AM by Nagarajan

Is there anything like LinqDataSource for WPF to work with?

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 4:25 AM by Roger Hendriks

Hi Scott,

Thanks for you answer, I myself add everything not regarding layout in the codebehind cs (even dynamic validators) but thats a personal choice. I guess "Tables" and the property "TableName" is also a little disturbing in this respect. I presume it's the object on which the actual table is mapped with attribute generated by the designer?

One question about performance. Is the example going to lead to 20 subselects for categories and suppliers? If so a join with paging example would be nice :)

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 4:54 AM by [ScottGu](#)

Hi Frans,

>>>>>>> The only thing I'm a bit bothered with is that it's not that generic when Orcas hits the streets. I fail to see why, as it's a new control, so you could have designed it as generic as possible from the ground up, but I have the feeling it wasn't meant to be in the first place. If you ship this as a non-generic datasource control, it will be a true sign MS doesn't understand that .NET is a framework others build upon, not an end-user product. Sorry to say it Scott, but 'in the future' means not before 2009, so effectively a missed chance. Will that sourcecode be shipped with orcas or later?

The DataSourceControl base class is designed to enable truly generic data source controls (like the one you've already built). All ASP.NET UI controls work against that, as opposed to specific implementations, so that they can generically be bound to any data provider.

The LinqDataSource is not designed to be fully generic for all ORMs - because different ORM implementations have slightly different semantics, having it work against all out of the box is a tall order. Our hope overtime is that we can add additional support so that more and more LINQ ORMs just work perfectly fine with it. We'll also be shipping the source code (hopefully around the time Orcas ships) so that other ORM implementations can adapt their own customized versions as needed.

Hope this helps,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 4:54 AM by [ScottGu](#)

Hi Nagarajan,

>>>>> Is there anything like LinqDataSource for WPF to work with?

Yes - WPF will also support databinding using LINQ. I'll put this on the list to blog in the future.

Thanks,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, July 18, 2007 4:57 AM by [ScottGu](#)

Hi Roger,

>>>>>> Thanks for you answer, I myself add everything not regarding layout in the codebehind cs (even dynamic validators) but thats a personal choice. I guess "Tables" and the property "TableName" is also a little disturbing in this respect. I presume it's the object on which the actual table is mapped with attribute generated by the designer?

The TableName in this case is based on the objects in the DataContext class - and does not require that the underlying database have a table name with that value. You could map your DataContext to instead refer to a differently named table if you want.

>>>>>> One question about performance. Is the example going to lead to 20 subselects for categories and suppliers? If so a join with paging example would be nice :)

The retrieval behavior depends on whether you shape the LINQ to SQL data context. I'll be doing a blog post in the near future that talks about how to-do this - in which case it will use a join with paging just like you are after. :-)

Thanks,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 18, 2007 5:52 AM by Michael

Is there an easier way to supply parameter values from the code behind page when binding against a DataSource? I did find an example of creating a custom parameter object to do this at the site linked below but it would be nice if it were built into the dotnet framework by default.

[blog.davidyack.com/.../3385.aspx](http://blog.davidyack.com/.../3385.aspx)

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 18, 2007 9:36 AM by [Brendan Enrick](#)

Excellent blog post as always. This was an excellent series. It is great content like this that keeps all of your happy readers here all the time.

Are you planning on writing another series of posts anytime in the near future?

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 18, 2007 10:25 AM by [Mike Griffin](#)

Question:

Why was the ability to have LINQ return DataTable's (or be converted as it were) removed from this latest Orca's release. Was it to prevent 3rd party ORM Vendors from being able to build upon Linq? As it stands it appears that the only thing LINQ can be used for is the ADO.NET Entity Framework?

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 18, 2007 10:28 AM by [Damien Guard](#)

Nice article bar the lack of insert and delete confirmation (not LINQ's fault I know - a limitations of the GridView control that hopefully the new ListView will address.

)amien

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 18, 2007 11:34 AM by CraX

I just wanted to thank you Scott, you're the sh@&!

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 18, 2007 12:11 PM by Orlin

Are cascading dropdowns in databound controls (formview or gridview) supported in this new version? That scenario is difficult to implement in ASP.NET 2.0.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 18, 2007 12:16 PM by Scott Gurvey

Another great article, thanks.

Will there be an equivalant LinqDataSourceControl for Windows Forms GUI? I have been trying to get my data into a DataGridView without success. Perhaps that isn't yet in Beta-1!

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 18, 2007 8:42 PM by Eric Newton

I can't wait to see this... prototyping web apps will be super easy, and reasonably able to "grow up" as the system matures.

As for the TableName, I assume its so the designer can have something to work with without having to have everybody write code... but of course the code option will give the best benefits from LINQ.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Thursday, July 19, 2007 12:31 AM by Nagarajan

Hi Scott,

Thank you very much for your reply about LinqDataSource for WPF. I am very much interested in developing WPF Desktop application for my company. The application users are over 3500. So, please add over all architecture and handling things like database connection and so on. I hope that the coming post will have enough details to start our development.

Regards,

Nagarajan

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Thursday, July 19, 2007 5:49 AM by [ScottGu](#)

Hi Brendan,

>>>>>> Are you planning on writing another series of posts anytime in the near future?

I'll be doing several more posts in this LINQ to SQL series. I'll also then be starting a ListView series that talks about some of the new capabilities of the <asp:listview> control.

Hope this helps,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)



Thursday, July 19, 2007 5:53 AM by [ScottGu](#)

Hi Mike,

>>>>>> Why was the ability to have LINQ return DataTable's (or be converted as it were) removed from this latest Orca's release. Was it to prevent 3rd party ORM Vendors from being able to build upon Linq? As it stands it appears that the only thing LINQ can be used for is the ADO.NET Entity Framework?

I sent email to the data team, and here is the response they came back with:

"I assume that Mike is talking about a feature that we had prior to Beta 1, which was a method that would shred an IEnumerable<T> into a DataTable. We decided to remove the feature because there wasn't a clear "right" way to shred the data. There were many problems without correct answers, such as the following.

- Should the schema be extended for sub-types?
- Should the object graph be shredded as tables/relations?
- How should we handle properties/types that don't correspond to a supported DataSet type?
- Should we use serialization logic?

Given all the problems and issues, we decided that it was better to make the right decision, so we pulled the feature until we could figure out the best way to do this. We had a DCR to address these issues, and ended up with a design that would address core scenarios, but we were unable to fit it in Orcas. Instead, we are going to document how to do object shredding into a DataTable, so that these issues can be resolved by the user for their particular scenario."

Note that features like IQueryable<T> make it possible for any ORM system to integrate very cleanly with LINQ, and to enable you to take full advantage of the new language and tool features against any backend database implementation.

