

5.2.0 – Looping Through Controls

Introduction

In the previous lesson, you were able to select a row from a `ListView` control and display the data on the `MessageUserControl1`. In this lesson, you will be shown how to loop through all the rows of a `ListView` display summary information. In this lesson you will be modifying both the **`ShoppingCart.aspx`** and the **`PurchaseOrder.aspx`** web forms.

ShoppingCart.aspx

ShoppingCart.aspx

What we want to do here is to create a summary of:

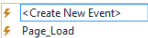
- Total number of products for sale by the category selected
- Total price of the products in the selected category

Add the following code below the `ListView`:

```
<br />
<asp:Button ID="ListViewSummary" runat="server" Text="Category Summary" CssClass="btn btn-primary" />
```

We need to add the **`OnClick`** parameter to this button:

```
<asp:Button ID="ListViewSummary" runat="server" Text="Category Summary" CssClass="btn btn-primary" OnClick="" />
v>
ObjectDataSource Control(s) -->
```



This will give us the following:

```
<asp:Button ID="ListViewSummary" runat="server" Text="Category Summary" CssClass="btn btn-primary" OnClick="ListViewSummary_Click"/>
```

Now, open the code behind file, **`ShoppingCart.aspx.cs`**, to see the new event method stub:

```
protected void ListViewSummary_Click(object sender, EventArgs e)
{
}
}
```

Modify this code to be:

Listing 1: ListViewSummary_Click Event Code

```
protected void ListViewSummary_Click(object sender, EventArgs e)
{
    int count = 0;
    decimal categoryPriceTotal = 0;
```

```

string msg;
//Loop through all the items in the ListView
foreach(ListViewItem item in ProductListLV.Items)
{
    count++;
    categoryPriceTotal += decimal.Parse((item.FindControl("PriceTextBox") as
TextBox).Text);
} //end foreach
msg = "Number of " + CategoryListDDL.SelectedItem.Text + " Products: " +
count.ToString() + ", Category Price Total: " + categoryPriceTotal.ToString();
MessageUserControl.ShowInfo("Category Summary", msg);
} //eom

```

Loop through all the
Items of the ListView.

Run the **ShoppingCart** web form, select a category, and your output should look like:

The screenshot shows a web form titled "Category Summary". Below the title, it displays "Number of Meat Products: 7, Category Price Total: 30.80". There is a dropdown menu for "Category:" with "Meat" selected, and a "Fetch" button. Below this is a table of products with columns: ID, Name, SKU, Description, Price, and a "Select" button. The table lists 8 items: 29 Beef Sausage, 30 Beef Sausage, 33 Chicken Burger, 34 Chicken Burger, 35 Chicken Wings, 31 Pork Sausage, 32 Pork Sausage, and another 32 Pork Sausage. At the bottom of the table is a pagination bar with "First", "1", and "Last" buttons. Below the table is a "Category Summary" button.

ID	Name	SKU	Description	Price	
29	Beef Sausage	M-BfSge-01	Large Beef Sausage	6.25	Select
30	Beef Sausage	M-BfSge-02	Small Beef Sausage	4.65	Select
33	Chicken Burger	M-CkBgr-01	Plain chicken burger pattie	3.25	Select
34	Chicken Burger	M-CkBgr-02	Spicy chicken burger pattie	3.30	Select
35	Chicken Wings	M-Wings-01	Cajun spiced chicken wings	5.25	Select
31	Pork Sausage	M-PkSge-01	Large Pork Sausage	4.55	Select
32	Pork Sausage	M-PkSge-02	Breakfast Sausage	3.55	Select

Figure 1: ListView Walk Through Output

PurchaseOrder.aspx

What we want to do here is to create a summary of:

- Total number of products for order by the supplier selected
- Total value of the products for the selected supplier

For a Product: $Value = Cost \times QuantityOnHand$

As with **ShoppingCart.aspx**, you need to add a button below the ListView:

We will need to add separate command buttons to the web form (outside of the ListView and GridView controls). Add the following code below the ListView control:

```

<br />
<asp:Button ID="SupplierSummary" runat="server" Text="Supplier Summary" CssClass="btn
btn-primary" />

```

```
protected void SupplierSummary_Click(object sender, EventArgs e)
{
}

```

Listing 2: ListView Walk Through Event Code

Loop through all the Items of the ListView.

The **Cost**, and **QOH** cells are read the same way we did previously

Figure 2: ListView Walk Through

Figure 3: Calculation Check with Excel

Uses for This Technique

Displaying the summary is a nice touch, but the process of looping through the controls is more useful when you want to collect all the information from a control, create a `List<T>`, and process the `List<T>` on this web form or in another file. A demonstration of this will be done in a later lesson.

Exercise

Complete Exercise 5.2.1.