6.2.0 – CRUD (Product Maintenance)

Introduction

In previous lesson you created the methods in the **ProductController.cs** for **C**reate, **R**ead, **U**pdate, and **D**elete operations on a database. In this lesson you will connect these methods to an ObjectDataSource, which will be connected to a ListView Control

Supporting Files

On Moodle, in the **Code Files (6.2.0)** folder, you will see:

- `ProductMaintenance_aspx.txt`
- `AdminMenu.txt`

These files will be needed for this lesson.

Web Site Setup

Create a folder on your web site called **Admin**. In this folder add the following web form (with master) called **ProductMaintenance.aspx**.

ListView Form (ProductMaintenance.aspx)

After this web form is created by Visual Studio, replace the contents of the file with **ProductMaintenance_aspx.txt**. Add the **MessageUserControl** to this web form like you did in Lesson 5.1.0. <mark>Do not forget to add this control in the empty row div</mark>.

Site Navigation Menu

With the new web form added, the navigation menu in the `Site.master` file needs to be updated. Add the contents of the **AdminMenu.txt** to a blank line before the **About** menu item.

ObjectDataSource Controls

As with previous lessons, the ODS controls need to be configured so that they are available to the data controls of the web form.

CategoryListODS

This ODS control will use the `GetAllCategories()` method of the **CategoryController.cs**.

Listing 1: CategoryListODS
```
<asp:ObjectDataSource ID="CategoryListODS" runat="server"
    OldValuesParameterFormatString="original_{0}"
    SelectMethod="GetAllCategories"
    TypeName="eStoreSystem.BLL.CategoryController">
</asp:ObjectDataSource>
```

SupplierListODS

This ODS control will use the `GetAllSuppliers()` method of the
`SupplierController.cs`.

Listing 2: SupplierListODS

```
<asp:ObjectDataSource ID="SupplierListODS" runat="server"
    OldValuesParameterFormatString="original_{0}"
    SelectMethod="GetAllSuppliers"
    TypeName="eStoreSystem.BLL.SupplierController">
</asp:ObjectDataSource>
```

ProductListODS

This ODS control will use the methods added to the **`ProductController.cs`** in Lesson 6.1.0.
Unlike previous lessons, we will need to select the **UPDATE**, **INSERT**, and **DELETE** tabs on the
Define Data Methods screen of the Configure Data Source wizard.

Figure 1: SELECT Method for ODS

Figure 2: UPDATE Method for the ODS

Figure 3: INSERT Method for the ODS

Figure 4: DELETE Method for the ODS

When you press Finish, your ODS code should look like the following (after some formatting for readability):

Listing 3: ProductListODS

```
<asp:ObjectDataSource ID="ProductListODS" runat="server"
    DataObjectTypeName="eStoreSystem.Data.Entities.Product"
    DeleteMethod="DeleteProduct"
    InsertMethod="AddProduct"
    OldValuesParameterFormatString="original_{0}"
    SelectMethod="ListAllProducts"
    TypeName="eStoreSystem.BLL.ProductController"
    UpdateMethod="UpdateProduct">
</asp:ObjectDataSource>
```

As we will be performing change operations on the database, we need to add some error checking to this ODS. There are only 3 methods that change data in the database (Insert, Update, and Delete) so we need to perform error checking for these.

Since all three database operations need error checking, we will add a new method to the code behind page, **ProductMaintenance.aspx.cs**, as shown below:

Listing 4: CheckForException()

```
protected void CheckForException(object sender, ObjectDataSourceStatusEventArgs e)
{
    MessageUserControl.HandleDataBoundException(e);
}//eom
```

Next, we need to add **OnInserted**, **OnUpdated**, and **OnDeleted** parameters to the ODS. Each of these will use the method coded in Listing 4 (above).
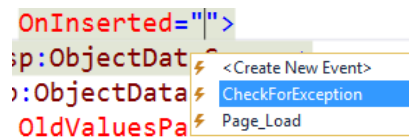