Hope this helps,

Scott

## # re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, July 19, 2007 5:58 AM by [ScottGu](#)

Hi Damien,

>>>>>> Nice article bar the lack of insert and delete confirmation (not LINQ's fault I know - a limitations of the GridView control that hopefully the new ListView will address.

Here is an old blog post I wrote that talks a little about how you can add a confirmation step before deleting something in the GridView: [weblogs.asp.net/.../434362.aspx](http://weblogs.asp.net/.../434362.aspx)

Hope this helps,

Scott

## # re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, July 19, 2007 5:59 AM by [ScottGu](#)

Hi Scott,

>>>>> Will there be an equivalent LinqDataSourceControl for Windows Forms GUI? I have been trying to get my data into a DataGridView without success. Perhaps that isn't yet in Beta-1!

You will definitely be able to use LINQ to SQL with WinForms and the DataGridView control there. I'll try and blog about this in the future.

Thanks,

Scott

## # re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, July 19, 2007 10:21 AM by Ben Hayat

Hi Scott;

I think your blogs and answers are very valuable to us to make decisions. My question is, why other product/program managers, VP, and other architects don't take the same approach of opening and sharing info about WPF, WCF, WF and SilverLight? Recently, there is so much silence about WF and WPF that sometime we wonder if there is any development activity there any more. Your blogs have given us lots of confidence about LINQ and the new SilverLight and ASP.Net, but I wish other PMs would do the same...

Thanks for any comments!

## # re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, July 19, 2007 11:45 AM by Winston Fassett

Keith --

We bumped into this issue with NHibernate and one of my colleagues wrote an alternative SQL Server 2005 dialect for NHibernate that we have been using on multiple projects for several months. It correctly implements paging using ROW\_NUMBER, only retrieving the rows for the given page. It hasn't made it into a stable version yet, but appears to be scheduled for HHibernate 1.2.1.

Here's a link to the JIRA ticket, which contains an attachment for the patched dialect:

[jira.nhibernate.org/.../NH-903.jsessionid=DE3BB77DBA6353F3CF0C0A58187B70D3](http://jira.nhibernate.org/.../NH-903.jsessionid=DE3BB77DBA6353F3CF0C0A58187B70D3)

I'd also like to mention that we wrote an NHibernateDatasource control a few months ago and it is very similar to this one, albeit a bit more amateurish, but that's what you get for free. It supports paging, sorting, updating, generating DataTables, and dynamic WHERE clauses bound to parameters. It's available at CodeProject:

[www.codeproject.com/.../NHibernateDataSource.asp](http://www.codeproject.com/.../NHibernateDataSource.asp)

Scott,

I am still curious about how LinqDataSource is handling the paging, and whether it uses ROW\_NUMBER. If it does, I'm particularly curious about how it gets the total page count. When we tried to do this in our data source control, we had to generate a second query that would perform a SELECT COUNT in order to get the total number of results available, creating a tradeoff between time saved by paging and the time lost in executing the COUNT query. Is there an easier way?

## # re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, July 19, 2007 11:57 AM by Winston Fassett

Scott, also...

I'm wondering if using the LinqDataSource control illuminated any weaknesses in ASP.NET data-binding. I've had a love/hate relationship with ASP.NET databinding, and I believe it's mainly because I use an ORM approach rather than a DataTable-based approach.

The most challenging thing is that 2-way ASP.NET databinding itself, i.e. <# Bind() #> expressions, is VERY restrictive. Half of it must be implemented by DataSourceControls, and half of it seems to be locked up in an exotic compile-time process with nowhere for the developer to hook in. Also, it seems to be designed for tidy two-dimensional DataTable-backed objects, rather than an object graph of variable depth. Even though I can make my data source control support nested binding, the Bind expression imposes a compile-time Regex that restricts the "." character, so I'm out of luck. The shallowness of the current 2-way databinding is frustrating.

Given the new emphasis on LINQ, and this new LinqDataSource, do you anticipate any changes to ASP.NET databinding that would facilitate binding to an object graph? I'd love to see some more flexibility. It would be great if developers could inject their own binding implementation via the web.config, perhaps as a Provider.

Are there any plans at all to revise the current approach to ASP.NET binding?

While I'm on a roll here, I'd just like to make some general comments on DataSource controls:

\* The basic concept of databound controls and DataSourceControls is great. They can save you a lot of coding and testing, especially if they are sitting on another abstraction such as an ORM.

\* Implementing DataSourceControls is a pain in the a\*\*, and I think this is why there are so few out there. I've never seen a good tutorial on how to write one, and we were basically fumbling around in the dark when we wrote ours.

\* Implementing data bound controls is also a real pain, especially if you don't extend the base classes. For example, writing a UserControl that could be backed by a DataSourceControl was far from simple.

\* Let's just say that in order to write DataSourceControls and data-bound controls, I found myself using Reflector a LOT.

Hope that didn't sound too negative. Like I said, it's love/hate.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Thursday, July 19, 2007 6:45 PM by [Juan María](#)

Hi, you can read this post in spanish here:

[thinkingindotnet.wordpress.com/.../linq-to-sql-5%c2%aa-parte-enlazar-controles-de-interfaz-de-usuario-con-el-asplinqdatasource](http://thinkingindotnet.wordpress.com/.../linq-to-sql-5%c2%aa-parte-enlazar-controles-de-interfaz-de-usuario-con-el-asplinqdatasource)

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Friday, July 20, 2007 2:18 AM by [Manuel Abadia](#)

Scott,  
Will bind expressions like Bind("Address.City") be supported for the next release?  
Not being able to work with subobjects is a BIG limitation of the data binding infrastructure.

[# VS Orcas, Visual Studio Code Name Orcas &raquo; Blog Archive &raquo; some of the new features of VS Orcas](#)

Saturday, July 21, 2007 3:58 AM by [VS Orcas, Visual Studio Code Name Orcas > Blog Archive > some of the new features of VS Orcas](#)

Pingback from VS Orcas, Visual Studio Code Name Orcas &raquo; Blog Archive &raquo; some of the new features of VS Orcas

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Saturday, July 21, 2007 5:55 PM by [Roger Jennings](#)

Pingback from [oakleafblog.blogspot.com/.../linq-update-for-week-of-july-16-2007.html](http://oakleafblog.blogspot.com/.../linq-update-for-week-of-july-16-2007.html)

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Sunday, July 22, 2007 1:00 PM by Eric Newton

Winston Fasset: I've actually been complaining about poor object-graph databinding for a while now... for instance, you are probably aware that as of VS2008.B1 you \*STILL\* cannot specify nested properties in a GridView BoundField.  
For instance:

```
<asp:BoundField DataField="Property1.ChildProperty1.ChildProperty1" />
won't work, even if you had the correct types.
```

Scott, what's the deal with this man? Now that we're dealing with object-graphs coming from LINQ, why to this day has the BoundField issue not been addressed?

