Hands-On Lab
Lab Manual

ASP.NET: Creating User Controls

Please do not remove this manual from the lab
The lab manual will be available from CommNet

Release Date: June 2003
Release Date: June 2003

Microsoft
Tech·Ed
2003

ถอนถอดอัลฟавท์

Release Date: June 2003

withdrawn

οδηγηθείσα

released

οδηγηθείσα

released
Contents

LAB 7: ASP.NET: CREATING USER CONTROLS .................................................................................................. 1
   Lab Objective ................................................................................................................................................... 1
   Exercise 1 – Creating and Using a Simple User Control ................................................................. 1
      Task 1 – Creating User Control........................................................................................................... 1
      Task 2 – Exposing Properties ............................................................................................................. 3
      Task 3 – Consuming User Control .................................................................................................... 4
   Exercise 2 – Creating and Using a menu Control .............................................................................. 6
      Task 1 – Creating User Control ........................................................................................................... 6
      Task 2 – Exposing Properties ............................................................................................................. 12
      Task 3 – Initializing the User Control ................................................................................................ 12
      Task 4 – Consuming User Control .................................................................................................... 13
   Exercise 3 – Creating and Using an Aggregate Control ............................................................. 15
      Task 1 – Creating Aggregate Control .............................................................................................. 15
      Task 2 – Exposing Properties ............................................................................................................. 16
      Task 3 – Create the Calendar page .................................................................................................. 16
      Task 4 – Consuming Aggregate Control .......................................................................................... 16
   Lab Summary ............................................................................................................................................. 16
Lab 7: ASP.NET: Creation User Controls

User controls are controls that you can define using the same programming techniques that you use to write Web Forms pages for your applications. You can create Web user controls when you need to create reusable pieces of UI that will be used throughout your application. A user control can contain HTML, server controls, and event-handling logic. Like a compiled control, it can expose properties that the host Web Forms page can set.

The steps to create user control are very similar to those for creating a Web Forms page. You design the UI visually by arranging ASP.NET server controls, HTML, and static text on the design surface, binding data and writing code to handle events raised by the controls.

Lab Setup:

The code for all these exercises is located at %SystemDrive%\Microsoft Hands-On-Lab\DEV-HOL07\ Solution.

Lab Objective

Estimated time to complete this lab: 60 minutes

The objective of this lab is to make you familiar with simple user controls. It teaches you how to create and use user controls in Web Forms Page. This lab will guide you to create three useful controls, a login page control, a menu control and a date selection control.

In this lab you will perform the following exercises:

- Creating and using a simple user control
- Creating and using an aggregate control

Exercise 1 – Creating and Using a Simple User Control

In this exercise, you will create a login page user control that allows the user to enter user id and password. The first step involves creating the user control. Then you will expose the properties of the user control and embed the control in a Web Forms page. You finally view it in a Web browser.

Task 1 – Creating User Control

- Using Microsoft Visual Studio .NET 2003 (VS03), create a new Web Project named Lab7-Ex1-CS
  - Select File | New | Project menu command.
  - In the New Project dialog box, click Visual C# Projects in Project Type list and ASP.NET web application in Templates list.
In **Location** field, enter "http://localhost/Lab7-Ex1-CS", and click **OK**.

- Open the Solution Explorer by selecting **View | Solution Explorer** menu command.
- In the solution explorer, right-click WebForm1.aspx and rename it to “container.aspx”.

- Add “login.ascx” **Web User Control** to the project:
  - Right-click the project (Lab7-Ex1-CS) in solution explorer and select **Add | Add Web User Control…** in context menu. **Add New Item** dialog box appears.
  - Modify the **Name** to “login.ascx” and click **Open**.

- To complete the user control, follow the steps:
  - To add a table to the user control, select **Table | Insert | Table** menu command.
  - **Insert Table** dialog box appears. Select 3 in **Rows** field and 2 in **Columns** field.
  - Click **OK** to close the **Insert Table** dialog box.
  - A table is added to the design view of user control.

