

ASP.NET: THE NEW PARADIGM FOR WEB APPLICATION DEVELOPMENT

Dr. Mike Morrison, University of Wisconsin-Eau Claire, morriscm@uwec.edu
Dr. Joline Morrison, University of Wisconsin-Eau Claire, [morriskjp@uwec.edu](mailto:morrisjp@uwec.edu)

ABSTRACT

ASP.NET is a set of class libraries within Microsoft's .NET framework that contain code to define objects to create Web applications. This paper describes the ASP.NET development and processing environment, and describes how this technology provides a significantly improved approach for creating Web-based applications.

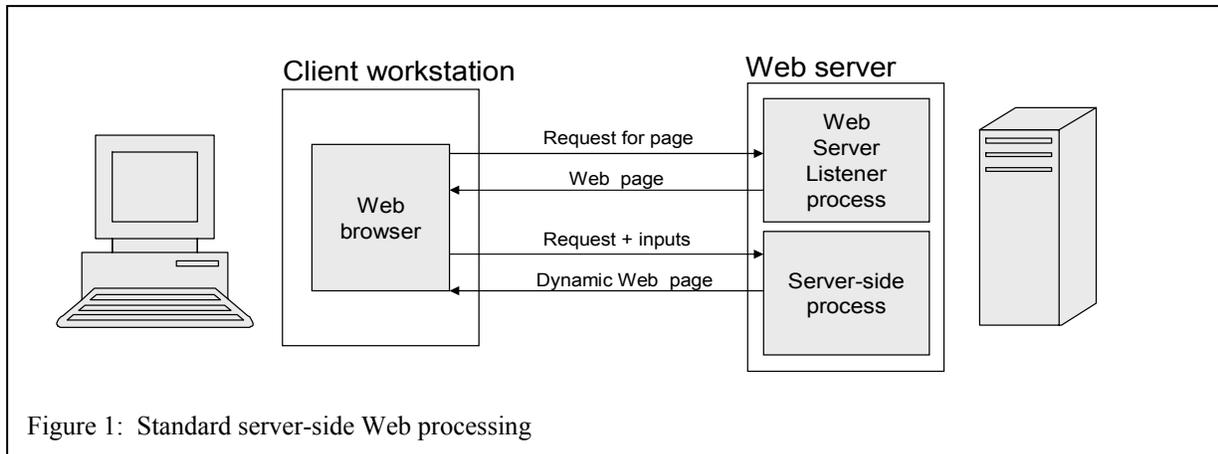
Keywords: ASP.NET, Web application development

INTRODUCTION

Consumers are using interactive World Wide Web applications for tasks ranging from e-commerce order input and tracking to tax return preparation. Most interactive Web applications are developed using traditional HTML Web forms in conjunction with Web processing technologies such as Active Server Pages (ASPs), Java Server Pages (JSPs), and PHP: Hypertext Preprocessor (PHP). These applications have limited interface options, and do not easily support a high degree of interaction between the user and the Web server. As these applications become larger and more complex, they become difficult to develop and maintain because the program code is embedded directly within the Web page HTML tags and text. Processing proceeds in a linear fashion, and communication between the user and the Web server is limited. ASP.NET provides an innovative new paradigm for creating interactive Web applications that are easier to develop and maintain, provide richer user interfaces, and support a higher level of interaction between the user and the Web server.

OVERVIEW OF EXISTING WEB PROCESSING TECHNOLOGIES

Most interactive Web applications use server-side processing, in which either a compiled program or script executes on the Web server. This program or script processes user inputs on HTML forms, and delivers appropriate responses as formatted Web pages. Figure 1 illustrates standard server-side Web processing.



In standard server-side Web processing, the client workstation requests a Web page from the Web server. The user enters inputs on this Web page, and submits the input values to the Web server. The Web server then executes a server-side process which processes the inputs, interacts with files and databases, and creates a dynamic Web page based on the user inputs. Interactive processing tasks such as error trapping are accomplished using client-side scripts written in languages such as JavaScript. These client-side scripts run on the user's workstation, and do not interact with the Web server.