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Monday, July 23, 2007 7:53 AM by [Zubair.NET!](#)

Hi Scott,  
Is LinqDatasource part of June 2007 CTP? because I couldn't find it due to reasons given below.

Couple of issues off topic, firstly I downloaded the VS Orcas June CTP bits but they appear to be corrupted and couldn't be extracted, then I installed the VWDX June 2007 CTP, it crashes alot and doesn't contain the Entity Framework Designer, it's just too hard to work without the designer to create your Entity DataModel. Does your team intend to ship it with Beta2?

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Monday, July 23, 2007 5:54 PM by [Test](#)

Hi what does this mean in 3 tier (web server talking to app server and appserver talking to database)? ARe we still stuck with Object datasource ?

[# Feature NET &laquo; Cosa importa se&#8230;.](#)

Tuesday, July 24, 2007 6:09 AM by [Feature NET « Cosa importa se....](#)

Pingback from Feature NET &laquo; Cosa importa se&#8230;.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 25, 2007 2:15 PM by [Majid Shahabfar](#)

Hi Scott,  
I wonder if you cover Compiled Queries in your next post please.  
Thank you in advance.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 25, 2007 2:27 PM by Dhaval Patel

What about paging, Scott? I know you covered the "Take" and the "Skip" features in your previous posts, but how do these relate to the LinqDataSource?  
Keep up the great work. Thanks!

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, July 25, 2007 5:29 PM by [movers](#)

I'm confused as well as to if we are stuck with the Object datasource. Maybe this was already explained... I'll re-read.  
I look forward to more posts on LINQ.



**# VS 2008 ベータ2 と同時に ASP.NET Futures July CTP がリリース予定**

Wednesday, July 25, 2007 8:26 PM by [ナオキにASP.NET（仮）](#)

Public Sector Developer Weblog からです。 WEBCAST FOLLOW UP: What's New for Web Developers in Visual Studio

**# Lambdas, and Take While, and Group By, Oh My!**

Thursday, July 26, 2007 5:17 PM by [The Visual Basic Team](#)

Hooray! Visual Basic 2008 Beta2 has been released to the wild today to return to its natural habitat

**# [Visual Studio] Les nouveautés de la Beta 2 de Visual Studio 2008 et du Framework .NET 3.5 Beta 2 !**

Friday, July 27, 2007 3:29 AM by [Thomas Lebrun](#)

La Beta 2 de Visual Studio 2008 est arrivée très récemment et avec elle, on retrouve pas mal de petites

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Friday, July 27, 2007 11:44 AM by [Sorin Dolha](#)

Really nice series. I went through all of it, and I can't be more happy that this is really live. I'm currently downloading the Beta 2 to test myself. I'm curious if the changes recorded by the DataContext will be then sent to the database in the same order that they happen (not like DataSet/TableAdapters where they were sent one table after the other, from top to down in each table). I just HOPE that it records changes and applies them in the appropriate order!

**# LINQ to SQL (パート 5 - ASP:LinqDataSourceコントロールを使用してUIをバインディング)**

Saturday, July 28, 2007 5:48 AM by [Chica's Blog](#)

LINQ to SQL (パート 5 - ASP:LinqDataSourceコントロールを使用してUIをバインディング)

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Saturday, July 28, 2007 1:37 PM by [Stefan Cruysberghs](#)

Great articles Scott ! I've installed Beta 2 and I'm trying to learn LINQ for SQL to use it in a new ASP.NET (Ajax) project. But I have 2 questions :

- 1) How to access a property of a sub-entity in the "Configure Where Expression" wizard ? e.g. I would like to set the CategoryName when querying the Products.
- 2) I have a database which uses integers for boolean values (-1=false, 1=true). So I would like to use Boolean properties in my Entity. How can I convert these values in the Get and Set method of my property ?

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Sunday, July 29, 2007 11:36 AM by [Thomas H. Lanier](#)

What's the trick to get the LinqDataSource to work with tables that contain a timestamp field?

I get the following error when I try to update a row in the GridView:

Exception Details: System.Data.Linq.ChangeConflictException: Row not found or changed

**# Digging into LinQDataSource**

Sunday, July 29, 2007 10:58 PM by [Andres Aguiar's Weblog](#)

The sample code that ScottGu showed for LinQDataSource made it look like it should be called LinQToSqlDataSource

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Monday, July 30, 2007 5:52 PM by [Travis](#)

Scott (or anyone) Can we use the LINQ to SQL designer with an external mapping file and external class files for the entities? Like the old Class Designer?

Also what is best practice for the DataContext class in an ASP.NET environment, should we store it in a HttpContext object or instantiate everytime we use it? Per user? One per application?

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Monday, July 30, 2007 6:04 PM by [Roger Jennings](#)

Pingback from [oakleafblog.blogspot.com/.../linq-changes-from-orcas-beta-1-to-vs.html](http://oakleafblog.blogspot.com/.../linq-changes-from-orcas-beta-1-to-vs.html) (LINQ Changes from Orcas Beta 1 to VS 2008 Beta 2)

--rj

**# What I am doing to become a better developer**

Monday, July 30, 2007 6:08 PM by [Wortzel's blog](#)

Shani added me to his tagged list, so I'll try to answer this question. As an introduction to the answer

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Monday, July 30, 2007 6:57 PM by [Roger Jennings](#)

Here's an anthology of VS 2008 Beta 2's changes to LINQ and its domain-specific implementations, which includes a brief description and link to this post: [oakleafblog.blogspot.com/.../linq-changes-from-orcas-beta-1-to-vs.html](http://oakleafblog.blogspot.com/.../linq-changes-from-orcas-beta-1-to-vs.html)

--rj

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 31, 2007 4:32 AM by [ScottGu](#)

Hi Sorin,

>>>>>>> Really nice series. I went through all of it, and I can't be more happy that this is really live. I'm currently downloading the Beta 2 to test myself. I'm curious if the changes recorded by the DataContext will be then sent to the database in the same order that they happen (not like DataSet/TableAdapters where they were sent one table after the other, from top to down in each table). I just HOPE that it records changes and applies them in the appropriate order!

Glad you enjoyed the series! In terms of how updates are sent back to the database, LINQ to SQL uses the PK/FK relationships to correctly send the updates in the appropriate order.

Hope this helps,  
Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 31, 2007 4:34 AM by [ScottGu](#)

Hi Travis,

>>>>>> Scott (or anyone) Can we use the LINQ to SQL designer with an external mapping file and external class files for the entities? Like the old Class Designer?

