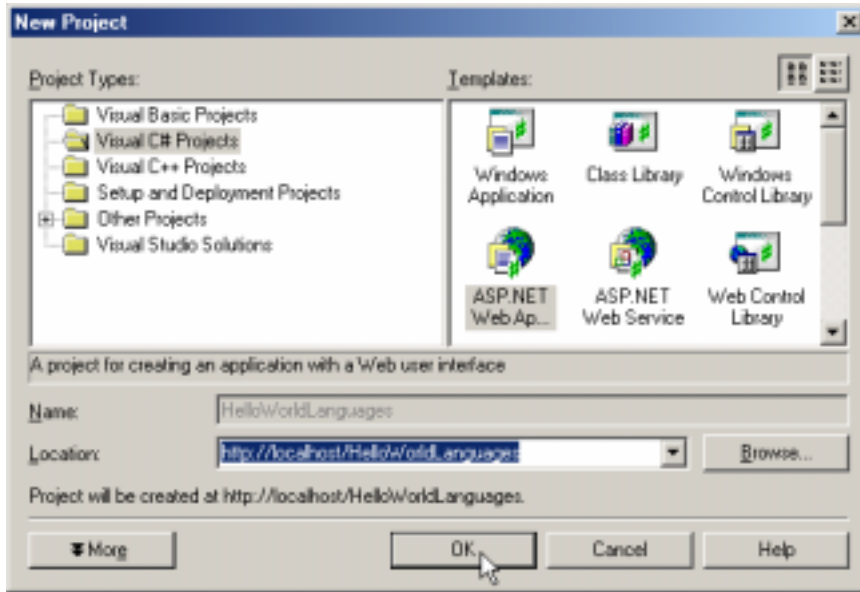


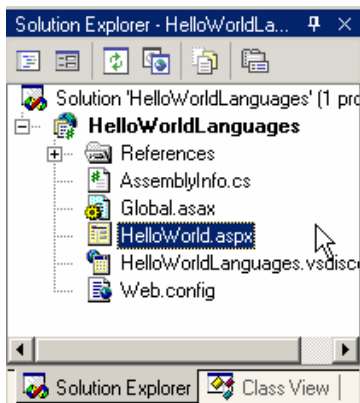
Tech Talk: ASP.NET  
Provided by Jenny Chang  
January 24, 2003

Walk Through Creating a Web Form in ASP.NET

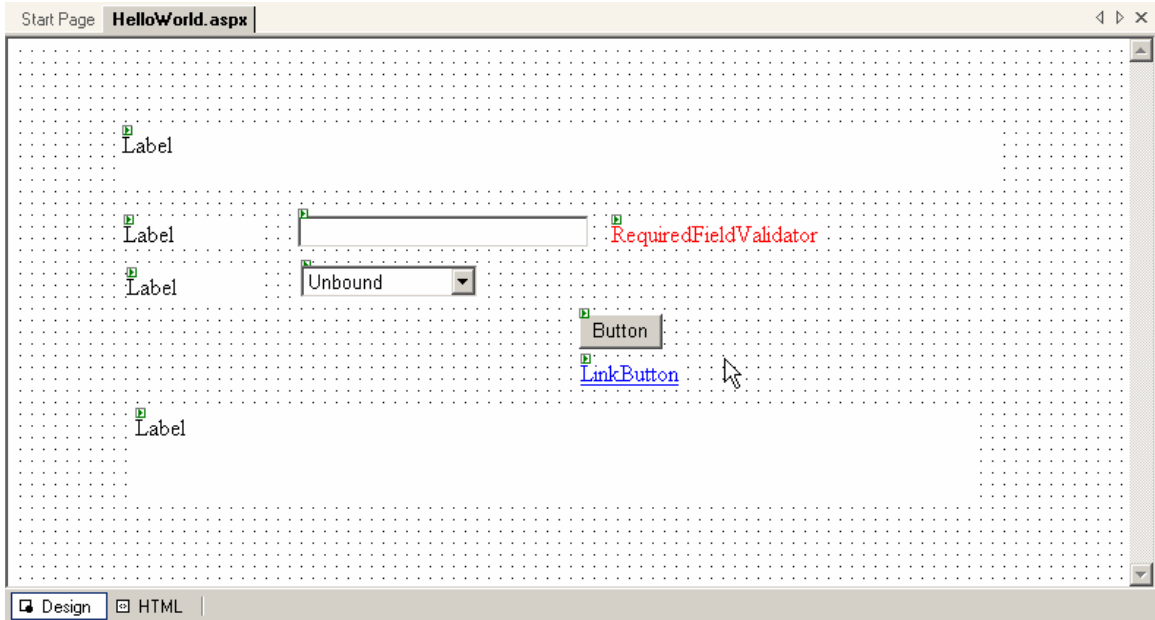
1. Open Visual Studio.NET
2. Click on **File>New> Project...** and select Visual C# Project and the ASP.NET Web Application template.
3. Name the application "HelloWorldLanguages"



