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 `MessageUserControl`. 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

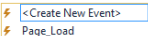What we want to do here is to create a summary of:

- 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) -->
```
```
⚡ <Create New Event>
⚡ Page_Load
```

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(ListViewDataItem 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:

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" />
```

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("ListView Summary", msg);
}//eom
```

> 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.