Yes - you can have the mappings stored in an external mapping file, or have them encoded as attributes on the actual entity classes.

>>>>>> Also what is best practice for the DataContext class in an ASP.NET environment, should we store it in a HttpContext object or instantiate everytime we use it? Per user? One per application?

In general I'd recommend instantiating it per use, or per request. You don't want to cache it per application, since changes are tracked - and you typically don't want to batch up multiple user's updates together.

Hope this helps,  
Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 31, 2007 10:21 AM by [Travis](#)

>>>>>>Yes - you can have the mappings stored in an external mapping file, or have them encoded as attributes on the actual entity classes.

Thanx for the Reply Scott, but can we use the \*Designer\* with an external mapping file? And class files? I can't seem to get it to work. It seems to lump all the classes and mappings inside the DataClasses1.designer.cs file.

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 31, 2007 3:22 PM by Todd

Is there a way to add unmapped properties to a mapped class? I've got a mapped class, Person, and am trying to add a property in an external partial class definition - but I get a 'Binding Error'. Is there perhaps an attribute I need to apply to the property?

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, July 31, 2007 6:02 PM by Eric Newton

Scott, can you please address Winston Fassett's comment?

BoundFields cannot work with object graphs. And now that object graphs will be coming from LinqDatasources, we're given unusable scenarios.

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 01, 2007 1:53 AM by [ScottGu](#)

Hi Travis,

>>>>>> Thanx for the Reply Scott, but can we use the \*Designer\* with an external mapping file?

I just checked with the LINQ to SQL team, and they said that the designer in the final shipping version of VS 2008 will not support external mapping files (instead you'd need to use the command-line SQLMetal.exe utility that ships with VS 2008 to-do this). They are, however, looking at enabling a separate download that enables this mapping support within the designer in the future though.

Hope this helps,  
Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 01, 2007 1:57 AM by [ScottGu](#)

Hi Todd,

>>>>>> Is there a way to add unmapped properties to a mapped class? I've got a mapped class, Person, and am trying to add a property in an external partial class definition - but I get a 'Binding Error'. Is there perhaps an attribute I need to apply to the property?

Any chance you could send me this scenario in email (scottgu@microsoft.com)? I can then loop you in with someone from the LINQ to SQL team to have them investigate why you might be getting this error.

Thanks!  
Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 01, 2007 1:58 AM by [ScottGu](#)

Hi Eric,

>>>>>> BoundFields cannot work with object graphs. And now that object graphs will be coming from LinqDatasources, we're given unusable scenarios.

Unfortunately that is the case today (and I would like to see it supported in the future - although I don't know when). What you can do, though, is convert a GridView's column to be a TemplateColumn and then use the <%# Eval("property.subproperty") %> syntax to traverse the object graph.

Hope this helps,  
Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 01, 2007 3:27 AM by [AJ.NET](#)

Scott,

great post (as always). One question though. (Since all other comments targeted at the data part, I would like to focus on the UI part, just for a change ;-))

While I welcome the fact that business rules are enforced in the business layer (rather than UI) your proposed way of error handling (nothing new actually) does not nicely interface with the validator infrastructure. Not the part that does the validation of course, but the validator features used for displaying validation errors.

I take it that there is no change to ASP.NET in that respect, e.g. validators evaluated only server side (kind of 2-phase validation), validators acting on multiple fields, or validators to catch certain exceptions and present them like any other validator does (perhaps a validator that uses the OnValidate method of the data model).

The consequence is that the user has to pass two different validations, leading to 2 or more roundtrips (if done serverside): ASP.NET validators enforcing proper input (all at once) and business validation afterwards telling him that the proper input is still invalid (using an exception, i.e. the next rule is already waiting to fire). Also similar feedback (from the users' perspective) is presented differently (e.g. validation summary + label for exceptions).



This is not entirely satisfactory.

Regards,  
Alexander

<#> , possibly the perfect server control &laquo; CodeRaptor

Wednesday, August 01, 2007 2:41 PM by [, possibly the perfect server control « CodeRaptor](#)

Pingback from , possibly the perfect server control &laquo; CodeRaptor

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Wednesday, August 01, 2007 4:37 PM by Todd

Scott,

>>>>>> Any chance you could send me this scenario in email (scottgu@microsoft.com)? I can then loop you in with someone from the LINQ to SQL team to have them investigate why you might be getting this error.

In trying to distill the problem down to a simple scenario to send, I found the problem. The problem was not in trying to add a property through a separate partial class declaration, but trying to use that additional property in an orderby clause in a LINQ query. I realize that LINQ to SQL performs the sorting at the database level and so it makes sense to not be able to sort by an unmapped property.

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Wednesday, August 01, 2007 4:43 PM by [ScottGu](#)

Hi Todd,

>>>>>>> In trying to distill the problem down to a simple scenario to send, I found the problem. The problem was not in trying to add a property through a separate partial class declaration, but trying to use that additional property in an orderby clause in a LINQ query. I realize that LINQ to SQL performs the sorting at the database level and so it makes sense to not be able to sort by an unmapped property.

Ahh - that makes sense. Note that one option you can use is to perform your initial query against the properties in database, and then call "ToList()" on the returned sequence. This will then materialize a local instance of the data. You could then do a second LINQ query against these results (which you can because LINQ can query any local IEnumerable sequence) and use your non-mapped property as part of that search.

Hope this helps,

Scott

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Wednesday, August 01, 2007 8:31 PM by [ramon.duraes](#)

Hi Scott

New Gridview + LinqDasource has option for insert data ?

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Wednesday, August 01, 2007 10:09 PM by [Stefan Olson](#)

Scott,

Thanks for your blog. Very helpful. I'm loving using Linq. Datasets drove me nuts!

I do have a problem though. I get this exception when calling DetailsView.InsertItem:

System.ArgumentException was unhandled by user code

Message="Object of type 'System.Int32' cannot be converted to type 'System.Int16'."

As you can see from the stack trace below, it's in the LinqDataSource. I've changed all smallints to ints in the DB \*&updated the DBML), but that didn't fix it. The odd thing is it worked fine yesterday....

Clearly it may be my problem, but how do I figure out what field it is?

