

to render the page on the computer screen. For example, the instruction `<h1>Hello</h1>` uses the opening `<h1>` tag and its closing `</h1>` tag to display the word "Hello" as a heading on the Web page. Many ASP pages also contain ASP tags that specify the controls to include on the Web page. In addition to the HTML and ASP tags, dynamic ASP pages contain code that tells the objects on the Web page how to respond to the user's actions. In this chapter, you will write the appropriate code using the Visual Basic programming language.

When a client computer's browser sends a request for an ASP page, the Web server locates the page and then sends the appropriate HTML instructions to the client. The client's browser uses the instructions to render the Web page on the computer screen. If the Web page is a dynamic one, like the Dollars to Pesos page shown in Figure 27-3, the user can interact with the page by entering data. In most cases, the user then clicks a button on the Web page to submit the data to the server for processing. When the server receives the data, it executes the Visual Basic code associated with the Web page. It then sends back the appropriate HTML, which now includes the result of processing the code and data, to the client for rendering in the browser window. Using the Dollars to Pesos Web page as an example, the user first enters the number of dollars and then clicks the Submit button, which submits the user's entry to the Web server. The server executes the Visual Basic code to convert the dollars to pesos and then sends back the HTML, which now includes the number of pesos. Notice that the Web page's HTML is interpreted and executed by the client computer, whereas the program code is executed by the Web server. Figure 27-4 illustrates the relationship between the client computer and the Web server.

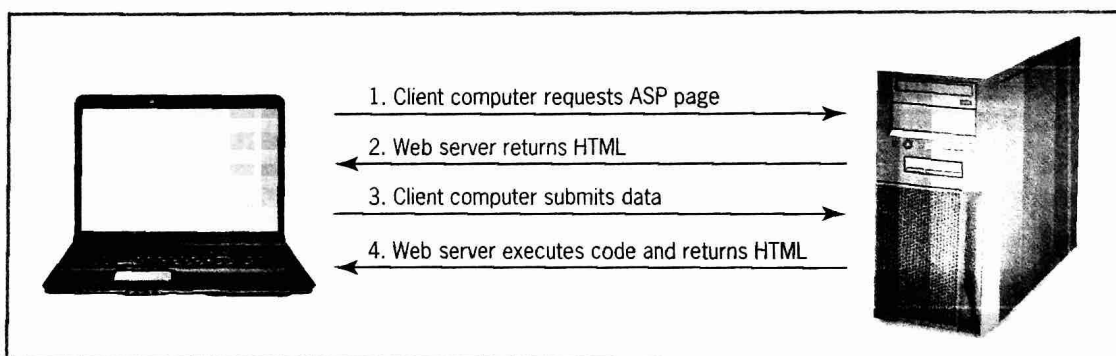


Figure 27-4 Illustration of the relationship between a client computer and a Web server
© Cengage Learning 2013

Mini-Quiz 27-1

See Appendix B for the answers.

1. ASP is an acronym for _____.
2. HTML is an acronym for _____.
3. The HTML instructions in a Web page are processed by the _____.
 - a. client computer
 - b. Web server

Creating a Web Application

You create a Web application in Visual Basic using Visual Studio 2012 for Web, which is available either as a stand-alone product (called Visual Studio Express 2012 for Web) or as part of Visual Studio 2012. You can download a free copy of Visual Studio Express 2012 for Web

from Microsoft's Web site. At the time of this writing, the address is <http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-for-web>. The following steps show you how to configure the Express edition. You should perform these steps only if you are using Visual Studio Express 2012 for Web.

To configure Visual Studio Express 2012 for Web:

615

1. **Windows 8:** If necessary, tap the **Windows logo** key to switch to the Windows 8 tile-based mode and then click the **VS Express for Web** tile.

Windows 7: Click the **Start** button on the Windows 7 taskbar and then point to **All Programs**. Click **Microsoft Visual Studio Express 2012** on the All Programs menu and then click **Visual Studio Express 2012 for Web**.

2. Click **TOOLS** on the menu bar and then click **Options** to open the Options dialog box. Click the **Projects and Solutions** node. Use the information shown in Figure 27-5 to select and deselect the appropriate check boxes.

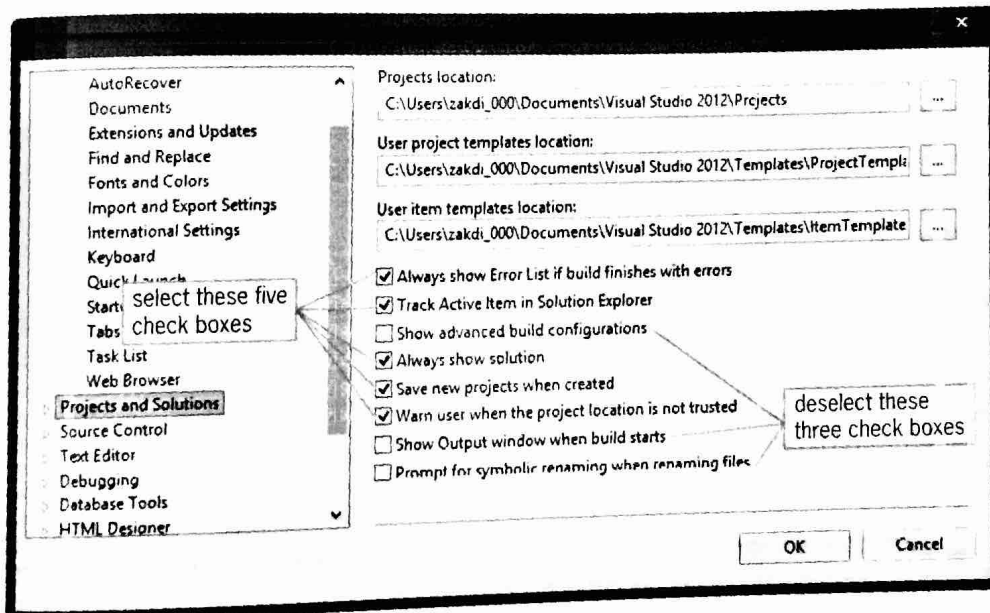


Figure 27-5 Options dialog box

3. Click the **OK** button to close the Options dialog box.
4. Click **TOOLS** on the menu bar and then point to **Settings**. If necessary, click **Expert Settings** to select it.

In the next set of steps, you begin creating the Brown Bear Toy Store Web application.

To begin creating the Web application:

1. If necessary, start Visual Studio 2012 or Visual Studio Express 2012 for Web.
2. If necessary, permanently display the Solution Explorer and Properties windows, and auto-hide the Toolbox window.
3. Click **FILE** on the menu bar and then click **New Web Site** to open the New Web Site dialog box. If necessary, click **Visual Basic** in the Installed Templates list, and then click **ASP.NET Empty Web Site** in the middle column of the dialog box.