If the server-side process is written using ASP, JSP, and PHP, the commands are embedded within the HTML tags and text of Web pages. If the server-side process uses the Common Gateway Interface (CGI) protocol for communicating with the Web server, the program embeds HTML tags and text within programmatic content. In either case, the Web page content (such as text, graphic images, and formatting instructions) is interleaved with the program content. As a result, these interactive Web pages become unwieldy, and are difficult to understand, maintain, and debug.

HTML forms have a limited set of interface elements, consisting of text input boxes, labels, lists, radio buttons, check boxes, radio buttons, submit buttons, and command buttons. If developers want to create more sophisticated elements such as interactive calendars or interactive grids to display data, they must do so using a complex set of HTML form elements combined with custom programmed server and client-side scripts and programs. Custom client-side scripts are also needed to validate user inputs.

OVERVIEW OF ASP.NET

Microsoft Corporation developed ASP.NET within its .NET framework to provide tools and a development environment to create highly interactive, complex Web applications. ASP.NET provides a set of class libraries containing code to define objects to create Web applications. ASP.NET relies upon *Web forms*, which are text files with .aspx file extensions. Web forms contain standard Web page elements such as text, command buttons, and text input boxes; but in addition they support a number of additional elements that aren't directly supported with existing HTML elements. These include a calendar and a variety of data oriented elements such as datagrids and data lists to support data based applications. These Web page elements can be

linked to programs that execute on the Web server, and respond to user events such as clicking a button or changing the text in a textbox. This allows Web developers to create Web pages using event-driven programming which provides a richer and more interactive user interface.

Web Form Processing Architecture

Figure 2 illustrates how a client browser interacts with a Web server during ASP.NET Web form processing. The client browser initially requests the Web form using the Web form's URL. The Web server processes the Web form, and renders the HTML source code for a Web page that represents the Web form elements. The user interacts with the Web page by performing actions or initiating events, such as entering data values or clicking command buttons. The browser records all of the events that the user raises on the HTML document in an event queue that it eventually communicates to the Web server.

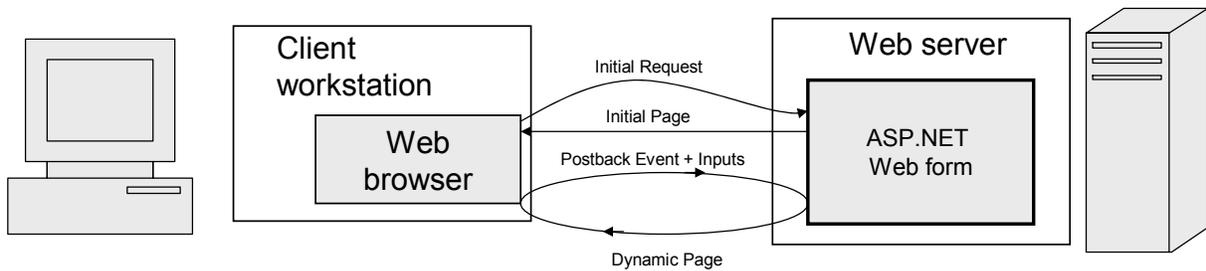


Figure 2: ASP.NET Web form processing

The Web form developer chooses which event (or events) will cause the browser to send, or *post*, the Web form back to the Web server. When the browser posts the Web form back to the server, the server runs the event handling procedures for each event that the user raised on the form. Every Web form has an associated *code behind file*, which is a text file that stores the form's event handlers. These event handlers can be written using Microsoft's VB .NET or C# programming languages. The .NET architecture was designed to support vendors desiring to adapt other languages for use with ASP.NET, may support additional language choices in the future. Separating the Web application into a Web form that defines the Web page appearance and an associated code behind file makes the Web application easier to develop, debug, and maintain.

ASP.NET processes user inputs based upon the code behind file's event handling code, and sends a modified HTML document back to the client's browser. The user can interact with this modified document, raise events, and post the Web form back to the server again. This process of repeatedly raising events on the client, executing event handlers on the server, and then returning a new HTML document to the client, is called *postback processing*.