...Stefan

StackTrace:

```
at System.ComponentModel.ReflectPropertyDescriptor.SetValue(Object component, Object value)
at System.Web.UI.WebControls.LinqDataSourceView.BuildDataObject(Type dataObjectType, IDictionary inputParameters)
at System.Web.UI.WebControls.LinqDataSourceView.BuildInsertDataObject(Object table, IDictionary values)
at System.Web.UI.WebControls.LinqDataSourceView.ExecuteInsert(IDictionary values)
at System.Web.UI.DataSourceView.Insert(IDictionary values, DataSourceViewOperationCallback callback)
at System.Web.UI.WebControls.DetailsView.HandleInsert(String commandArg, Boolean causesValidation)
at System.Web.UI.WebControls.DetailsView.InsertItem(Boolean causesValidation)
at controls_memberdetails.SaveDetails() in c:\websites\newhiremystuff\controls\memberdetails.ascx.cs:line 65
at member_favourites.Button1_Click(Object sender, EventArgs e) in c:\websites\newhiremystuff\member\favourites.aspx.cs:line 22
at System.Web.UI.WebControls.Button.OnClick(EventArgs e)
at System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument)
at System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument)
at System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument)
at System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData)
at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
```

InnerException:

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, August 02, 2007 12:25 AM by [Stefan Olson](#)

Further to my previous cast issue, I was able to fix it if I changed all smallint to int and all smallmoney to int in my DB table (not really a fix :-)). If I have a smallmoney (AccountBalance) in my table, it fails when binding. Details below. If I change it back to int, all is well...

I'll try to figure what causes the previous issue tomorrow.

...Stefan

Exception Details: System.InvalidCastException: Specified cast is not valid.

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

[InvalidCastException: Specified cast is not valid.]

System.Data.SqlClient.SqlBuffer.get\_Int32() +980012

System.Data.SqlClient.SqlDataReader.GetInt32(Int32 i) +39

Read\_DynamicClass1(ObjectMaterializer`1 ) +3557

System.Data.Linq.SqlClient.ObjectReader`2.MoveNext() +26

System.Collections.Generic.List`1..ctor(IEnumerable`1 collection) +441

System.Linq.Enumerable.ToList(IEnumerable`1 source) +54

[TargetInvocationException: Exception has been thrown by the target of an invocation.]

System.RuntimeMethodHandle.\_InvokeMethodFast(Object target, Object[] arguments, SignatureStruct& sig, MethodAttributes methodAttributes, RuntimeTypeHandle typeOwner) +0

System.RuntimeMethodHandle.InvokeMethodFast(Object target, Object[] arguments, Signature sig, MethodAttributes methodAttributes, RuntimeTypeHandle typeOwner) +72

System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo culture, Boolean skipVisibilityChecks) +371

System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo culture) +29

System.Web.UI.WebControls.LinqDataSourceView.ToList(IQueryable query, Type dataObjectType) +121

System.Web.UI.WebControls.LinqDataSourceView.ExecuteSelectQuery(LinqDataSourceSelectEventArgs selectEventArgs, Object selectResult, Object table, Boolean storeOriginalValues) +328

System.Web.UI.WebControls.LinqDataSourceView.ExecuteSelect(DataSourceSelectArguments arguments) +500

System.Web.UI.DataSourceView.Select(DataSourceSelectArguments arguments, DataSourceViewSelectCallback callback) +17

System.Web.UI.WebControls.DataBoundControl.PerformSelect() +149

System.Web.UI.WebControls.BaseDataBoundControl.DataBind() +70

System.Web.UI.WebControls.FormView.DataBind() +4

System.Web.UI.WebControls.BaseDataBoundControl.EnsureDataBound() +82

System.Web.UI.WebControls.FormView.EnsureDataBound() +163

System.Web.UI.WebControls.CompositeDataBoundControl.CreateChildControls() +69

System.Web.UI.Control.EnsureChildControls() +87

System.Web.UI.Control.PreRenderRecursiveInternal() +50

System.Web.UI.Control.PreRenderRecursiveInternal() +170

System.Web.UI.Control.PreRenderRecursiveInternal() +170

System.Web.UI.Control.PreRenderRecursiveInternal() +170

System.Web.UI.Control.PreRenderRecursiveInternal() +170

System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +2041

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, August 02, 2007 12:47 AM by Troy Hunt

Hi Scott, in terms of utilising the "where" clause in the LinqDataSource, how do you handle type mismatches between string and guid? For example, I have a "where" clause that takes a selected value from a drop down list (which I assume is treated as a string although is a guid in the DB) and uses it as a control parameter to filter a selection from my LINQ class. This results in an error like this: "Operator '==' incompatible with operand types 'Guid' and 'String'".

Obviously doesn't handle any implicit conversions between the two. Any other suggestions in terms of a "no code" approach to this?

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, August 02, 2007 7:03 PM by Phil

Is it, or will it be, possible to do declarative databinding with the LinqDataSource to a Stored Procedure, as well as a Table? I'm working with Beta 2 now that it has dropped, so this is my first attempt at Linq. All I'm seeing is the ability to bind to a Table.

Thanks!

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Friday, August 03, 2007 2:46 AM by Dion

Hi Scott,

I will be developing a web application using Beta 2 version and am currently researhing and reading as much as possible to get me started on the new stuff.

I seen quite a few examples on selecting and bing in ASP.NEt using linq to SQL.

I like the tutorial on this page.

I would like to have the source code. Could you please send me the code on startatrix@hotmail.com or point me to a download link.

Thanks

# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Friday, August 03, 2007 4:09 AM by [ScottGu](#)

Hi ramon.duraes,

>>>>>> New Gridview + LinqDasource has option for insert data ?

The GridView doesnt' support insert directly - although you can use a detailsview control with it to enable this.

The new Listview control does also support defining an insert template.

Hope this helps,

Scott



**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Friday, August 03, 2007 4:16 AM by [ScottGu](#)

Hi Stefan,

>>>>> Further to my previous cast issue, I was able to fix it if I changed all smallint to int and all smallmoney to int in my DB table (not really a fix :-)). If I have a smallmoney (AccountBalance) in my table, it fails when binding. Details below. If I change it back to int, all is well...

Would you be able to send me an email (scottgu@microsoft.com) with details of this? I can then loop you in with the developer who owns the LinqDataSource control and we can figure out how to get this working.

Thanks,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Friday, August 03, 2007 4:16 AM by [ScottGu](#)

Hi Phil,

>>>>> Is it, or will it be, possible to do declarative databinding with the LinqDataSource to a Stored Procedure, as well as a Table? I'm working with Beta 2 now that it has dropped, so this is my first attempt at Linq. All I'm seeing is the ability to bind to a Table.

Right now you'll only be able to-do purely declarative binding to Tables. What you can do, though, is handle the selecting event and use it to call any method on the DataContext that you want.