4. If necessary, change the entry in the Web location box to **File System**. The File System selection allows you to store your Web application in any folder on either your computer or a network drive.
5. In this chapter, you will be instructed to store your Web applications in the ClearlyVB2012\Chap27 folder on the E drive; however, you should use the drive letter that contains your data files. In the box that appears next to the Web location box, replace the existing text with **E:\ClearlyVB2012\Chap27\BrownBear**. Figure 27-6 shows the completed New Web Site dialog box if you are using Visual Studio 2012. Your New Web Site dialog box will look slightly different if you are using Visual Studio Express 2012 for Web.

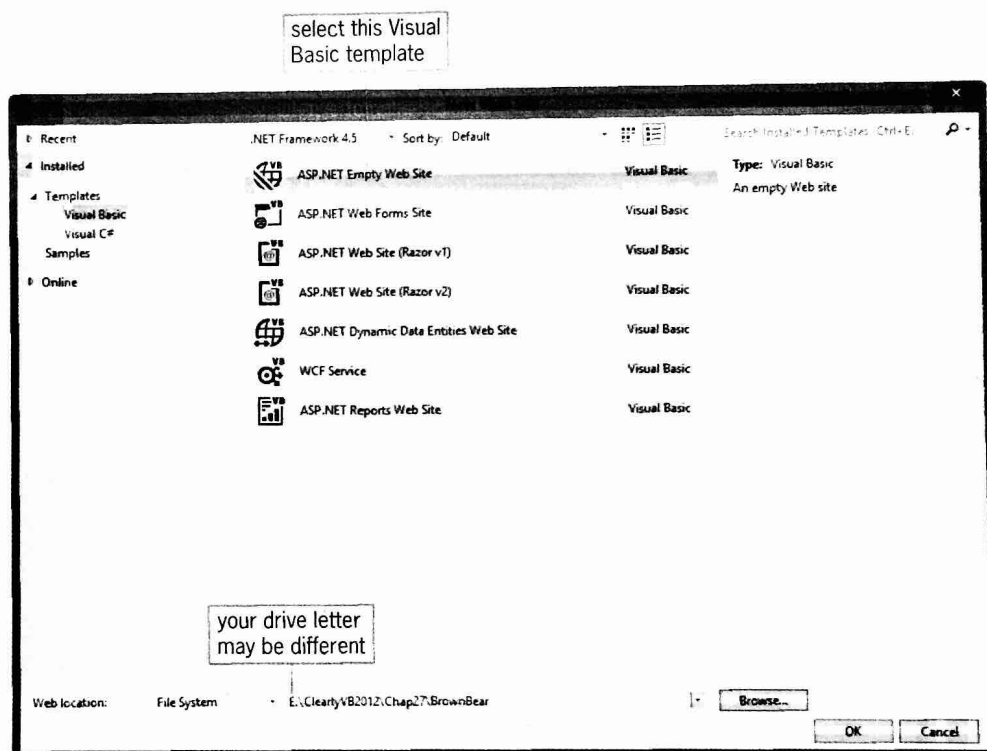


Figure 27-6 New Web Site dialog box

6. Click the **OK** button to close the dialog box. The computer creates an empty Web application named BrownBear.

Adding the Default.aspx Web Page to the Application

After creating an empty Web application, you need to add a Web page to it. The first Web page added to an application is usually named Default.aspx.

To add the Default.aspx Web page to the application:

1. Click **WEBSITE** on the menu bar and then click **Add New Item** to open the Add New Item dialog box. (If WEBSITE does not appear on the menu bar, click the Web application's name in the Solution Explorer window.)
2. If necessary, click **Visual Basic** in the Installed list and then (if necessary) click **Web Form** in the middle column of the dialog box. Verify that the Place code in separate file check box is selected, and that the Select master page check box is not selected. As indicated in Figure 27-7, the Web page will be named Default.aspx.

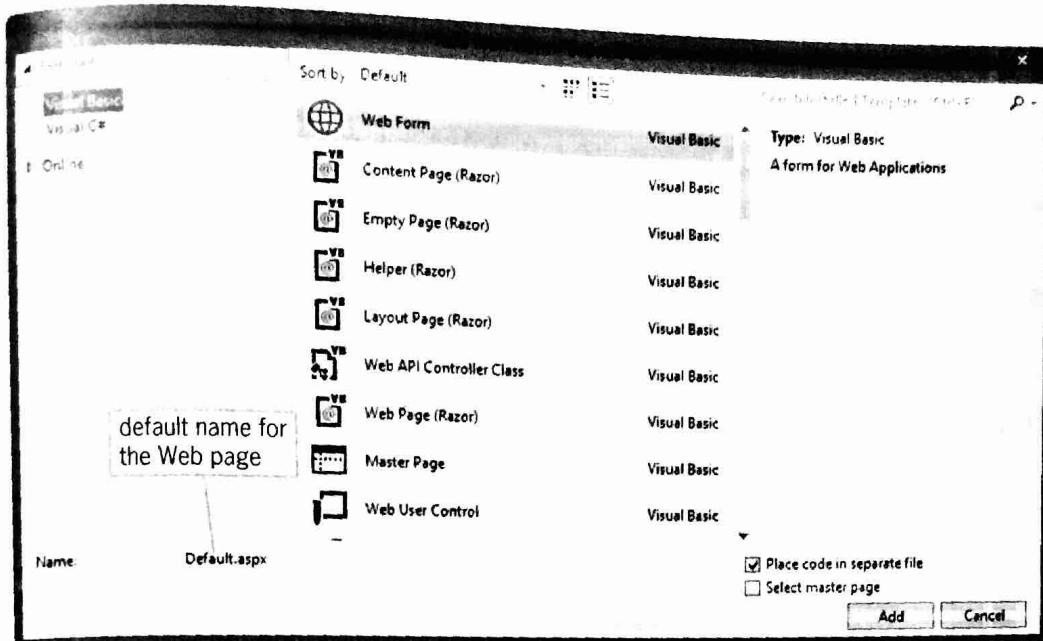


Figure 27-7 Add New Item dialog box

- Click the **Add** button to display the Default.aspx page in the Document window. If necessary, click the **Design** tab that appears at the bottom of the IDE. When the Design tab is selected, the Web page appears in Design view in the Document window, as shown in Figure 27-8. You can use Design view to add text and controls to the Web page. If the Formatting toolbar does not appear on your screen, click **VIEW** on the menu bar, point to **Toolbars**, and then click **Formatting**. If the div tag does not appear in the Document window, click either the <div> button at the bottom of the IDE or the **rectangle** below the body tag.