  - To open **Toolbox**, click Toolbox button ( ) in toolbar or select **View | Toolbox** menu command.
  - Drag a **Label** from Toolbox and drop it in top left cell of the table.
  - Right-click the label and select Properties.
  - Set **Text** property of the label to “User ID” using **Properties** window.
  - Drag a **Textbox** from toolbox and drop it in top right cell of table.
  - Right-click the textbox and select Properties.
  - Set the (ID) property of textbox to “User”
  - Drag another **Label** from toolbox and drop it in center left cell of table.
  - Right-click the label and select Properties.
  - Set the **Text** property of label to “Password” using the **Properties** window.
  - Drag another **Textbox** from toolbox and drop it in right center cell of table.
  - Right-click the textbox and select Properties.
  - Set the following properties of the textbox using **Properties** window:
    - (ID) - “Pass”
    - **TextMode** - **Password** (select from drop down list)
  - Drag a **Button** from toolbox and drop it in bottom left cell of table.
  - Right-click the button and select Properties.
  - Set the **Text** property of button to “Submit” using **Properties** window.
Press **F7** to switch from design view to code-behind file (login.ascx.cs).

Define **UserId** property below the **Page_Load** event handler in code behind file of user control.

```csharp
private void Page_Load(object sender, System.EventArgs e)
{
    // Put user code to initialize the page here
}

public string UserId
{
    get
    {
        return User.Text;
    }
    set
    {
        User.Text = value;
    }
}
```

Similarly, define **Password** property below **UserId** property definition.

```csharp
public string Password
```
Task 3 – Consuming User Control

- Select View | Solution Explorer menu command.
- Double-click the container.aspx in solution explorer.
- Drag login.ascx from solution explorer and drop it in the design area.
- Select View | Toolbox menu command.
- Double-click Label on Web Forms tab of Toolbox to add it to the page. Drag the label control below the user control.
- Right-click the label control and select Properties.
- Clear the Text property of label control.

Your container.aspx file design view should look like the following image:
Figure 7.2 Adding User Control and label

- Add Code-Behind declaration for the user control.
  - Press F7 to switch from the design view to the code-behind file.
  - In the declaration area, add a line to declare the user control Login1 as menu.
    
    ```
    public class WebForm1 : System.Web.UI.Page
    {
    protected System.Web.UI.WebControls.Label Label1;
    protected Login Login1;
    ...
    ...
    }
    ```

- Set the Text property of label control in Page_Load event handler:

  ```
  private void Page_Load(object sender, System.EventArgs e)
  {
  if (Page.IsPostBack)
  {
  Label1.Text = "The UserId is " + Login1.UserId + "<br>");
  Label1.Text += "The Password is " + Login1.Password + "<br>");
  }
  ```
• Press **F5** to run the container.aspx and test the user control. Enter the values in **User Id** and **Password** fields and click **Submit**.

• Close the Web browser.

• To close Visual Studio .NET 2003, select **File | Exit** menu command.

**Exercise 2 – Creating and Using a menu Control**

In this exercise, you will create a menu selection control that passes the selected item back to the calling page, in the URL query string. The control displays information categories in a menu control. It gets the categories and populates the menu dynamically from a data source. The first step involves creating the user control. Then you will expose the properties of the user control and embed the control in a Web Forms page. You finally view it in a Web browser.

**Task 1 – Creating User Control**

• Using Microsoft Visual Studio .NET 2003 (VS03), create a new Web Project named **Lab7-Ex2-CS**
  o Click **Start**. Select **Programs | Microsoft Visual Studio .NET 2003 | Microsoft Visual Studio .NET 2003**
  o Select **File | New | Project** menu command.
  o In the **New Project** dialog box, click **Visual C# Projects** in Project Type list and **ASP.NET web application** in Templates list.
  o In **Location** field, enter "http://localhost/Lab7-Ex2-CS", and click **OK**.

• Open the Solution Explorer by selecting **View | Solution Explorer** menu command.

• In the solution explorer, right-click WebForm1.aspx and rename it to “menuContainer.aspx”.

• Add “menu.ascx” **Web User Control** to the project:
  o Right-click the project (Lab7-Ex2-CS) in solution explorer and select **Add | Add Web User Control...** in context menu. **Add New Item** dialog box appears.
  o Modify the **Name** to “menu.ascx” and click **Open**.

• Add a **DataList** control to “menu.ascx” web server control.
  o To open **Toolbox**, click Toolbox button ( ![Toolbox Icon](image)) in toolbar or select **View | Toolbox** menu command.
  o Click the **Web Forms** tab of the toolbox.
  o Double-click **DataList** in the toolbox. Control is added with the default name **DataList1**.