Hope this helps,

Scott

**# LinqDataSource And How You Can \*Maximize\* Roundtrips to Your Database**

Friday, August 03, 2007 1:04 PM by [David Hayden \[MVP C#\]](#)

I was reading Scott Guthrie's excellent series of posts on LINQ when I came to the following post

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Saturday, August 04, 2007 3:52 AM by [chen](#)

I get the following error when I try to update a row in the GridView:

Exception Details: System.Data.Linq.ChangeConflictException: Row not found or changed

**# LinqDataSource and ListView - a Dream Team - Andreas Kraus: C#, ASP.NET, Silverlight**

Sunday, August 05, 2007 1:41 PM by [LinqDataSource and ListView - a Dream Team - Andreas Kraus: C#, ASP.NET, Silverlight](#)

Pingback from LinqDataSource and ListView - a Dream Team - Andreas Kraus: C#, ASP.NET, Silverlight

**# &#8230; LinqDataSource and ListView - a Dream Team - Andreas Kraus: C# &#8230;**

Sunday, August 05, 2007 5:46 PM by [» LinqDataSource and ListView - a Dream Team - Andreas Kraus: C# ...](#)

Pingback from &#8230; LinqDataSource and ListView - a Dream Team - Andreas Kraus: C# &#8230;

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Monday, August 06, 2007 1:38 AM by [ScottGu](#)

Hi Chen,

>>>>> I get the following error when I try to update a row in the GridView: Exception Details: System.Data.Linq.ChangeConflictException: Row not found or changed

This could occur because the underlying row in the database was updated between the original value being retrieved by the GridView and the actual update being applied. LINQ to SQL by default automatically enables this "optimistic concurrency" check for you to ensure that you don't accidentally overwrite other changes without realizing it.

In a future LINQ to SQL blog post I'll talk about techniques you can use to handle these cases and prompt the user as to whether they really want to apply their changes.

Thanks,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Monday, August 06, 2007 11:27 AM by [Shloma Baum](#)

Scott,

How does Link to SQL fit in a 3-tier environment? since what you expose is most probably a method to retrieve a Product class, so obviously you will not be able to get to the Categories class as it doesn't have access to the DataContext? or maybe I'm missing something simple? so how will the LinkDataSource know how to apply paging etc., so basically we fall back again to use a client/server approach.

This does not only apply to this excellent post but rather an overall question on Link in regards to remoting.

Regards,

Shloma

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, August 07, 2007 1:35 AM by Mike

Scott,

Is it possible to perform a sub-query using the LinqDataSource? For example, say I want to select a list of Customers as well as the customer's last order date from the Orders table. Is this possible create this query with the LinqDataSource designer alone?

Thank you,

Mike

**# ASP.NET 3.5 controls**

Tuesday, August 07, 2007 10:34 AM by [Daniel Moth](#)

ASP.NET 3.5 controls

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Tuesday, August 07, 2007 5:26 PM by [Arnaud Weil](#)

Excellent article Scott. And excellent control. I simply love when things are that easy. It reminds me of ObjectDataSource and typed DataSet, except that here we have a cleaner model than TableAdapters.

Now, I'm eager to seeing the same exact simplicity in WPF...

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 08, 2007 1:05 AM by [ScottGu](#)

Hi Mike,

>>>>>> Is it possible to perform a sub-query using the LinqDataSource? For example, say I want to select a list of Customers as well as the customer's last order date from the Orders table. Is this possible create this query with the LinqDataSource designer alone?

To perform a custom sub-query, you'll want to handle the LinqDataSource's Selecting event. I'll show how to do this in an upcoming blog post. :-)

Thanks,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 08, 2007 8:49 AM by Greg McCarty

Beta2 doesn't seem to have the LinqDataSource control in any toolbox? What am I missing?

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 08, 2007 9:02 AM by Greg McCarty

Also, the GridView smart tasks in Beta2 don't have the edit or delete options...

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 08, 2007 9:04 AM by Greg McCarty

Oops. Never mind about the editing and deleting option missing. I had to build once before they showed up...

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 08, 2007 10:49 AM by Greg McCarty

Orcas Beta2 - The EditItemTemplate example doesn't work for me. The SelectedValue attribute is not available. A change followed by an update does not change the underlying value.

Any thoughts?

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 08, 2007 10:52 AM by [ScottGu](#)

Hi Greg,

>>>>>> Oops. Never mind about the editing and deleting option missing. I had to build once before they showed up...

Cool - just to confirm, you have the LinqDataSource and GridView working ok now?

Thanks,

Scott

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Wednesday, August 08, 2007 2:40 PM by Greg McCarty

>>> Cool - just to confirm, you have the LinqDataSource and GridView working ok now?

Sort of. The example about changing the Supplier and Category seems to be off. The asp:DropDownList doesn't support the 'SelectedValue' attribute. So I don't know how to update those fields (or, more importantly, that style of data indirection).

Any thoughts?

GregM

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Thursday, August 09, 2007 8:12 AM by [daCodez](#)

I'm wondering if we'll be able to do things like query excel files with linq.

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Thursday, August 09, 2007 9:13 AM by manu

Hi Scott,

linq 2 sql is great!

I started with some example, so I created a business layer that uses Linq2SQL.

In this scenario I'm using an ObjectDataSource instead of a LinqDataSource.

I think that in a layered application - with a BLL - Linq2SQL is not usefull!

Is it true?

Thanks

Manu

**# re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)**

Thursday, August 09, 2007 10:22 AM by Max

I have a very simple database, no timestamp fields, I am the only user updating the database.

I have a LinqDataSource bound to a GridView and when I goto edit a row I get the following error:

Row not found or changed.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.Linq.ChangeConflictException: Row not found or changed.

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, August 09, 2007 1:48 PM by [ScottGu](#)

Hi Manu,

>>>>>> I started with some example, so I created a business layer that uses Linq2SQL. In this scenario I'm using an ObjectDataSource instead of a LinqDataSource. I think that in a layered application - with a BLL - Linq2SQL is not usefull!

I think you'll actually find LINQ to SQL extremely useful with ObjectDataSource scenarios. In these cases you can use LINQ to SQL as your DAL layer.

Hope this helps,

Scott

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, August 09, 2007 1:49 PM by [ScottGu](#)

Hi Greg,

>>>>> Sort of. The example about changing the Supplier and Category seems to be off. The asp:DropDownList doesn't support the 'SelectedValue' attribute. So I don't know how to update those fields (or, more importantly, that style of data indirection).

The intellisense for the <asp:dropdownlist> doesn't shows the SelectedValue property, but it is there on the control. :-) If you still have problems send me an email with your code-snippet and I can help get it working.

Thanks,

Scott

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Thursday, August 09, 2007 1:50 PM by [ScottGu](#)

Hi Max,

>>>>>> I have a LinqDataSource bound to a GridView and when I goto edit a row I get the following error:

Do you have the DataKeyNames attribute set on the GridView? It could be that this is empty - in which case the primary key loopup index isn't being saved.