Developing ASP.NET Applications in the Visual Studio .NET IDE

The Visual Studio .NET IDE provides multiple windows for defining Web form elements, creating and debugging code behind files, and preparing ASP.NET Web applications for deployment on a production Web server. Developers can use a drag and drop approach to create Web forms that display HTML form controls such as buttons, lists, textboxes, and radio buttons.

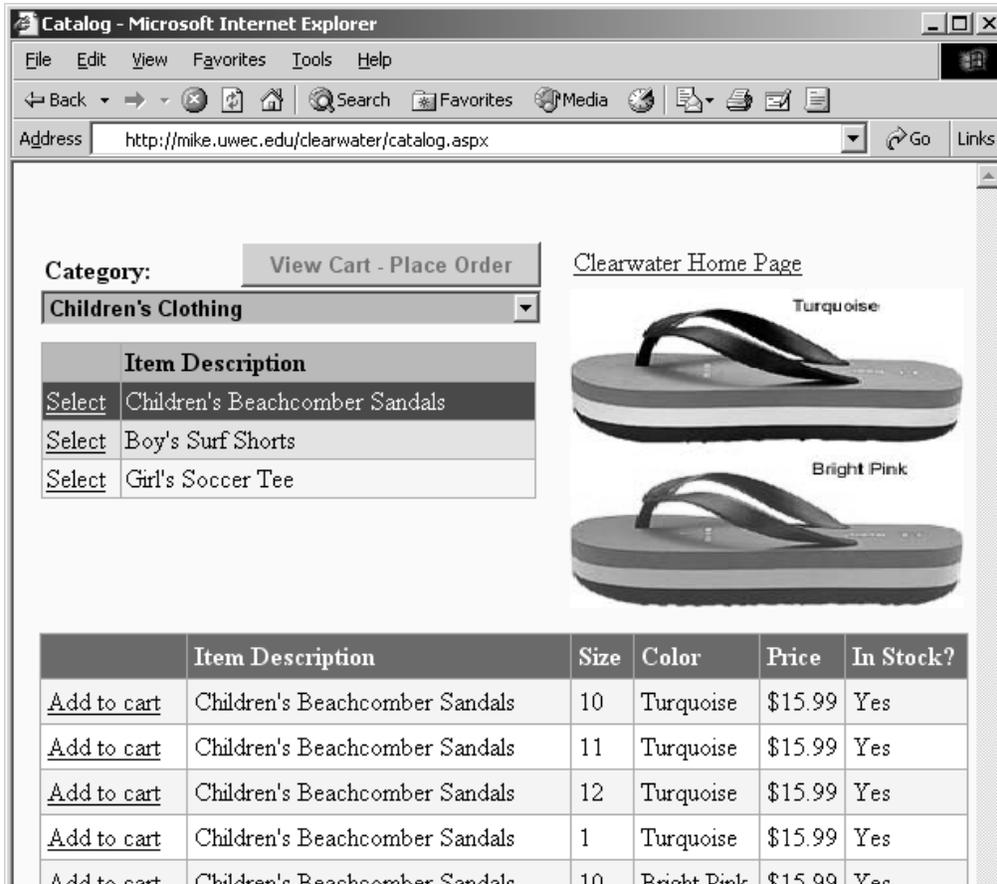
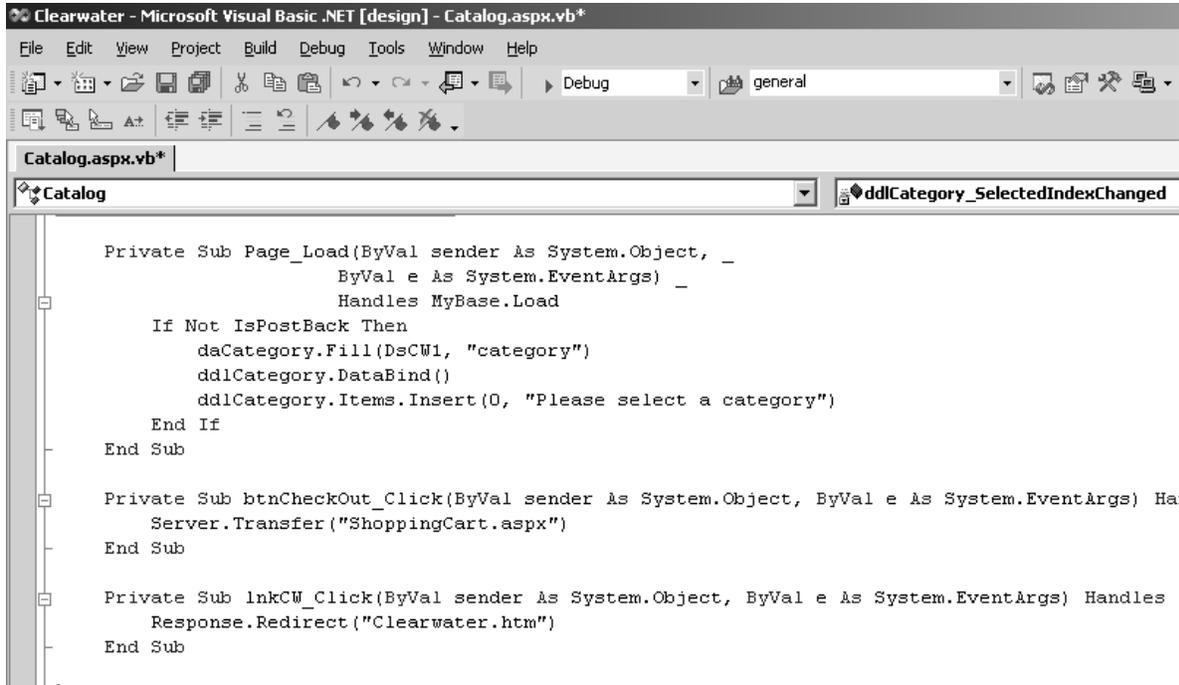