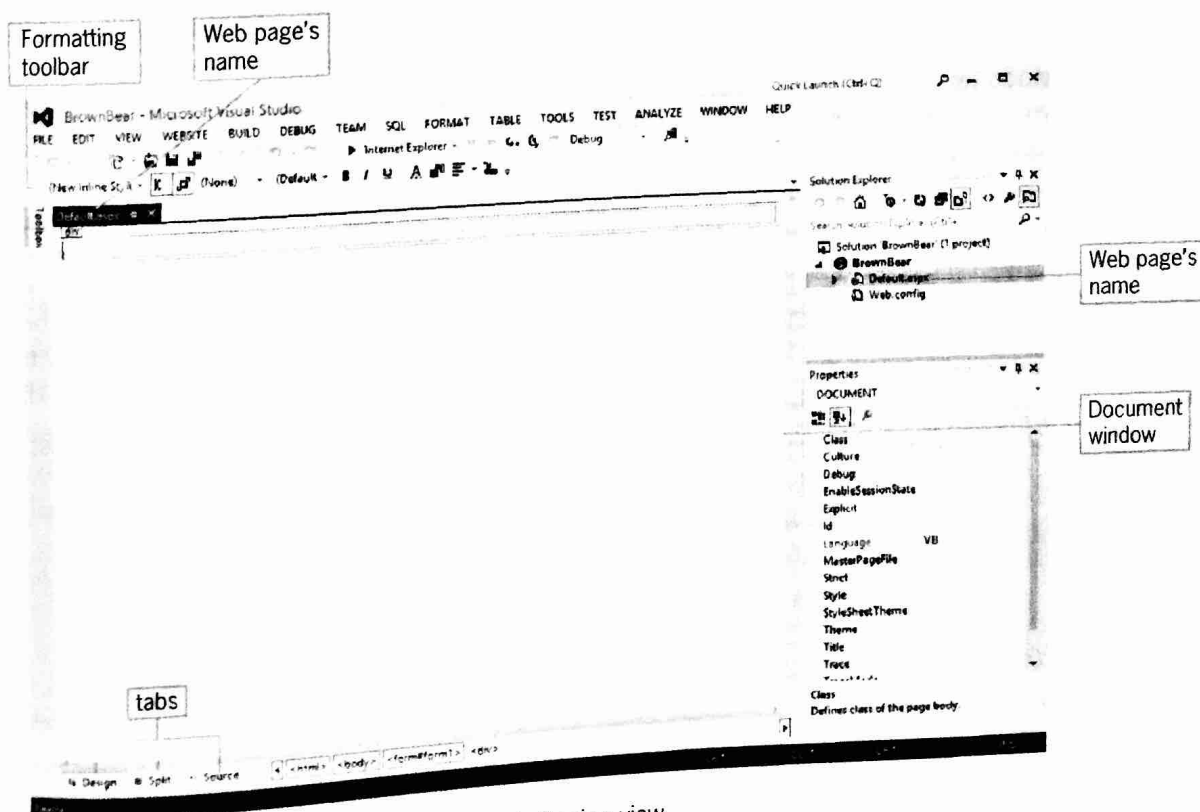


Figure 27-8 Default.aspx Web page shown in Design view

4. Click the **Source** tab to display the Web page in Source view. This view shows the HTML and ASP tags that tell a browser how to render the Web page. The tags are automatically generated for you as you are creating the Web page in Design view. Currently, the Web page contains only HTML tags.
5. Click the **Split** tab to split the Document window into two parts. The upper half displays the Web page in Source view, and the lower half displays it in Design view.
6. Click the **Design** tab to return to Design view, and then auto-hide the Solution Explorer window.

What Is Your Title?

You can use the Properties window to include a title on a Web page. The properties appear in the Properties window when you select DOCUMENT in the window's Object box.

To include a title on the Web page:

1. Click the **down arrow** button in the Properties window's Object box and then click **DOCUMENT** in the list. (If DOCUMENT does not appear in the Object box, click the Design tab.) The DOCUMENT object represents the Web page.
2. If necessary, click the **Alphabetical** button in the Properties window to display the properties in alphabetical order. Click **Title** in the Properties list. Type **Brown Bear Toy Store** in the Settings box and then press **Enter**.
3. Auto-hide the Properties window. Save the application by clicking either the **Save All** button on the Standard toolbar or the **Save All** option on the FILE menu.

Adding Static Text to a Web Page

All Web pages contain some text that the user is not allowed to edit, such as a company name or the caption that identifies a text box. Text that cannot be changed by the user is referred to as **static text**. You can add static text to a Web page by simply typing the text on the page itself; or, you can use a label control that you dragged to the Web page from the Toolbox window. In the following set of steps, you will type the static text on the Web page.

To add static text to the Web page:

1. If necessary, click **inside the rectangle** that appears below the div tag at the top of the Document window. The div tag defines a division in a Web page. (If the div tag does not appear in the Document window, click the <div> button at the bottom of the IDE.)
2. Enter the following four lines of text. Press **Enter** twice after typing the last line.
Brown Bear Toy Store
750 Main Street
Atlanta, GA 30301
(111) 555-5555
3. Save the application.

You can use either the FORMAT menu or the Formatting toolbar to format the static text on a Web page. Figure 27-9 indicates some of the tools available on the Formatting toolbar.

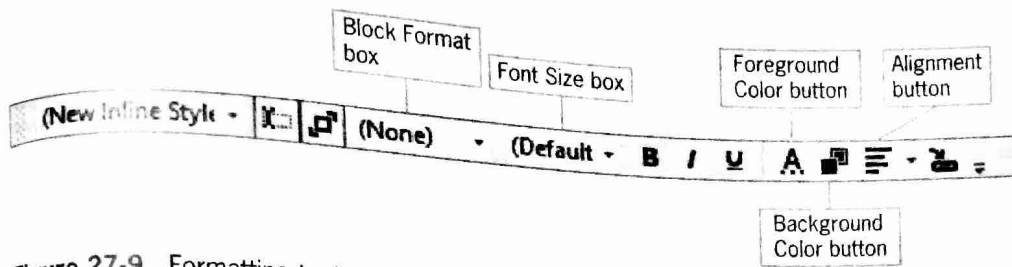


Figure 27-9 Formatting toolbar

To use the Formatting toolbar to format the static text:

1. Select (highlight) the Brown Bear Toy Store text on the Web page. Click the **down arrow** in the Block Format box on the Formatting toolbar. (If the Formatting toolbar does not appear on your screen, click VIEW on the menu bar, point to Toolbars, and then click Formatting.) See Figure 27-10.