Thanks,

Scott

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Friday, August 10, 2007 5:43 PM by Connie Phan

Hi Scott,

I'm having issue with the dbml file. After I defined the data model, then close the file, when I open it again, it's now an XML file instead of the graphic interface with the tables. Am I doing something wrong or is this a bug? Thanks!!

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Saturday, August 11, 2007 5:08 AM by Howard

I have been monitoring LINQ to SQL since last year when it was DLINQ and now it's in Beta2 I've been experimenting with it. I had high hopes I could drop my own data access layer and use LINQ to SQL.

Sadly LINQ to SQL does not make the grade.

The first major issue is that the designer and DBML file do not support refresh of the tables/columns/views. How on earth can you launch this product without this feature??? My current DB has 120 tables and about 40 views. Am I supposed to keep deleting and re-adding objects when I change them?

Second, there isn't enough extensibility in the LINQ to SQL classes to enable any meaningful app-level security to be built into it. My datalayer does this so queries automatically get filters added to any table access, meaning the data layer enforces security, not the end user app.

But discussing with others on MSDN forums it does not seem that there is any way of doing this - the DataContext does not have a hooks to adjust how tables/views are accessed and there is no way to override this.

I realise LINQ to SQL is a big step forward and for someone who is happy to implement security in the app/ui code (or has no security need) then it's really good. Shame it didn't quite make that last hurdle..

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Monday, August 13, 2007 1:28 AM by [ScottGu](#)

Hi Howard,

>>>>>> The first major issue is that the designer and DBML file do not support refresh of the tables/columns/views. How on earth can you launch this product without this feature??? My current DB has 120 tables and about 40 views. Am I supposed to keep deleting and re-adding objects when I change them?

You can delete and re-add individual tables using the designer. Alternatively, you can use the command-line sqlmetal.exe utility which can recreate the objects automatically.

>>>>>> Second, there isn't enough extensibility in the LINQ to SQL classes to enable any meaningful app-level security to be built into it. My datalayer does this so queries automatically get filters added to any table access, meaning the data layer enforces security, not the end user app.

There are actually Load() methods that you can override on the DataContext class that provides interception points when you query data. Alternatively, you might want to create a class that encapsulates the DataContext and through which your UI and other business logic retrieves data. You could then add your security filtering logic in these methods.

Hope this helps,

Scott

<#> re: LINQ to SQL (Part 5 - Binding UI using the ASP:LinqDataSource Control)

Monday, August 13, 2007 1:30 AM by [ScottGu](#)

Hi Connie,



>>>>> I'm having issue with the dbml file. After I defined the data model, then close the file, when I open it again, it's now an XML file instead of the graphic interface with the tables. Am I doing something wrong or is this a bug? Thanks!!

That is pretty odd - did you change the file-type at all?

Thanks,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Monday, August 20, 2007 3:27 AM by shashank sinha

hi i m trying to create views for the tables and then work with them but i m unable to achieve any successful result.

Can u guide me through to some blogs or articles where in i can get some help

Thanks and regards,

shashank sinha

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Monday, August 20, 2007 11:11 AM by [ScottGu](#)

Hi Shashank,

>>>>>>>Can u guide me through to some blogs or articles where in i can get some help

Here is the LINQ online forum where people should be able to help with any LINQ related questions: [forums.microsoft.com/.../ShowForum.aspx](http://forums.microsoft.com/.../ShowForum.aspx)

Thanks,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Thursday, August 23, 2007 5:53 PM by Wesley Raven

Hi Scott,

Thank you very much for your incredibly useful posts, and your obvious hard work on answering all of these questions. Your time is greatly appreciated by all Im sure.

I have run into a snag with the LinqDataSource I hope someone can assist me with; When the source of one of my where clause parameters is a control and my datatype is an integer I get this error:

"Operator '==' incompatible with operand types 'Int32' and 'Object'"

I have tried binding to different types of controls like a HiddenField, or a Label, and all result in the same error. I have tried manually selecting the datatype of the parameter as an Int32 without any luck. The only way I can bind to an integer type is if the source for the where parameter is in the querystring, or the session; not a control.

I am currently working around this by using the Selecting event on the LinqDataSource, and then doing my simple query (for example, to get a single user, "e.result = From t in DC Where t.UserID = Convert.ToInt32(UserIDLabel.Text)").

Can anyone tell me where I'm going wrong when I try to assign the source of the where parameter directly to a control? Do I have to do some manual datatype conversion somewhere? Thanks!

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Tuesday, August 28, 2007 1:42 AM by Kevin

This is so crazy-cool it's actually making me vaguely ill with happiness. That's a weird feeling.

Thanks very very much for making things this straight forward and easy to grasp, I think you may have just singlehandedly made my job (and by extension, my life) much easier. :)

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Wednesday, August 29, 2007 1:08 PM by Mark Henke

Why can one not drag a stored proc onto the ORM and have it perhaps (optionally) create the Object based on the output of that sp as well as creating the method that returns an anonymous type?

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Thursday, August 30, 2007 3:23 AM by [YellowWee](#)

hi,I'm tring to set "DropDownList" in the "EditItemTemplate",and In VS2008 Beta2's "Properties window",I can't find the "SelectedValue" and "SelectedIndex" properties,also can't find with "intellisense".

this confused me,whether or not .net 3.5 don't have this properties.Not bad I found it in MSDN Library later.

may be it's litter bug in vs2008 beta2,hope it help you to perfect VS2008.

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Saturday, September 08, 2007 12:52 PM by [ScottGu](#)

Hi Mark,

>>>>>> Why can one not drag a stored proc onto the ORM and have it perhaps (optionally) create the Object based on the output of that sp as well as creating the method that returns an anonymous type?

This is actually what the designer does by default. I discuss this more in my blog post on SPROCs here: [weblogs.asp.net/.../linq-to-sql-part-6-retrieving-data-using-stored-procedures.aspx](http://weblogs.asp.net/.../linq-to-sql-part-6-retrieving-data-using-stored-procedures.aspx)

Hope this helps,

Scott

[# re: LINQ to SQL \(Part 5 - Binding UI using the ASP:LinqDataSource Control\)](#)

Thursday, September 13, 2007 5:07 AM by [johnzered](#)

Hi!

You are writing great articles!

I have one problem with defining my ORM, I need to connect to two databases. That's ok in my development enviroment when i set the data source on the classes in another database the problem appears in production when our customers may have another connectionstring and then ORM won't automaticallay find my database.

Example:

On my classes from the second databases I change the datasource property to databasename.dbo.tablename

This works great but then if I try my application in production the second database might be something else eg anotherdbname.dbo.tablename. Then when I run the application I get an error that it can't find databasename.dbo.tablename.