Figure 5: Add OnInserted="CheckForException"

Repeat this process for the other parameters and your ODS code should look like:

```
<asp:ObjectDataSource ID="ProductListODS" runat="server"
    DataObjectTypeName="eStoreSystem.Data.Entities.Product"
    DeleteMethod="DeleteProduct"
    InsertMethod="AddProduct"
    OldValuesParameterFormatString="original_{0}"
```

```
    SelectMethod="ListAllProducts"
    TypeName="eStoreSystem.BLL.ProductController"
    UpdateMethod="UpdateProduct"
    OnInserted="CheckForException"
    OnUpdated="CheckForException"
    OnDeleted="CheckForException">
</asp:ObjectDataSource>
```

ListView Control

As shown in Lesson 4.3.0, we can configure the ListView by setting its Data Source, then configuring its look. Unlike Lesson 4.3.0, we will use all the templates as the ODS has all the methods for **CRUD**. On the Configure ListView wizard, make sure your configuration is the same as that shown below (only the style can be different):
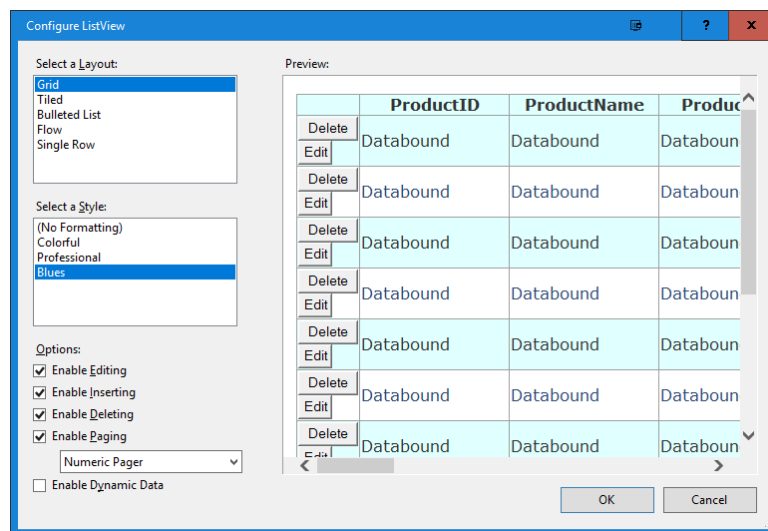
Figure 6: Configure ListView

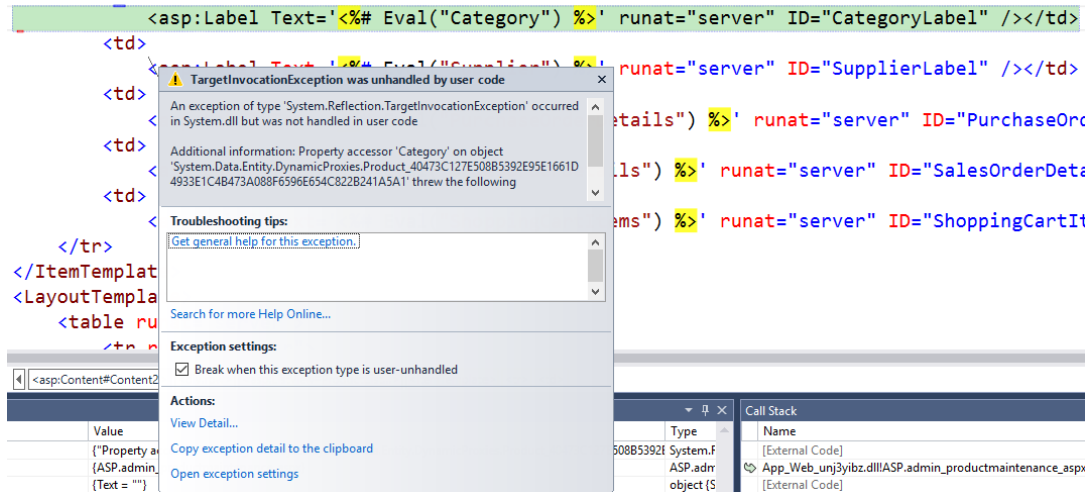If you run this web page now you will get an exception:

Figure 7: Navigation Property Exception

The exception occurs because the DbContext class is bringing back the Navigation properties along with all the properties of the Entity.
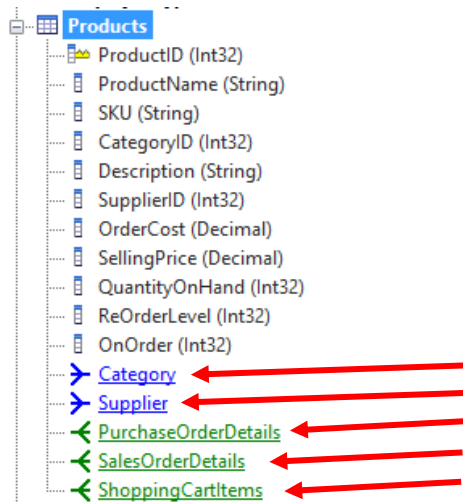
Figure 8: Products Entity in LINQPad

```
170            <td>
171                <asp:Label Text='<%# Eval("Category") %>' runat="server" ID="CategoryLabel" /></td>
172            <td>
173                <asp:Label Text='<%# Eval("Supplier") %>' runat="server" ID="SupplierLabel" /></td>
174            <td>
175                <asp:Label Text='<%# Eval("PurchaseOrderDetails") %>' runat="server" ID="PurchaseOrderDetailsLabel" /></td>
176            <td>
177                <asp:Label Text='<%# Eval("SalesOrderDetails") %>' runat="server" ID="SalesOrderDetailsLabel" /></td>
178            <td>
179                <asp:Label Text='<%# Eval("ShoppingCartItems") %>' runat="server" ID="ShoppingCartItemsLabel" /></td>
```

Figure 9: Code that Causes the Exception

This is an easy fix. All you need to do is to delete these rows from **all** the templates of the ListView. Once done, your output should look like:

Wired CRUD Demo (Product Maintenance)

	ProductID	ProductName	SKU	CategoryID	Description	SupplierID	OrderCost	SellingPrice	QuantityOnHand	ReOrderLevel	OnOrder
Delete Edit	1	Apple	F-Apple-01	1	Small apples	1	1.25	2.10	150	55	0
Delete Edit	2	Apple	F-Apple-02	1	Medium apples	1	1.35	2.20	140	45	0
Delete Edit	3	Apple	F-Apple-03	1	Large apples	1	1.45	2.60	130	40	0
Delete Edit	4	Banana	F-Banan-01	1	Organic bananas	15	2.34	3.99	40	20	0
Delete Edit	5	Banana	F-Banan-02	1	Imported bananas	1	1.74	2.38	67	70	0
Delete Edit	6	Pineapple	F-PineA-01	1	Imported Orgainc Pineapple	15	4.75	6.25	50	60	20
Delete Edit	7	Mango	F-Mango-01	1	Imported Orgainc Mango	15	2.34	3.50	40	35	0
Delete Edit	8	Tomato	V-Tomat-01	2	Garden Fresh Tomato	2	1.75	2.75	80	50	0
Delete Edit	9	Tomato	V-Tomat-02	2	Orgainc Fresh Tomato	15	2.25	3.25	30	20	0
Delete Edit	10	Carrot	V-Carrt-01	2	Garden Fresh Carrot	2	1.45	1.95	100	55	0
Insert Clear											

First 1 Last

© 2019 - eStore

Figure 10: ListView CRUD Output (1)

DO NOT Edit, Delete, or Update any records at this time!

Currently, the look is not the best; what does the `CategoryID=1`, or the `SupplierID=1`, mean? It would be nice to see the category name, and the supplier name. We can do this by modifying the `ItemTemplate`, `AlternatingItemTemplate`, `EditItemTemplate`, `InsertItemTemplate`, and the `SelectedItemTemplate` to use a `DropDownList` instead of a `Label`.