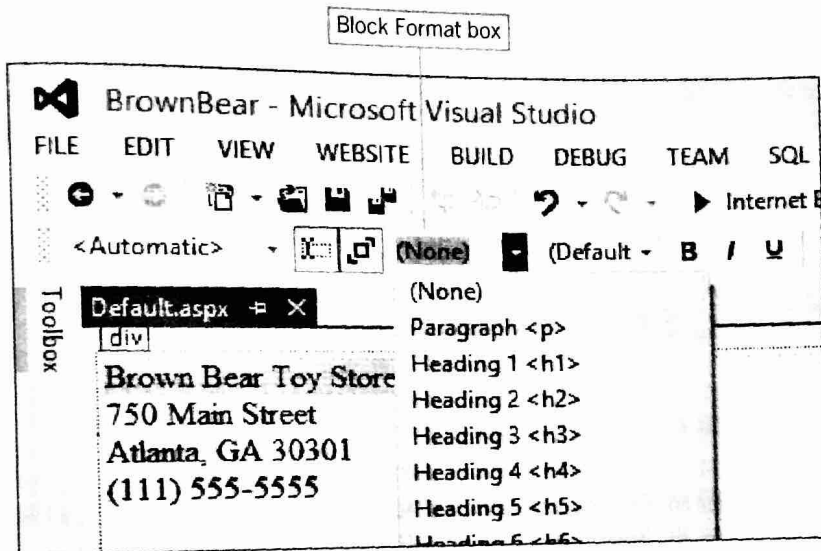


Figure 27-10 Result of clicking the arrow in the Block Format box

2. Click **Heading 1 <h1>**.
3. Select the address and phone number text on the Web page. Click the **down arrow** in the Block Format box and then click **Heading 2 <h2>**.
4. Now, you will use the Formatting toolbar's Alignment button to center all of the static text. Select all of the static text on the Web page and then click the **down arrow** on the Alignment button. See Figure 27-11.



Figure 27-11 Result of clicking the Alignment button

5. Click **Justify Center**. The selected text appears centered, horizontally, on the Web page. Click **anywhere below the phone number** to deselect the text, and then save the application.

Viewing a Web Page in Full Screen View

While you are designing a Web page, you can use the Full Screen option on the VIEW menu to determine how the Web page will appear to the user.

To view the Web page using the Full Screen option:

1. Click **VIEW** on the menu bar and then click **Full Screen** on the menu. See Figure 27-12. Although not identical to viewing in a browser window, full screen view provides a quick and easy way to verify the placement of controls and text on the Web page.

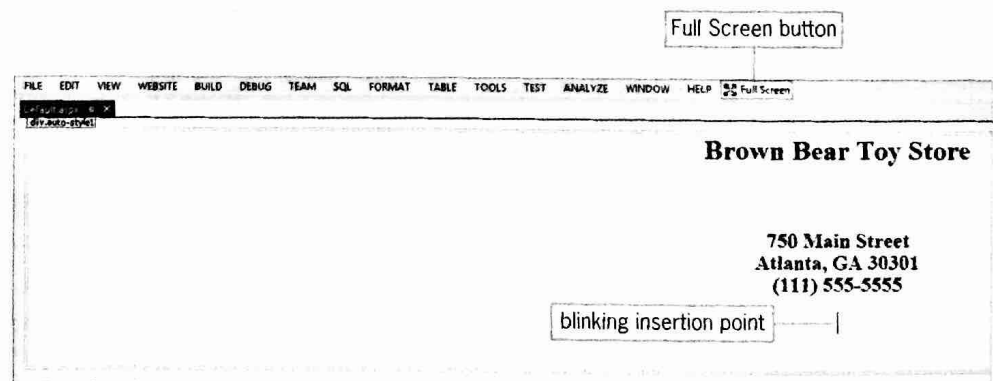


Figure 27-12 Default.aspx Web page displayed in full screen view

2. Click the **Full Screen** button to return to the standard view. (If you mistakenly clicked the window's Close button, click the Full Screen button, right-click Default.aspx in the Solution Explorer window, and then click View Designer.)

Adding Another Web Page to the Application

In the next set of steps, you will add a second Web page to the application. The Web page will display the store's hours of operation.

To add another Web page to the application:

1. Click **WEBSITE** on the menu bar and then click **Add New Item**. (If **WEBSITE** does not appear on the menu bar, click the Web application's name in the Solution Explorer window.)
2. If necessary, click **Visual Basic** in the Installed list and then (if necessary) click **Web Form** in the middle column of the dialog box. Change the filename in the Name box to **Hours** and then click the **Add** button. The computer appends the .aspx extension on the filename and then displays the Hours.aspx Web page in a separate Document window.
3. Temporarily display the Solution Explorer window. Notice that the window now contains the Hours.aspx filename.
4. Click the **Hours.aspx** tab and then temporarily display the Properties window. Click the **down arrow** button in the Properties window's Object box and then click **DOCUMENT** in the list. Change the Web page's Title property to **Brown Bear Toy Store**.
5. Click the **Hours.aspx** tab. The blinking insertion point should be inside the rectangle that appears below the div tag. (If the div tag does not appear in the Document window, click the <div> button at the bottom of the IDE.) Type **Please visit us during these hours:** and press **Enter** twice.
6. Next, enter the following three lines of text. Press **Enter** twice after typing the last line.
Monday – Friday 8am – 10pm
Saturday 9am – 6pm
Closed Sunday
7. Select the Please visit us during these hours: text. Click the **down arrow** in the Font Size box on the Formatting toolbar and then click **x-large (24pt)**. Also click the **I** (Italic) button on the Formatting toolbar.
8. Select the three lines of text that contain the store hours. Click the **down arrow** in the Font Size box and then click **large (18pt)**. Also click the **B** (Bold) button on the Formatting toolbar.
9. Now, you will change the color of the selected text. Click the **Foreground Color** button on the Formatting toolbar to open the More Colors dialog box. Click **any red hexagon** and then click the **OK** button.
10. Select all of the static text on the Web page. Click the **down arrow** on the Alignment button and then click **Justify Center**.
11. Click the **second blank line** below the store hours to deselect the text, and then save the application.

Adding a Link Button Control to a Web Page

The Toolbox window provides tools for adding controls to a Web page. In the next set of steps, you will add a **link button control** to both Web pages. When it is clicked, the link button control on the Default.aspx page will display the Hours.aspx page. The link button control on the Hours.aspx page, on the other hand, will return the user to the Default.aspx page.

To add a link button control to both Web pages:

1. First, you will add a link button control to the Hours.aspx page. Permanently display the Toolbox window. Expand the Standard node, if necessary, and then click the **LinkButton** tool. Drag your mouse pointer to the Web page, positioning it in the second line below the Closed Sunday text, and then release the mouse button. See Figure 27-13.