• Right-click dataist control and select **Property Builder**.
In the **General** tab of **Property Builder** dialog box, set the **Direction** property under Repeat Layout to **Horizontal**.
Switch to the **Format** tab and expand **Items | Selected Items**.

Check the **Bold** checkbox in **Appearance** groupbox.

---

**Figure 7.4 Setting Direction property to Horizontal**
To close the DataList1 Properties dialog box, click OK.

Right-click datalist control and select Edit Template | Separator Template option. Observe the change in appearance of datalist control.

To separate the menu items, first select the DataList1 and then click below the Separator Template label and type the “|” character as shown in following image:

![Separator Template](image)

Figure 7.6 Modifying Separator Template

To add a Hyperlink control and set the binding expression, follow the steps:

- Right-click the datalist control and select Edit Template | Item Templates option.
- Select the datalist control.
Position the pointer just below ItemTemplate.

Select View | Toolbox menu command.

Drag the HyperLink control from Toolbox and drop it just below ItemTemplate OR click below ItemTemplate and double-click the HyperLink control in Toolbox.

Right-click hyperlink control and select Properties.

Select (DataBindings) in the Properties window.

Click the ellipsis button in the value field.

HyperLink1 DataBindings dialog box appears. In the Bindable Properties treeview, select Text property if not selected by default.

In the Binding for Text group, under Simple binding, expand the Container node and select DataItem. This will bind the Text property of the each hyperlink to the data item of the Container, which in this case is the datalist control. The datalist control will receive its data items from an array, which will be created in the host Web Forms page.

In the Bindable Properties treeview, select the NavigateUrl property. Note that when you click the NavigateUrl property, the icon next to Text property changes to show that it is data bound.

In the Binding for NavigateUrl group, under Simple Binding, select ItemIndex inside Container. This is the index number of the item in the array. To append this number to the URL, you must format it as a parameter so that the URL retrieves the data item at that index number. Name the parameter "category". To do this, enter this following string in the Format box.

?category={0}

Now you must complete the URL so that it points to the host Web Forms page.

Click Custom binding expression radio button and add the following code to the beginning of your already created databinding expression:

Request.Path +
Your complete custom databinding expression should be:

```c#
Request.Path + DataBinder.Eval(Container, "ItemIndex", "?category={0}")
```

![HyperLink1 DataBindings](image)

**Figure 7.8 HyperLink1 DataBindings**

- To close the DataBinding dialog box, click OK.
- To apply styles to the control, follow these steps:
  - Switch to HTML view by clicking HTML tab down the design area.

![Switching to HTML view](image)

**Figure 7.9 Switching to HTML view**

- Add the following highlighted HTML content after the first line of document:

```html
<%@ Control Language="c#" AutoEventWireup="false"......................
```
Task 2 – Exposing Properties

- Press F7 to switch from design view to code-behind file (menu.ascx.cs).
- Add a public class-level declaration for a string array, and name it **values**. This array will hold the list of data items, which will appear in menu.

```csharp
public string[] values;
```

- Similarly, declare **selection** as Integer and initialize to **-1**

```csharp
public int selection = -1;
```

This variable will hold the index value for the menu item that the user selects. Since -1 is not an index value in the array, the variable will not be assigned an index value on initialization.

The top section of your code should look something like this:

```csharp
Public abstract class menu : System.Web.UI.UserControl
{
    protected System.Web.UI.WebControls.DataList DataList1;
    public string[] values;
    public int selection = -1;
    ...
    ...
}
```

Task 3 – Initializing the User Control

As the control is ready now, you need to add the code to initialize the connection between the control and the data. Use the page load event to do the initialization.

- In the **Page_Load** procedure, set values to the **DataSource** property for the datalist control.

```csharp
DataList1.DataSource = values;
```

- Set the selection to be the **SelectedIndex** property for the datalist control.

```csharp
DataList1.SelectedIndex = selection;
```

- Call the **DataBind()** method of the datalist control.

```csharp
DataList1.DataBind();
```

The complete code for **Page_Load** procedure is:

```csharp
private void Page_Load(object sender, System.EventArgs e)
```
• Save the control code-behind page by selecting **File | Save menu.ascx.cs** menu command.