CategoryListDDL

This `DropDownList` control will use the **CategoryListODS** as its data source. It will also need to know which Category the Product belongs to. The code you need for this control is shown below:

Listing 5: CategoryListDDL

```
<asp:DropDownList ID="CategoryListDDL" runat="server" Enabled="false"
    DataSourceID="CategoryListODS"
    DataTextField="DisplayText"
    DataValueField="IDValue"
    SelectedValue='<%#Bind("CategoryID") %>'>
</asp:DropDownList>
```

> This parameter is not in the intellisense.

	ProductID	ProductName	SKU	CategoryID	D
Delete Edit	1	Apple	F-Apple-01	Fruit ▾	S
Delete Edit	2	Apple	F-Apple-02	Fruit ▾	M

Figure 11: Category DDL Added

SupplierListDDL

This `DropDownList` control will use the **SupplierListODS** as its data source. It will also need to know which Supplier this Product is from. The code you need for this control is shown below:

Listing 6: MediaTypeListDDL

```
<asp:DropDownList ID="SupplierListDDL" runat="server" Enabled="false"
    DataSourceID="SupplierListODS"
    DataTextField="DisplayText"
    DataValueField="IDValue"
    SelectedValue='<%#Bind("SupplierID") %>'>
</asp:DropDownList>
```

> The `SelectedValue` parameter is not in the intellisense.

	ProductID	ProductName	SKU	CategoryID	Description	SupplierID	
Delete Edit	1	Apple	F-Apple-01	Fruit ▾	Small apples	Best Western Fruit ▾	1
Delete Edit	2	Apple	F-Apple-02	Fruit ▾	Medium apples	Best Western Fruit ▾	1

Figure 12: Supplier DDL Added

Enabling DropDownLists

In making the previous changes, all the `DropDownList` controls were visible but disabled (not changeable). We might want to be able to change the Category and/or the Supplier of a product. With `Enabled="false"` this is not possible. You will need to remove this parameter from the `DropDownList`s in the `EditItemTemplate` and the `InsertItemTemplate`.

Primary Key Fields

If you press the Edit button on a row in the ListView, you will notice that you can change the `ProductID`, which is a Primary Key value. This is **NOT** allowed.
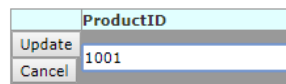
	ProductID
Update Cancel	1001

Figure 13: Editable Primary Key

Additionally, you will notice a `TextBox` for the Insert row. This, too, is **NOT** allowed.

Insert Clear	

Figure 14: Insert Primary Key Value

EditItemTemplate

Change the control from a `TextBox` to a `Label` for the `ProductID`:

```
<asp:Label Text='<%# Bind("ProductID") %>' runat="server" ID="ProductIDLabel" />
```

InsertItemTemplate

Remove the `TextBox` and use the code below:

```
<td> </td>
```


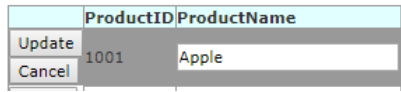Figure 15: Fixed EditItemTemplate

Figure 16: Fixed InsertItemTemplate

Finishing Touches

You may have noticed that the numeric page numbers may be hard to read. You can change this by changing the styling of the table row for the pager from:

```
<td runat="server" style="text-align: center; background-color: #5D7B9D; font-family:
Verdana, Arial, Helvetica, sans-serif; color: #FFFFFF">
```

To:

```
<td runat="server" style="text-align: center; background-color: #E0FFFF; font-family:
Verdana, Arial, Helvetica, sans-serif; color: #000000">
```
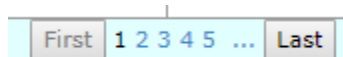
This will give you:

Figure 17: Modified Styling of the Navigation Footer

Now that we have DDL controls for Category and Supplier, we should change the headings in the **LayoutTemplate** to be:

```
<tr runat="server" style="background-color: #E0FFFF; color: #333333;">
    <th runat="server"></th>
    <th runat="server">ProductID</th>
    <th runat="server">ProductName</th>
    <th runat="server">SKU</th>
    <th runat="server">Category</th>
    <th runat="server">Description</th>
    <th runat="server">Supplier</th>
    <th runat="server">OrderCost</th>
    <th runat="server">SellingPrice</th>
    <th runat="server">QuantityOnHand</th>
    <th runat="server">ReOrderLevel</th>
    <th runat="server">OnOrder</th>
</tr>
```

The next step is **VERY IMPORTANT**! You will not be able to delete a record from the database because the ListView does not know about the Primary Key. To enable deleting you need to modify the ListView code to be:

```
<asp:ListView ID="ProductListLV" runat="server" DataSourceID="ProductListODS"
InsertItemPosition="LastItem" DataKeyNames="ProductID">
```

TEST

To test the CRUD operations, beyond displaying all the data, you will need to create a new Product record. First start the web site with **Ctrl-F5** to run without debugging. When the page loads, press the Insert button and you should get:
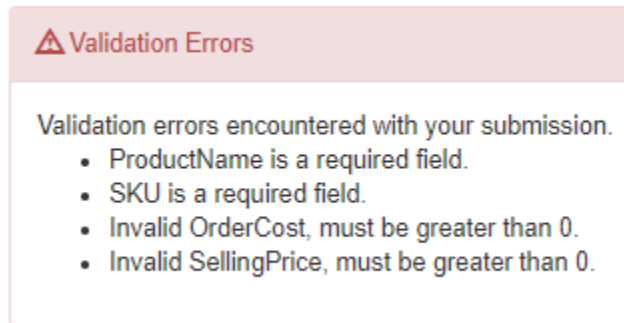
⚠ Validation Errors

Validation errors encountered with your submission.
- ProductName is a required field.
- SKU is a required field.
- Invalid OrderCost, must be greater than 0.
- Invalid SellingPrice, must be greater than 0.

Figure 18: Validation Errors - Insert Blank Record

This shows our Data Annotations on the Product class are working. Next try to add a sample record; a sample to add is shown below (yours can be different):

Insert Clear		New Product	NP-NewPr01	Housewares ▼		Household Supplies ▼	0.1	1.00	10	5		0

Figure 19: Sample Add Product

The new record is added. Navigate to the last page of the ListView to view the new record:

Delete Edit	78	New Product	NP-NewP01	Housewares ▼		Household Supplies ▼	0.10	1.00	10	5		0
Insert Clear				Alcohol ▼		Best Western Fruit ▼						

Figure 20: New Record Added

Try editing this new record. Were you successful?

Delete Edit	78	New Product	NP-NewP01	Housewares ▼		Household Supplies ▼	0.20	1.30	100	50		0

Figure 21: Edited New Record

Now try to Delete this new record. Were you successful?

Delete Edit	77	Kitchen Utensils	H-KUtns-01	Housewares ▼	Complete set of kitchen utensils	Household Supplies ▼	8.95
Insert Clear				Alcohol ▼		Best Western Fruit ▼	
						First ... 6 7 8 Last	

Figure 22: New Record Was Deleted

Add another new record. What happened?

Delete Edit	77	Kitchen Utensils	H-KUtns-01	Housewares ▾	Complete set of kitchen utensils	Household Supplies ▾	8.95	12.25	40	20	0
Delete Edit	79	Another Product	NP-NewPr02	Housewares ▾		Everything Kitchen ▾	0.30	1.75	25	20	0
Insert Clear				Alcohol ▾		Best Western Fruit ▾					

First ... 6 7 8 Last

Figure 23: Another New record Added

Notice that the `ProductID=78`, which was the record we deleted, is missing; the next available Primary Key value was used instead. This is why the Delete functionality is rarely used!

Exercise

Complete Exercise 6.2.1.