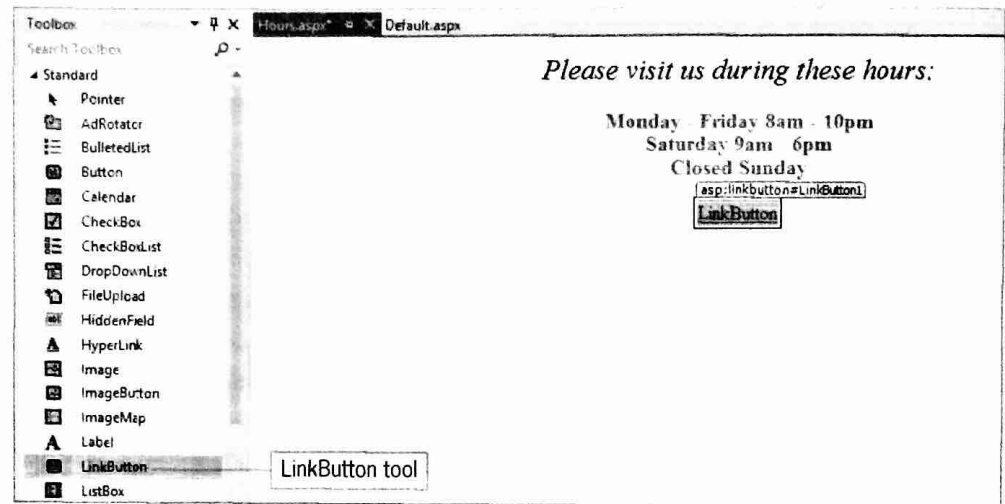


Figure 27-13 Link button control added to the Hours.aspx Web page

2. Temporarily display the Properties window. Change the control's Text property to **Home Page**. Click **PostBackUrl** in the Properties list and then click the ... (ellipsis) button to open the Select URL dialog box. Click **Default.aspx** in the Contents of folder list. See Figure 27-14.

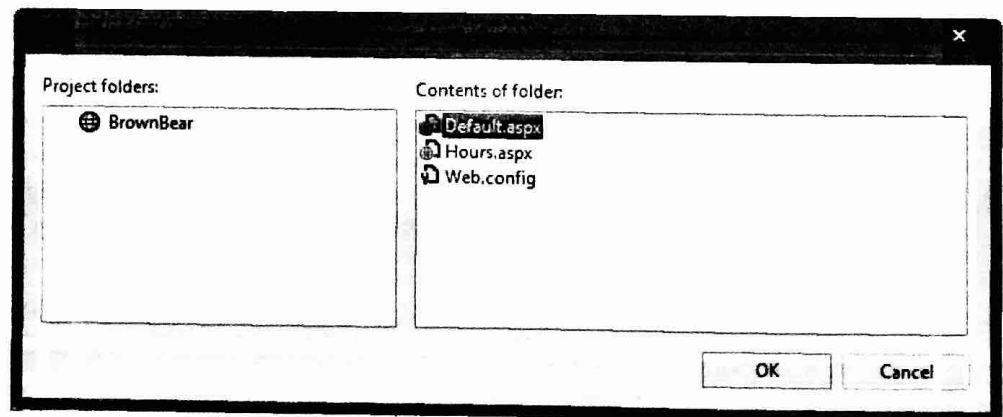


Figure 27-14 Select URL dialog box

3. Click the **OK** button to close the dialog box and then click the **Web page**.
4. Now you will add a link button control to the Default.aspx page. Click the **Default.aspx** tab. Click the **LinkButton** tool. Drag your mouse pointer to the Web page, positioning it two lines below the phone number, and then release the mouse button. See Figure 27-15.

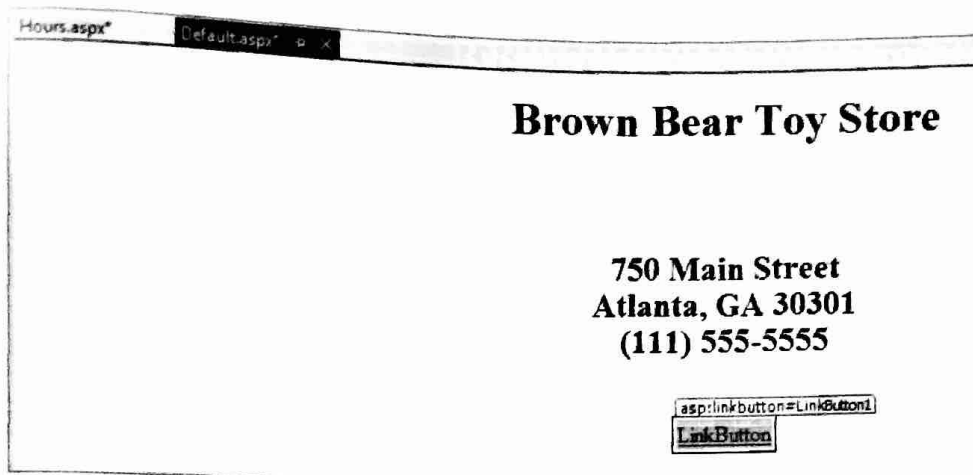


Figure 27-15 Link button control added to the Default.aspx Web page

5. Temporarily display the Properties window. Change the control's Text property to **Store Hours** and then change its PostBackUrl property to **Hours.aspx**. Click the **OK** button to close the Select URL dialog box.
6. Click the **Web page** and then save the application.

Starting a Web Application

Typically, you start a Web application either by pressing Ctrl+F5 or by clicking the Start Without Debugging option on the **DEBUG** menu. The method you use—the shortcut keys or the menu option—is a matter of personal preference. If you prefer to use a menu option, you might need to add the Start Without Debugging option to the **DEBUG** menu. You can add the option to the menu by performing the next set of steps. If you prefer to use the Ctrl+F5 shortcut keys, you can skip the next set of steps.

To add the Start Without Debugging option to the **DEBUG** menu:

1. First, you will determine whether your **DEBUG** menu already contains the Start Without Debugging option. Click **DEBUG** on the menu bar. If the **DEBUG** menu contains the Start Without Debugging option, close the menu by clicking **DEBUG** again, and then skip the remaining steps in this set of steps.
2. If the **DEBUG** menu does *not* contain the Start Without Debugging option, close the menu by clicking **DEBUG** again. Click **TOOLS** on the menu bar and then click **Customize** to open the Customize dialog box.
3. Click the **Commands** tab. The Menu bar radio button should be selected. Click the **down arrow** in the Menu bar list box. Scroll down the list until you see **Debug**, and then click **Debug**.
4. Click the **Add Command** button to open the Add Command dialog box, and then click **Debug** in the Categories list. Scroll down the Commands list until you see **Start Without Debugging**, and then click **Start Without Debugging**. Click the **OK** button to close the Add Command dialog box.
5. Click the **Move Down** button until the Start Without Debugging option appears below the Start/Continue option, as shown in Figure 27-16. (Your dialog box will look slightly different if you are using Visual Studio Express 2012 for Web.)