**Task 4 – Consuming User Control**

• Select **View | Solution Explorer** menu command.
• Double-click the menuContainer.aspx in solution explorer.
• Drag **menu.ascx** from solution explorer and drop it in the design area.
• Select **View | Toolbox** menu command.
• Double-click **Label** on Web Forms tab of **Toolbox** to add it to the page. Drag the label control below the user control.

Your menuContainer.aspx file design view should look like the following image:

![Figure 7.10 Adding User Control and label](image)

• Right-click the label control and select Properties.
• Set the following properties of label control using **Properties** window:
  - **Text** - "" (clear the text)
  - **Width** - 500
• Save your page by selecting File | Save menuContainer.aspx menu command or by clicking the save button ( ) in toolbar.

• Add Code-Behind declaration for the user control.
  o Press F7 to switch from the design view to the code-behind file.
  o In the declaration area, add a line to declare the user control Menu1 as menu.

```csharp
public class WebForm1 : System.Web.UI.Page
{
    protected System.Web.UI.WebControls.Label Label1;
    protected menu Menu1;
    ...
    ..
```

• Now you need to set the properties of user control and set the Label control text. To do that, follow the steps:
  o Declare an array in Page_Load procedure and assign it to values property of User control.

```csharp
String[] values = new String[] {"News", "Events", "Reference"};
Menu1.values = values;
```

• Get the category value from query string and assign it to selection property of User control. Make sure that you convert this to integer.

```csharp
String selectionId = Request.Params["category"];
if(selectionId != "")
{
    int SelectedIndex = Convert.ToInt16(selectionId);
    Menu1.selection = SelectedIndex;
}
```

• Now set the Text property of label.

```csharp
int SelectedIndex = Convert.ToInt16(selectionId);
Menu1.selection = SelectedIndex;
Label1.Text = "Current selection is: " + values[selectedIndex];
```

The complete code for Page_Load procedure is:

```csharp
String[] values = new String[] {"News", "Events", "Reference"};
Menu1.values = values;
String selectionId = Request.Params["category"];  
if(selectionId != "")
```
```csharp
int SelectedIndex = Convert.ToInt16(selectionId);
Menu1.selection = SelectedIndex;
Label1.Text = "Current selection is: " + values[SelectedIndex];
}
```

- Save menuContainer.aspx.cs file by selecting File | Save menuContainer.aspx.cs menu command or by clicking the save button () in toolbar.
- Press F5 to run the menuContainer.aspx and test the user control. Click each of the menu selections to test their functionality.
- Close the Web browser.
- To close Visual Studio .NET 2003, select File | Exit menu command.

**Exercise 3 – Creating and Using an Aggregate Control**

In this exercise, you will create an aggregate control and will see how this control can be used in a Web Forms page. The control allows the user to select two dates (start date and end date) using a Calendar control. First you will create the user control, and then you will expose the properties of user control. After that you will create a Web Forms page having a calendar control. User Control will use this page for date selection. Finally you will use the user control in a separate Web Forms page.

**Task 1 – Creating Aggregate Control**

- Using Microsoft Visual Studio .NET 2003 (VS03), create a new Web Project named Lab7-Ex3-CS.
  - Select File | New | Project menu command.
  - In the New Project dialog box, click Visual C# Projects in Project Type list and ASP.NET web application in Templates list.
  - In Location field, enter "http://localhost/Lab7-Ex3-CS", and click OK.
- Select View | Solution Explorer menu command.
- In solution explorer, right-click WebForm1.aspx and rename it to "datesContainer.aspx".
- Add a Web User Control to the project with “dates.ascx” name.
  - Right-click the project (Lab7-Ex3-CS) in solution explorer and select Add | Add Web User Control... option in context menu. Add New Item dialog box appears.
  - Modify the Name to “dates.ascx” and click Open.
- To complete the aggregate control, follow the steps:
  - Select View | Toolbox menu command.
Double-click **Label** in **Toolbox**. A label is added in design area.

- Right-click the label and select Properties.
- Set the following properties for label using **Properties** window:
  
  **Text** - “From Date:”

Double-click **TextBox** in toolbox. A textbox is added to the page.

- Right-click the textbox and select Properties.
- Set the following properties for textbox in **Properties** window:
  
  **ID** - “txtFromDate”

Click **HTML** tab in **Toolbox**.

Double-click **Button** in **Toolbox**.

