Code Repository Exercises

Web Controls

Purpose: The purpose of this exercise is to familiarize you with ASP.NET Web Server Controls, including their advantages over HTML server controls. You will see that Web Server controls can be manipulated in both the content and code-behind files. You will learn the implications of setting a control's A*utoPostBack* property. You will also utilize Web Server Control events and learn about the page lifecycle.

Directory Name: WebControls

Required Pages/Components:

1. Download the accompanying zip file, extract it, and place all of the files (including the *ControlDemo* subdirectory) in the *WebControls* directory created for this exercise.
2. Open the *ControlDemo.aspx* page found inside the *ControlDemo* subdirectory. Run this page and view the examples of the various basic Web controls. Go back and study the code to familiarize yourself with how the controls work.
3. Run the application and test out the *GreetingCardMaker.aspx* page. Make sure you understand how it works before moving on.
4. Copy the *GreetingCardMaker.aspx* file and paste it as a new file in the *WebControls* directory. Rename this copied file to: *NewGreetingCardMaker.aspx* (this will rename the code-behind file as well). Make the following changes to *NewGreetingCardMaker.aspx* and *NewGreetingCardMaker.aspx.cs*:
    a. Remove the code that populates the *lstBackColor* DropDownList from the code-behind file. Populate this DropDownlist with the same values (White, Red, Green, Blue, Yellow), but do it in the content file instead. (This is simply to show you that controls can be manipulated in the content file or code-behind file).
    b. Instead of allowing the user to simply select the default image, give the user the capability of selecting one or more images to display on the card. Replace the *chkPicture* CheckBox with a list control that will allow the user to select one or more items (i.e. a CheckBoxList or a ListBox). Each item in the list control should display a descriptive name for the picture (e.g. 'Graduation') and store the image file name as its value (e.g. 'graduation.jpg'). When the user clicks 'update', each image that the user has selected should display on the greeting card. (Hint: you will need to dynamically create a new Image object for each item that the user selects and add the Image objects to the Controls collection of the *pnlCard* panel. To do this, utilize the techniques you learned in the *C#* exercise.)
    c. Give the user the capability of entering the name of the person sending the card. The user should be able to enter the name in a TextBox. If **(and only if)** something is

entered in the textbox, the following text should appear at the bottom of the card: "From: *Name*", where *Name* is the text entered into the TextBox.

d. Update the page to refresh the card immediately any time a user changes one of the card settings (meaning that the user doesn't have to click 'Update'). To do this, consolidate all the code that updates the card into a single private method. Set the *AutoPostBack* property to true for all of the controls. Create event hander methods for each of the controls' default events, and call the private method in each of these methods. (This is demonstrated in the "Generating the Cards Automatically" section on pages 199-200 of the MacDonald Text).

e. Make any formatting changes to the Card to enhance its aesthetic appeal. Feel free to add additional functionality if you would like.

Header on Default.aspx: Web Controls

Pages linked from Default.aspx: NewGreetingCardMaker.aspx