4. Click OK
5. Once your application is successfully create, in the Solution Explorer, rename the file and class from WebForm1 to HelloWorld



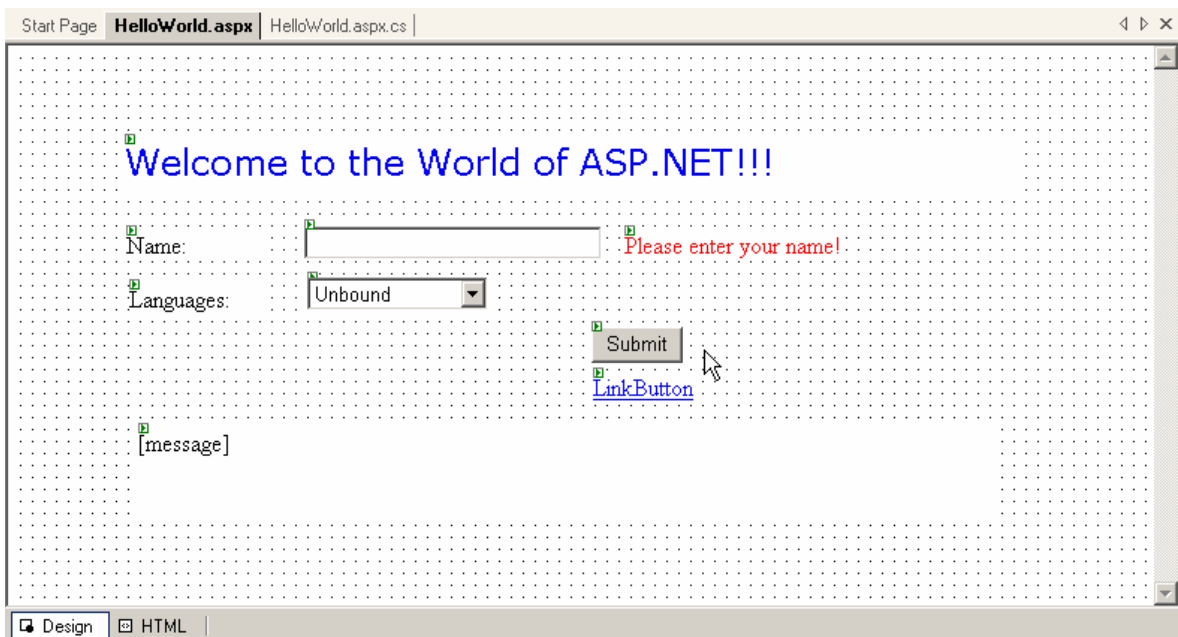
6. Let's add several controls to the Web form. Under Toolbox, click on Web Forms tab and add 4 labels, 1 textfield, 1 drop down list, 1 button, 1 RequiredFieldValidator, and 1 LinkButton to the Web form. Your form should look like the following:



7. Click on the first label and go to the Properties windows.
8. Rename the text field to "Welcome to the World of ASP.NET!!!"
9. Change the Font to "Verdana," set the Size to "Large," and set the ForeColor to "Blue" or to whatever you like.
10. Next, change the text field for the second and third label to "Name:" and "Languages" respectively.
11. Click on the last label and set the text field to blank, and change the ID to "message"
12. Click on the textfield and set the ID to "yourName"
13. Click on the drop down list and set the ID to "ddlLanguages"
14. Lastly, change the text field of the button to "Submit"
15. Click on "RequiredFieldValidator" and change the following properties:

*ErrorMessage* = "Please enter your name!"  
*ControlToValidate* = "yourName"

16. Your form should now look as follows:



17. Now, double click on the Submit button to open up the "code behind." We want to implement the event when this button is clicked by the client.
18. Inside the Page\_Load method, type the following statements:

```
if (!IsPostBack)
{
    ddlLanguages.Items.Add("English");
    ddlLanguages.Items.Add("Spanish");
    ddlLanguages.Items.Add("German");
    ddlLanguages.Items.Add("French");
}
```

Note: Whenever a page is first loaded or is not posting back, the above list is displayed in the drop down menu. So you don't have to re-add every time your page is reloaded or refreshed.

19. Inside the Button1\_Click method, type the following statements:

```
switch (ddlLanguages.SelectedItem.Text)
{
    case "English":
        message.Text = "Hello World, " + yourName.Text + "!";
        break;
    case "Spanish":
        message.Text = "Hola Mundo, " + yourName.Text + "!";
        break;
    case "German":
        message.Text = "Hallo Welt, " + yourName.Text + "!";
        break;
    case "French":
        message.Text = "Bonjour Monde, " + yourName.Text + "!";
        break;
}
```

20. Hit **Build>Build Solution**
21. In Solution Explorer, right click on file "HelloWorld.aspx" and select "View in Browser"
22. Enter your name and hit the Submit button. Try leaving the name field blank and hit the Submit button.
23. Congrats! You have successfully created your first ASP.NET page!

### Understanding Request and Response Objects

While the Request and Response objects are less important in Visual Studio.NET than they were in "classic" ASP due to PostBack and CodeBehind, they still can be used to obtain information from or send information to the web browser. This tutorial demonstrates how these objects can be used in Visual Studio.NET.

1. Open Visual Studio.Net
2. Create ASP.NET Application project called "ResponseRequest"
3. On WebForm1.aspx, drag and drop 2 buttons and 2 hyperlink controls from the Toolbox
4. On the property window for the first button, rename the Text value to "Click here for Response.Write". This will send a response back to the same page.
5. Double-click on the button to open the button click event. In the method, type the following:

**Response.Write("Responding with a Response.Write");**

6. On the property window for the second button, rename the Text value to "Click here for Response.Redirect". This will send a response to another page by redirecting the values.
7. Double-click on the second button to open the button click event. In the method, type the following:

**Response.Redirect("WebForm2.aspx");**

8. On the property window for the first hyperlink, rename the Text value to "Sending value 1 through somevalue". Still on the property window, set the NavigateUrl to "WebForm2.aspx?somevalue=1"
9. On the property window for the first hyperlink, rename the Text value to "Sending value 2 through somevalue". Still on the property window, set the NavigateUrl to "WebForm2.aspx?somevalue=2"
10. Add a new Web form to the project by clicking Project → Add Web Form...
11. Then click Open
12. On WebForm2.aspx, add a label from the Toolbox and rename the Text to "myLabel" in the property window.
13. Double-click on the WebForm2.aspx design view, then type the following:

**Label1.Text = "This is the result from query string: " + Request.QueryString["somevalue"];**

14. Click Build → Build Solution to compile and build the project.
15. View the WebForm1.aspx on your browser.
16. Congratulations! You are now more familiar with response and request objects in ASP.NET!