

## 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:

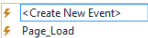
- Total number of products for sale by the category selected
- Total price of the products in the selected category

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-info" 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:

**Category Summary**

Number of Meat Products: 7, Category Price Total: 30.80

Category:

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-info" />

```

Add the **OnClick** parameter to this buttons, creating a new event. Code this event as shown below:

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

The **ListView** walk through event code is:

*Listing 2: ListView Walk Through Event Code*

```
protected void SupplierSummary_Click(object sender, EventArgs e)
{
    int count = 0;
    decimal inventoryCost = 0;
    string msg;
    foreach (ListViewDataItem item in ProductListLV.Items)
    {
        count++;
        inventoryCost += int.Parse((item.FindControl("QOHTextBox") as TextBox).Text) *
decimal.Parse((item.FindControl("CostTextBox") as TextBox).Text);
    }//end for
    msg = "Number of " + SupplierListDDL.SelectedItem.Text + " Products = " +
count.ToString() + ", Inventory Cost = " + inventoryCost.ToString();
    MessageUserControl.ShowInfo("Supplier Summary", msg);
} //eom
```

Loop through all the  
Items of the ListView.

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

Supplier Summary

Number of Organic Meats Products: 3, Inventory Cost: 418.00

Supplier: Organic Meats Fetch

ID	Name	SKU	Cost	ROL	QOH	Ordered	
33	Chicken Burger	M-CkBgr-01	2.60	40	50	0	Select
34	Chicken Burger	M-CkBgr-02	2.65	50	40	20	Select
35	Chicken Wings	M-Wings-01	4.55	75	40	100	Select

Supplier Summary

Figure 2: ListView Walk Through

	A	B	C
1	Cost	QOH	Cost
2	2.60	50	130.00
3	2.65	40	106.00
4	4.55	40	182.00
5	Inv Cost		418.00
6			

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.