Figure 3 - Example Web form in the Visual Studio .NET IDE

Developers can also create *Web form controls*, which can have associated server-side event handlers. Web form controls can generate a single HTML form inputs such as buttons or text, or can generate a myriad of HTML elements corresponding to calendars and interactive data grid displays.

The IDE makes it very easy to visually create attractive Web pages. Moreover, developers can easily create and debug the event handlers associated with the Web page elements. Figure 3 shows an example Web form in the Visual Studio .NET that contains a variety of rich server controls.

Figure 4 shows an example of a Web form code behind file, which contains the event handlers for the form elements.



```
Clearwater - Microsoft Visual Basic .NET [design] - Catalog.aspx.vb*
File Edit View Project Build Debug Tools Window Help
Debug general
Catalog.aspx.vb*
Catalog ddlCategory_SelectedIndexChanged

Private Sub Page_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles MyBase.Load
    If Not IsPostBack Then
        daCategory.Fill(DsCW1, "category")
        ddlCategory.DataBind()
        ddlCategory.Items.Insert(0, "Please select a category")
    End If
End Sub

Private Sub btnCheckOut_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Ha:
    Server.Transfer("ShoppingCart.aspx")
End Sub

Private Sub lnkCW_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    Response.Redirect("Clearwater.htm")
End Sub
```

Figure 4 – Web form code behind file

Using Validation Controls to Validate Form Inputs

An important Web page programming task involves validating user inputs before the client browser submits a Web page to the Web server. Most Web applications use client-side scripts written in JavaScript to validate user inputs. These validation functions are embedded directly within the HTML document code, and can be tedious to program, and difficult to maintain. ASP.NET expedites form input validation through the use of validation controls, which are form elements that automatically generate client-side JavaScript functions to validate user inputs. When the Web server renders the Web form's HTML source code, it places commands in the code that call standard JavaScript validation functions that are in a JavaScript code library on the client workstation. Validation controls can confirm that the user entered a value in a required field, that the value is within a required range or within a specific set of allowable values, or that an input matches the format of a regular expression such as a date or telephone number. Figure 5 shows a Web form containing validation controls.