3. A link button control's _____ property specifies the Web page that will appear when the control is clicked.
 - a. PostBackUrl
 - b. PostBack
 - c. Url
 - d. none of the above

Dynamic Web Pages

A dynamic Web page contains controls with which the user can interact. It also contains code that tells the controls how to respond to the user's actions. In the following sections, you will finish creating the dynamic Web page shown earlier in Figure 27-3. The Web page provides a text box for the user to enter the number of American dollars. When the user clicks the page's Submit button, the button's Click event procedure displays the corresponding number of Mexican pesos in a label control. In the next set of steps, you will add the text box, label, and button control to the Web page.

To add the controls to the Web page:

1. If necessary, permanently display the Solution Explorer, Properties, and Toolbox windows.
2. Click **FILE** on the menu bar and then click **Open Web Site**. If necessary, click the **File System** button in the Open Web Site dialog box. Click the **DollarsToPesos** folder. If a message box appears and asks whether you want to use IIS Express or the Visual Studio Development Server, click the **Yes** button to use IIS Express.
3. If the Default.aspx page is not open in the Document window, right-click **Default.aspx** in the Solution Explorer window and then click **View Designer**. Position the blinking insertion point as shown in Figure 27-23. (When the static text was entered on the Web page, two space characters were included after each colon.)

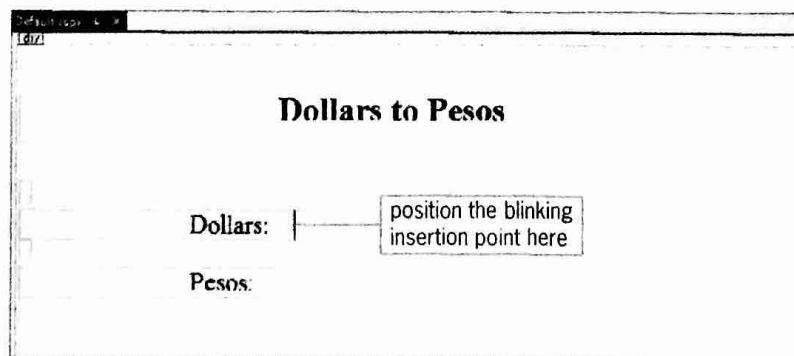


Figure 27-23 Partially completed Default.aspx page

4. In addition to the static text, the Web page will contain a text box, a label, and a button. Drag a text box control to the Web page. Position the control at the location of the blinking insertion point and then release the mouse button. Next, position the blinking insertion point as shown in Figure 27-24.

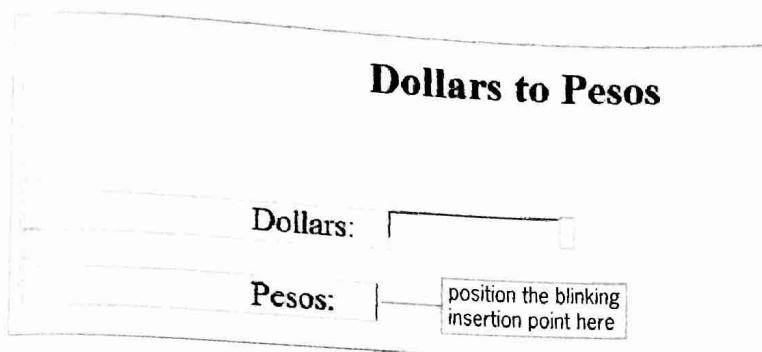


Figure 27-24 Text box added to the Web page

5. Drag a label control to the Web page. Position the control at the location of the blinking insertion point and then release the mouse button.
6. Click the **blank line** below the Pesos: text and then press **Enter** to insert another blank line. If necessary, press **Tab** until the blinking insertion point is aligned with the P in Pesos:. Now, drag a button control to the Web page. Position the control at the location of the blinking insertion point and then release the mouse button. See Figure 27-25.

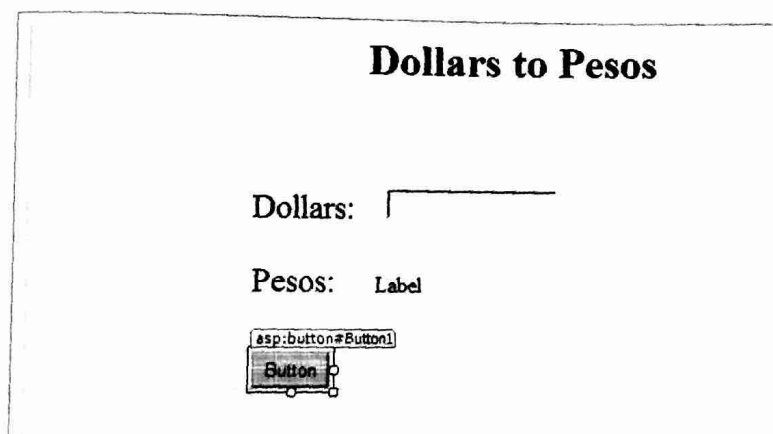


Figure 27-25 Controls added to the Web page

7. Auto-hide the Toolbox window.

Next, you will complete the interface by setting the appropriate properties of the controls.

To complete the interface:

1. Unlike Windows controls, Web controls have an ID property rather than a Name property. Use the Properties window to set the button control's ID property (which appears at the top of the Properties list) to **btnSubmit**. Then, set its Text property to **Submit**.
2. Click the **label control** on the Web page and then set its properties as follows:

ID	lblPesos
BorderStyle	Solid
BorderWidth	1px
Text	0
Width	90px

3. Change the label control's BackColor property to a pale yellow.
4. Click the **text box control**. Change the control's ID and Width properties to **txtDollars** and **90px**, respectively.
5. Click a **blank area** on the Web page. See Figure 27-26.

Figure 27-26 Completed interface for the Default.aspx page

6. Auto-hide the Properties window.

Coding the Submit Button's Click Event Procedure

The Submit button's Click event procedure should convert the number of American dollars to Mexican pesos and then display the result on the Web page. At the time of this writing, an American dollar was equivalent to 12.75 Mexican pesos. As you do when coding a control on a Windows form, you enter the code for a control on a Web page in the Code Editor window.

To code the Submit button's Click event procedure:

1. Right-click the **Web page** and then click **View Code**. The Default.aspx.vb window opens. The .vb extension on the filename indicates that the file contains Visual Basic code. In this case, the file is referred to as the code-behind file because it contains code that supports the Web page. See Figure 27-27.