I hope you understand my problem. Do you know a way to solve this by defining it in web.config or somewhere else?  
//johnzered

# Using the LinqDataSource: A practical look

Tuesday, September 18, 2007 11:30 PM by [Todd Anglin's Code Campground](#)  
One of the new controls coming in the .NET 3.5/Visual Studio 2008 releases later this year is the LinqDataSource

# campground &raquo; Using the LinqDataSource: A practical look

Thursday, October 11, 2007 5:33 PM by [campground » Using the LinqDataSource: A practical look](#)  
Pingback from campground &raquo; Using the LinqDataSource: A practical look

# Linq to SQL ed il binding usando il controllo LinqDataSource

Sunday, December 09, 2007 7:00 AM by [{ The Smoking Code }](#)  
Linq to SQL ed il binding usando il controllo LinqDataSource

# Linq to SQL ed il binding usando il controllo LinqDataSource

Sunday, December 09, 2007 7:43 AM by [Mighell's Mobile Blog](#)  
In questi giorni, stò implementando una applicazione non eccessivamente complessa in ASP.Net e, dato

# LinQ操作汇总（From CSharpSamples）

Monday, December 17, 2007 12:50 AM by [edobnet](#)  
基本方法可以查看这里[weblogs.asp.net/.../linq-to-sql-part-5-binding-ui-using-the-...](#)

# Overloaded again

Sunday, January 27, 2008 7:05 PM by [Paul Speranza's .Net Life](#)  
I have just read Keith Harvey's post Slow Down Microsoft, I Can't Take It! about the pace the development

# LinqDataSource exceptions

Thursday, January 31, 2008 9:05 PM by [Raj Kaimal](#)  
Prerequisite: LinqDataSource &amp; SqlDataSource Master/Details When working with the LinqDataSource

# ASP.NET ListView control and LinqDataSource - Display data without a line of code

Wednesday, February 06, 2008 6:07 PM by [Janko`s Blog](#)  
The new ASP.NET ListView control is a powerfull control that encapsulates the best features from ASP

# Detailed Look: Key components in LINQ to SQL and their Key Roles

Thursday, February 28, 2008 9:26 AM by [Corey Gaudin](#)  
As part of this blog, I plan to have an on-going set of articles that takes a detailed look into some

# Jon Kruger&#8217;s Blog &raquo; LINQ to SQL talk stuff

Friday, February 29, 2008 7:03 PM by [Jon Kruger's Blog » LINQ to SQL talk stuff](#)  
Pingback from Jon Kruger&#8217;s Blog &raquo; LINQ to SQL talk stuff

# madhavrao&#8230;

Saturday, March 08, 2008 2:27 AM by [madhavrao...](#)  
Pingback from madhavrao&#8230;

# Lets dig in to some more detail on LinqDataSource Control. &laquo; Rangatharan&#8217;s Weblog

Saturday, March 08, 2008 8:12 AM by [Lets dig in to some more detail on LinqDataSource Control. « Rangatharan's Weblog](#)  
Pingback from Lets dig in to some more detail on LinqDataSource Control. &laquo; Rangatharan&#8217;s Weblog

# 转:LinQ操作汇总（From CSharpSamples）

Sunday, March 23, 2008 1:46 AM by [Winner.Net\(2007\)](#)  
LinQ操作汇总（FromCSharpSamples）  
基本方法可以查看这里[weblogs.asp.net/.../linq-to-s...](#)

# LINQ here, LINQ there, LINQ everywhere | two geeks&#8230;

Monday, April 21, 2008 11:47 PM by [LINQ here, LINQ there, LINQ everywhere | two geeks...](#)  
Pingback from LINQ here, LINQ there, LINQ everywhere | two geeks&#8230;

# ruSpiegel.net - ?????????????? ?????????????? ?????????????? &raquo; ?????????? ??????????  
&raquo; ????????????????????????????????????? LINQ to SQL (????????????? 9 -  
?? LINQ ?????????????????????? ?? ?????

Wednesday, May 14, 2008 7:41 PM by [ruSpiegel.net - ?????????????????? ?????????????????? ?????????????????? » ?????????? ?????????? » ????????????????????????????????????? LINQ to SQL \(????????????? 9 - ?????????????????????????????????????? ?????????????????????????????????????? LINQ ?????????????????????? ?? ??????????????????????????????????\)](#)

Pingback from ruSpiegel.net - ?????????????? ?????????????? ?????????????? &raquo; ?????????? ?????????? &raquo; ?????????????????????????????? LINQ to SQL (?????????? 9 - ?????????????????????????? ?????????????????????????????? LINQ ?????????????????? ?? ?????????????????? ??????????????????)

# ?????????????? &raquo; Blog Archive &raquo; LINQ to SQL?????Part 5 - Binding UI using the ASP:LinqDataSource Control

Sunday, July 06, 2008 5:14 AM by ?????????????? » Blog Archive » LINQ to SQL?????Part 5 - Binding UI using the ASP:LinqDataSource Control

Pingback from ?????????????? &raquo; Blog Archive &raquo; LINQ to SQL?????Part 5 - Binding UI using the ASP:LinqDataSource Control

# entitydatasource toolbox

Wednesday, July 09, 2008 10:38 PM by entitydatasource toolbox

Pingback from entitydatasource toolbox

# GridView with paging and sorting and LINQ &laquo; Peter’s log

Tuesday, August 12, 2008 12:32 PM by GridView with paging and sorting and LINQ « Peter’s log

Pingback from GridView with paging and sorting and LINQ &laquo; Peter’s log

# Composants pour le binding avec Linq et avec Entity

Friday, October 03, 2008 4:22 PM by Composants pour le binding avec Linq et avec Entity

Pingback from Composants pour le binding avec Linq et avec Entity

# How to handle type exception in LINQ Validation rule

Monday, October 06, 2008 5:38 PM by LINQ in Action roller

Hi I'm trying to re-write Scott's code from VB to C# for his LINQ validation rule. Basically, I have

# Baby name meaning and origin for Briscoe

Wednesday, December 24, 2008 8:44 PM by Baby name meaning and origin for Briscoe

Pingback from Baby name meaning and origin for Briscoe

# Adventures with LINQ to SQL

Friday, January 16, 2009 12:39 AM by Ahmed's Random Expressions

A while back I got the opportunity to use LINQ to SQL (L2S) for an internal web application. What is

# 6. LINQ to SQL (Part 6 - Retrieving Data Using Stored Procedures)

Wednesday, February 18, 2009 8:55 PM by Sample Weblog

Over the last few weeks I’ve been writing a series of blog posts that cover LINQ to SQL. LINQ to