- Right-click the button and select Properties.
- Set the following properties for the button:
  
  **Value** - “… ”
  
  **name** - “btnFrom”

Click the **Web Forms** tab of **Toolbox**.

- Double-click **Label** in **Toolbox**. A label is added in design area.
- Right-click the label and select Properties.
- Set the following properties for label using **Properties** window:
  
  **Text** - “To Date:”

Double-click **TextBox** in **Toolbox**. A textbox is added to the page.

- Right-click the textbox and select Properties.
- Set the following properties for textbox.
  
  **ID** - “txtToDate”

Click **HTML** tab of the **Toolbox**.

- Double-click **Button** in **Toolbox**.
- Right-click the button and select Properties.
- Set the following properties for the button:
  
  **Value** - “… ”
  
  **name** - “btnTo”

- To apply styles to the control, follow the steps:
  
  - Switch to **HTML** view by clicking the HTML tab down the design area.
Figure 7.11 Switching to HTML view

- Add following highlighted HTML content after the first line of document:

  ```html
  <table style="BORDER-RIGHT: black 1px solid; BORDER-TOP: black 1px solid; BORDER-LEFT: black 1px solid; BORDER-BOTTOM: black 1px solid" border="2" cellspacing="2" cellpadding="2" width="100%" borderColor="#99cccc" bgColor="#ccccff"><tr><td>
</td></tr></table>

- Add following HTML content at the end of the document:

  ```html
  </td></tr></table>
  ```

- In the HTML content, look for the `btnFrom` button and call `btnFromDate_Click()` method for `onClick` event.

  ```html
  <INPUT type="button" onclick="btnFromDate_Click()" value="..." name="btnFrom"
  ```

- Similarly, look for the `btnTo` button and call `btnToDate_Click()` method for `onClick` event.

  ```html
  <INPUT type="button" onclick="btnToDate_Click()" value="..." name="btnTo"
  ```

- Write the code for `btnFromDate_Click()` and `btnToDate_Click()` procedures after first line as shown below:

  ```html
  <script language="JavaScript">
  function btnFromDate_Click(){
    window.open("selectDate.aspx?tag=From","","width=300,height=250,scrollbars=yes")
  }
  function btnToDate_Click(){
    window.open("selectDate.aspx?tag=To","","width=300,height=250,scrollbars=yes")
  }
  </script>
  ```

- Switch to design view by clicking **Design** tab.

  ![Design Tab](image)

Figure 7.12 Switching to Design view
The design view of the dates.ascx page should look like this:

![Figure 7.13 Dates Control view](image)

**Task 2 – Exposing Properties**

- Press F7 to switch from design view to code-behind file (dates.ascx.cs).
- Add a public class-level declaration for string **From Date** and string **To Date**.

The top section of your code should look something like this:

```csharp
Public abstract class dates : System.Web.UI.UserControl
{
    protected System.Web.UI.WebControls.TextBox txtFromDate;
    protected System.Web.UI.WebControls.TextBox txtToDate;
    public string FromDate;
    public string ToDate;
    ...  
    ...
}
```

- In **Page_Load** procedure, set values for both the properties.

```csharp
private void Page_Load(object sender, System.EventArgs e)
{
    FromDate = txtFromDate.Text;
    ToDate = txtToDate.Text;
}
```

**Task 3 – Create the Calendar page**

- Select **View | Solution Explorer** menu command.
- Right-click the project (Lab7-Ex3-CS) in the Solution Explorer and select **Add | Add Web Form** .... **Add New Item** dialog box appears.
Modify Name to “selectDate.aspx” and click Open.

Select View | Toolbox menu command.

Double-click the Calendar control to add it in the form. A calendar control with default name Calendar1 is added.

Click the HTML tab of the Toolbox.

Double-click Button in Toolbox. A button is added to the form.

Drag the button below the calendar control.

Right-click the button and select Properties.

Set the following properties of the button control using Properties window.

value - “OK”
name - “btnOk”

The design view of selectDate.aspx page should look like the following image:

Figure 7.14 Design of selectDate Web Forms page

Switch to HTML view by clicking the HTML tab down the design area.

Add two server side hidden variables (selection and tag) inside the form as shown below:

```
<form id="Form1" method="post" runat="server">
<input id="selection" type="hidden" runat="server" NAME="selection">
<input id="tag" type="hidden" runat="server" NAME="tag">
```

“selection” hidden variable holds the value of selected date and is used to send the value back to opening window when OK button is pressed. “tag” hidden variable is used to carry the value of
query string parameter “tag”, which differentiates the From button on user control from the To button on user control.

- Look for the **btnOk** button in HTML content of selectDate.aspx page and call **showDate()** method for **onClick** event.