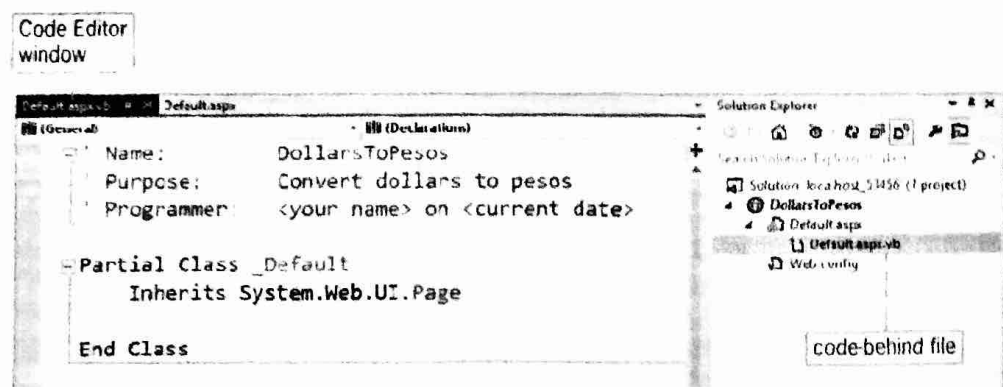


Figure 27-27 Code Editor and Solution Explorer windows

2. Replace <your name> and <current date> in the comments with your name and the current date, respectively.
3. Auto-hide the Solution Explorer window. Open the btnSubmit control's Click event procedure. Type the following comment and then press **Enter** twice:
' converts dollars to pesos
4. The procedure will use a Double named constant to store the conversion rate of 12.75. Enter the following Const statement:
Const dblPESO_RATE As Double = 12.75
5. The procedure will use two Double variables to store the number of American dollars and the number of Mexican pesos. Enter the following Dim statements. Press **Enter** twice after typing the second Dim statement.
Dim dblDollars As Double
Dim dblPesos As Double
6. Now, you will store the user's entry in the dblDollars variable. Enter the following TryParse method:
Double.TryParse(txtDollars.Text, dblDollars)
7. Next, you will convert the dollars to pesos and then store the result in the dblPesos variable. Enter the following assignment statement:
dblPesos = dblDollars * dblPESO_RATE
8. Finally, you will display the number of pesos in the lblPesos control. Enter the following assignment statement:
lblPesos.Text = dblPesos.ToString("N2")

Figure 27-28 shows the code entered in the btnSubmit_Click procedure.

```
Protected Sub btnSubmit_Click(sender As Object, e As
    ' converts dollars to pesos

    Const dblPESO_RATE As Double = 12.75
    Dim dblDollars As Double
    Dim dblPesos As Double

    Double.TryParse(txtDollars.Text, dblDollars)
    dblPesos = dblDollars * dblPESO_RATE
    lblPesos.Text = dblPesos.ToString("N2")
End Sub
```

Figure 27-28 btnSubmit_Click procedure

In the next set of steps, you will test the application to verify that it is working correctly.

To test the Dollars to Pesos application:

1. Save the application, and then start it by pressing **Ctrl+F5**. Your browser requests the Default.aspx page from the Web server. The server locates the page and then sends the appropriate HTML instructions to your browser for rendering on the screen.

2. Click the **Dollars** box and then type **10**. Click the **Submit** button; doing this submits your entry to the Web server, along with a request for additional services. (If the message "Do you want AutoComplete to remember web form entries?" appears, click the No button.) The server processes the code contained in the button's Click event procedure and then sends the appropriate HTML to the browser for rendering on the screen. See Figure 27-29.

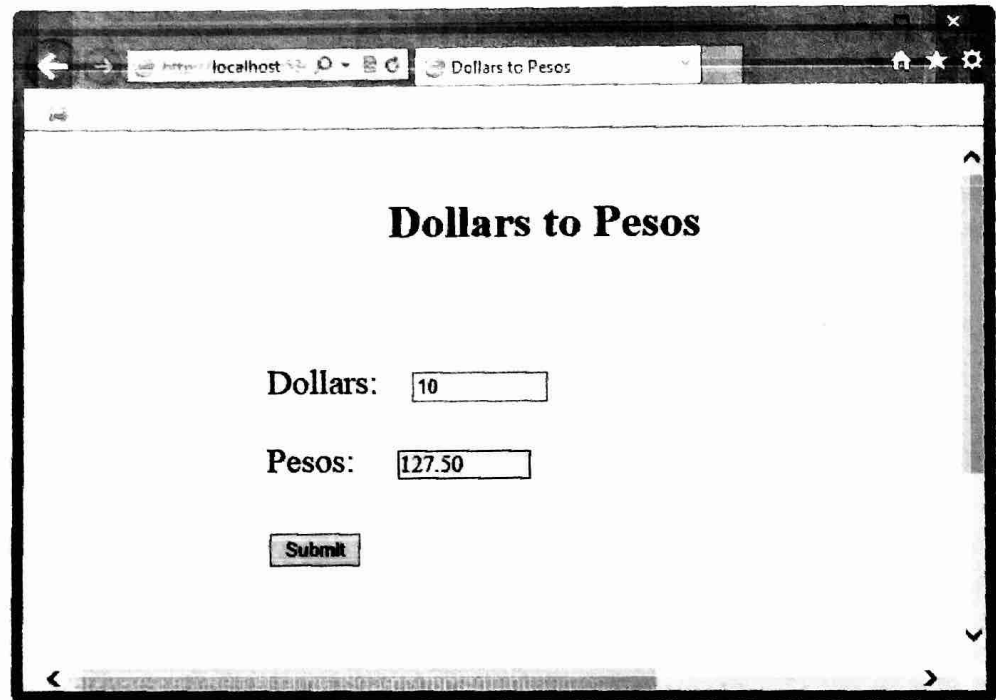


Figure 27-29 Result of clicking the Submit button

3. Close the browser window and then close the Code Editor window.

Can You Validate This?

The Validation section of the toolbox provides several tools for validating user input. The tools are referred to as **validator tools**. The name, purpose, and important properties of each validator tool are listed in Figure 27-30. In the Dollars to Pesos application, you will use a `RequiredFieldValidator` control to verify that the user entered the number of dollars.

Name	Purpose	Properties
CompareValidator	compare an entry with a constant value or the property stored in a control	ControlToCompare ControlToValidate ErrorMessage Operator Type ValueToCompare
CustomValidator	verify that an entry passes the specified validation logic	ClientValidationFunction ControlToValidate ErrorMessage
RangeValidator	verify that an entry is within the specified minimum and maximum values	ControlToValidate ErrorMessage MaximumValue MinimumValue Type
RegularExpressionValidator	verify that an entry matches a specific pattern	ControlToValidate ErrorMessage ValidationExpression
RequiredFieldValidator	verify that a control contains data	ControlToValidate ErrorMessage
ValidationSummary	display all of the validation error messages in a single location on a Web page	DisplayMode HeaderText

Figure 27-30 Validator tools

© Cengage Learning 2013

To verify that the user entered the number of dollars:

1. Click **to the immediate right of the txtDollars control** and then press the **Spacebar** three times.
2. Permanently display the Toolbox window. If necessary, expand the Validation section. Click the **RequiredFieldValidator** tool and then drag your mouse pointer to the Web page. Position your mouse pointer to the right of the txtDollars control and then release the mouse button. The RequiredFieldValidator1 control appears on the Web page.
3. Temporarily display the Properties window. Set the following properties for the RequiredFieldValidator1 control:

ControlToValidate	txtDollars
ErrorMessage	Required entry
ForeColor	choose a red hexagon
4. Click an **empty area** on the Web page and then auto-hide the Toolbox window. Save the application.