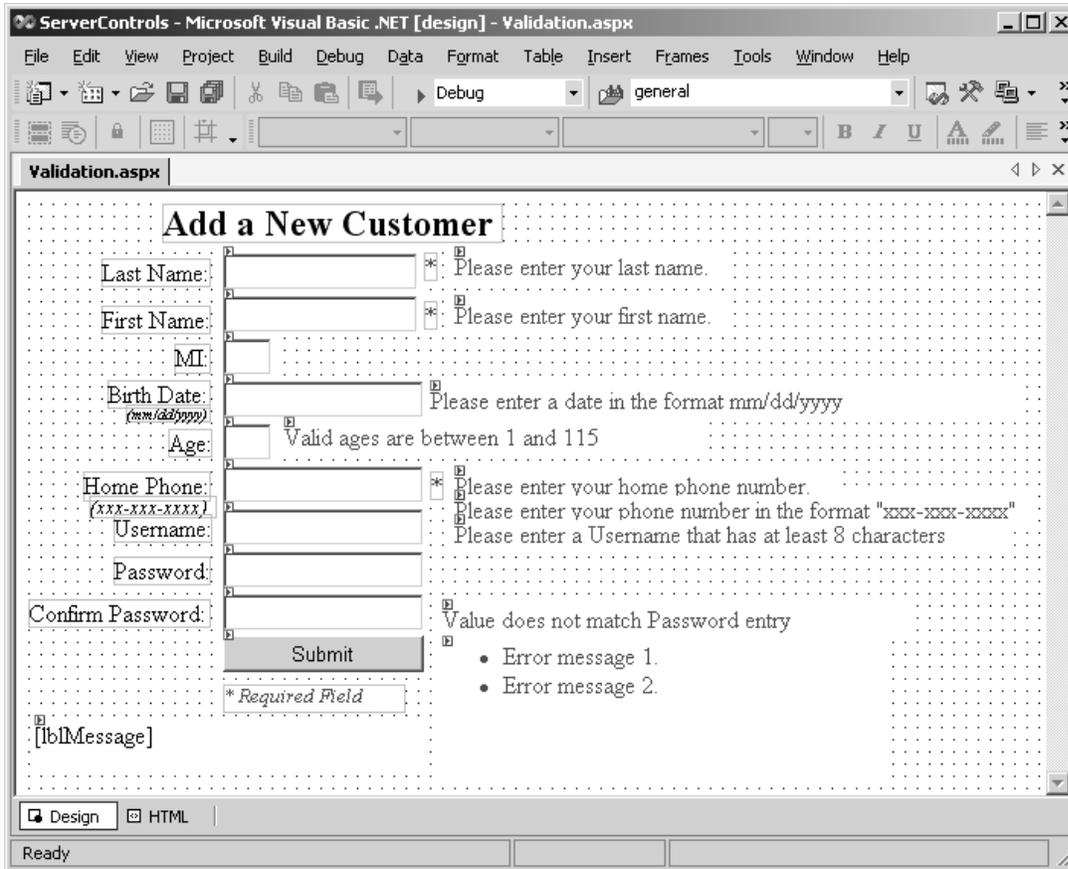


Figure 5 – Web form with validation controls

SUMMARY

ASP.NET addresses many limitations in current Web application development techniques. The Visual Studio .NET IDE provides an excellent environment for creating, debugging, and managing ASP.NET Web forms, which are text files with an .aspx extension that represent Web pages that allow a higher degree of interaction between the user and the Web server. Each Web form has an associated code behind file, which contains the code for event handlers that process user inputs and respond to user events. Separating the Web form's visual elements from its program code makes Web forms easier to create, debug, and maintain. Overall, ASP.NET provides a revolutionary environment that enables Web developers to quickly and easily create interactive Web pages.