```html
value="Ok" onclick=showDate()>
```

- Write the **showDate()** method to write selected date of calendar in the textbox of opening window:

```javascript
function showDate(){
    var tagVal = Form1.tag.value;
    if(tagVal.indexOf("From")!=-1){
        window.opener.Form1.Dates1_txtFromDate.value = Form1.selection.value;
    }else{
        window.opener.Form1.Dates1_txtToDate.value = Form1.selection.value;
    }
    window.close();
}
</script>
```

- Switch to design view by clicking **Design** tab.
- Right-click the **Calendar** control and select Properties. Click events icon in the **Properties** window.
Double-click the **SelectionChanged** event. This will switch you to the code behind page of the `selectDate.aspx` page and will create an event procedure for SelectionChanged event. Assign values to `selection` and `tag` hidden variables in this procedure. “**selection**” is assigned to the current selected date in the calendar control and “**tag**” is assigned to the value of tag which is a part of query string.

```csharp
private void Calendar1_SelectionChanged(object sender, System.EventArgs e)
{
    selection.Value = Calendar1.SelectedDate.ToShortDateString();
    tag.Value = Request.QueryString["tag"];  
}
```

Similarly assign the values in **Page_Load** procedure also so that the assignment happens when the page is loaded first time.

```csharp
private void Page_Load(object sender, System.EventArgs e)
{
    selection.Value = Calendar1.SelectedDate.ToShortDateString();
    tag.Value = Request.QueryString["tag"];  
}
```
• Save the selecteDate.aspx.cs file by selecting File | Save selecteDate.aspx.cs menu command or by clicking save button (\f) in the toolbar.

Task 4 – Consuming Aggregate Control

• Select View | Solution Explorer menu command.
• Double-click datesContainer.aspx in solution explorer.
• Drag dates.ascx from Solution Explorer and drop it in design area of datesContainer.aspx page.
• Select View | Toolbox menu command.
• Click the Web Forms tab of the Toolbox.
• Double-click Label in Toolbox. A label is added to the page with default name Label1.
• Right-click the Label control and select Properties.
• Clear Text property.
• To add a button to the page, double-click Button in the Toolbox.
• Right-click the button and select Properties.
• Set Text property of the button to “Submit” in Properties window.
• Align the newly added label and button below the Dates1 user control as shown in following image:

![Image](image.png)

Figure 7.16 Adding the Aggregate Control

• Switch to HTML view by clicking the HTML tab down the design area.
• Look for HTML code for the button and call btnSubmit_Click procedure for OnClick event. We will set the label value when the button is pressed.
<asp:Button id="Button1" OnClick="btnSubmit_OnClick"

• Write definition for btnSubmit_OnClick procedure.

<%@ Register TagPrefix="uc1" TagName="dates" Src="dates.ascx" %>
<script language=C# runat=server>
void btnSubmit_OnClick(object sender, EventArgs e){
    Label1.Text = "From Date: "  + Dates1.FromDate + "<br>";
    Label1.Text += "To Date: "  + Dates1.ToDate;
}
</script>

• Save your file by selecting File | Save datesContainer.aspx menu command or by clicking save button (保存) in the toolbar.

• To run the "datesContainer.aspx" page, press F5 and test the user control.
  o In browser, click the button next to first textbox (txtFromDate). It will open the "selectDate.aspx" page in a separate window.
  o Select a date in calendar control and click OK. It will close the current window and selected date will appear in txtFromDate textbox.
  o Click the button next to second textbox (txtToDate). It will open the "selectDate.aspx" page in separate window.
  o Select a date in calendar control and click OK. It will close the current window and selected date will appear in txtToDate textbox.
  o Click Submit button. Selected dates will be displayed in Label1.

• Close the Web browser.

• To close Visual Studio .NET 2003, select File | Exit menu command.

Lab Summary

User controls allow developers to create customized controls using the same programming techniques that are used in creating Web Forms pages. Such controls have ascx extension.

In this lab you have performed the following exercises:

• Creating and using a simple user control
• Creating and using an aggregate control