5. Start the application by pressing **Ctrl+F5**. Click the **Submit** button without entering a value in the txtDollars control. (If a Web page opens and displays the "Server Error in '/' Application." message, refer to the Important note that follows Figure 27-31.) The RequiredFieldValidator control displays the "Required entry" message, as shown in Figure 27-31.

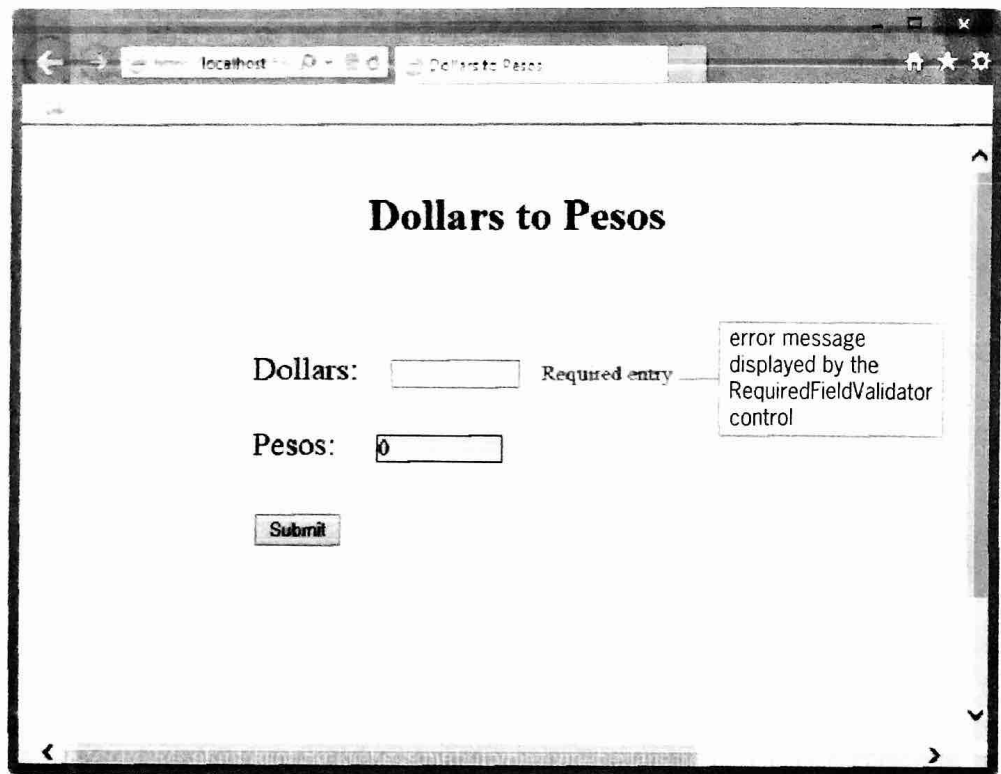


Figure 27-31 Result of clicking the Submit button when the Dollars box is empty

Important note: At the time of this writing, there was an unresolved issue with some of the validator controls. Until this problem is fixed, you can use the following workaround: First, close the browser window. Next, right-click Web.config in the Solution Explorer window and then click Open. Change both occurrences of "4.5" to "4.0", and then save the application. Close the Web.config window and then repeat Step 5.

6. Click the **Dollars** box and then type **20**. Click the **Submit** button. The error message is removed from the Web page and the number 255.00 appears in the lblPesos control. See Figure 27-32.

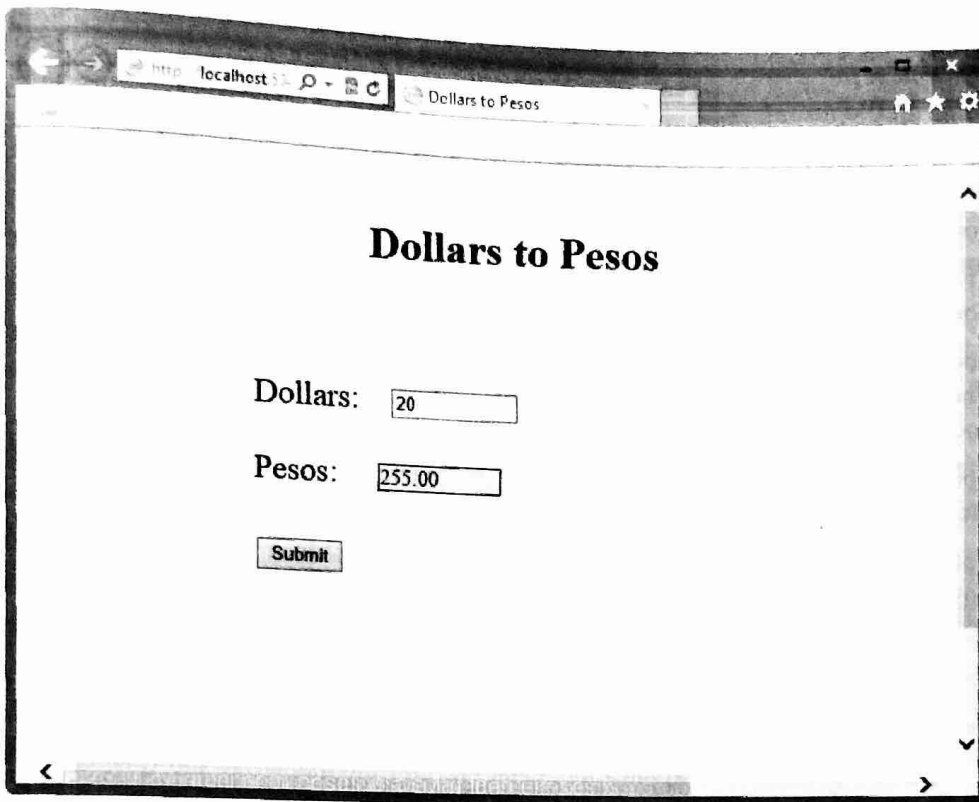


Figure 27-32 Number of pesos displayed in the Pesos box

7. Close the browser window and then close the application.

Mini-Quiz 27-3

See Appendix B for the answers.

1. The Visual Basic code in a Web page is processed by the _____.
 - a. client computer
 - b. Web server
 2. You can use a _____ control to verify that a control on a Web page contains data.
 - a. RequiredFieldValidator
 - b. RequiredField
 - c. RequiredValidator
 - d. none of the above
-

Summary

- The Web consists of Web pages that are stored on Web servers.
- A client computer uses a browser to request a Web page from a Web server. It also uses the browser to view the Web page.
- Web pages can be either static or dynamic (interactive).