

# ASP.NET

## Outline

- 23.1 Introduction**
- 23.2 .NET Overview**
  - 23.2.1 .NET Framework**
  - 23.2.2 ASP (Active Server Pages) .NET**
- 23.3 Setup**
- 23.4 JScript .NET**
- 23.5 A Simple ASP.NET Example**
- 23.6 Web Forms**
- 23.7 Session Tracking**
  - 23.7.1 Cookies**
  - 23.7.2 Session Tracking with HttpSessionState**
- 23.8 ASP.NET and XML**
- 23.9 Reading and Writing Text Files**
- 23.10 Connecting to a Database in ASP.NET**
- 23.11 Code-Behind Approach**
- 23.12 ASP.NET Web Services**
- 23.13 Web Resources**

# Objectives

- In this lesson, you will learn:
  - To program ASP.NET pages using JScript .NET.
  - To understand how ASP.NET pages work.
  - To understand the differences between client-side scripting and server-side scripting.
  - To create Web services.
  - To use and manipulate XML files with ASP.NET.
  - To understand Web forms and code-behind files.
  - To be able to use session tracking in an ASP.NET application.
  - To use ActiveX Data Objects .NET (ADO.NET) to access a database.

## 23.1 Introduction

- ASP.NET
  - Server-side technology that dynamically builds documents in response to client requests
  - Can be used on a server to create Web applications
  - Supports over 25 programming languages
  - Object-oriented programming

## 23.2 .NET Overview

- Independent from a specific programming language
- Promotes software reuse
- Include tools for porting, adapting existing software components
- Web services

## 23.2.1 .NET Framework

- Manages and executes applications
- Framework Class Library (FCL)
  - Enforces security and supplies many other programming capabilities
  - Reusable components that programmers can incorporate into their applications
- Common Language Runtime (CLR)
  - Executes programs written in any .NET-compatible programming language
- .NET Compact Framework

## 23.2.2 ASP (Active Server Pages) .NET

- Multi-tier, database-intensive applications
- Includes optimizations for performance, testing and security
- ASPX files
- XHTML documents
  - Static

## 23.3 Setup

- Microsoft .NET Framework
- .NET Framework System Development Kit (SDK)
  - Tools, examples, reference files and tutorials building .NET applications

## 23.3 Setup

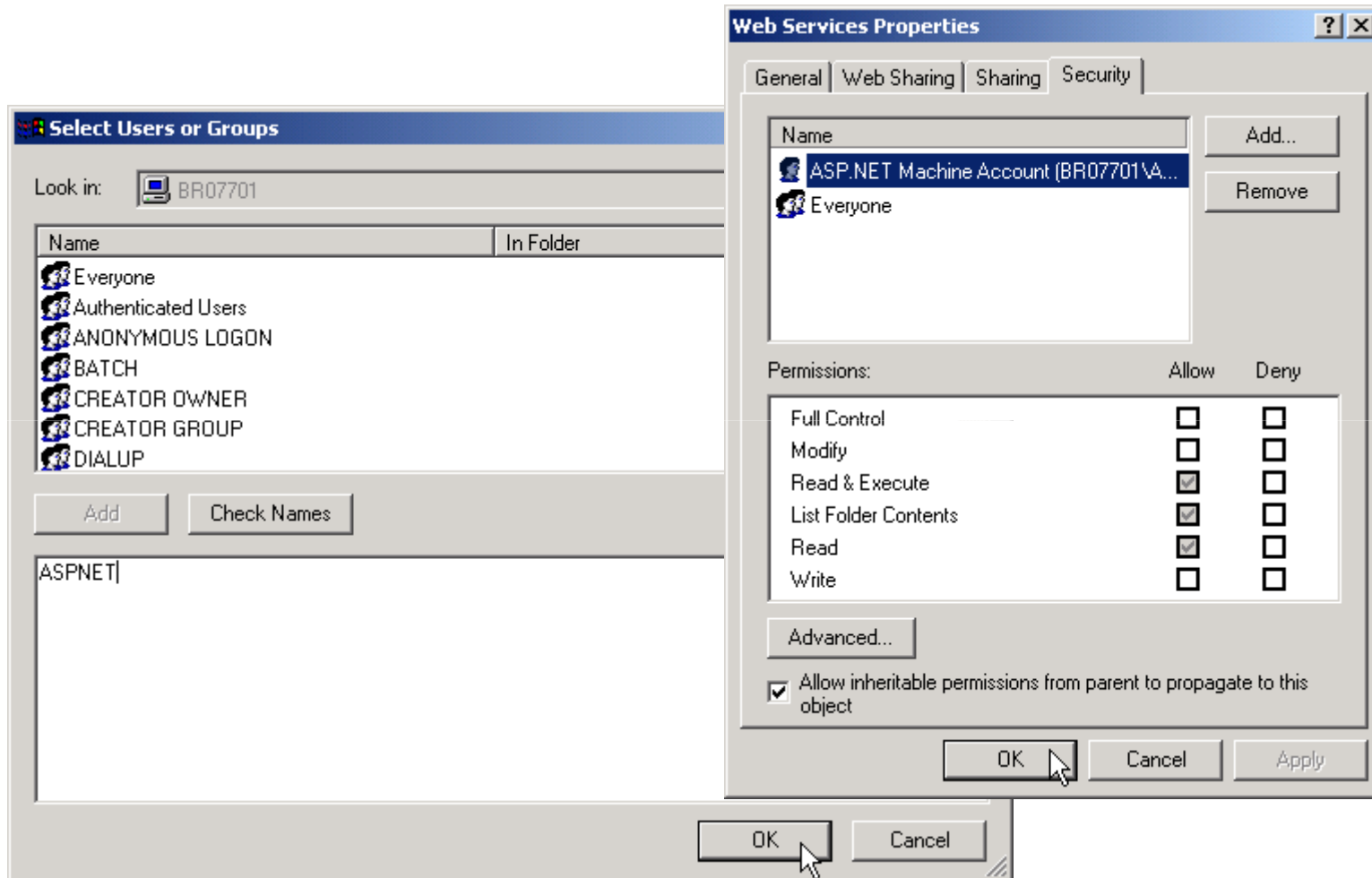


Fig. 23.1 Adding ASP.NET security permissions to a folder (on the NTFS file system).



## 23.3 Setup

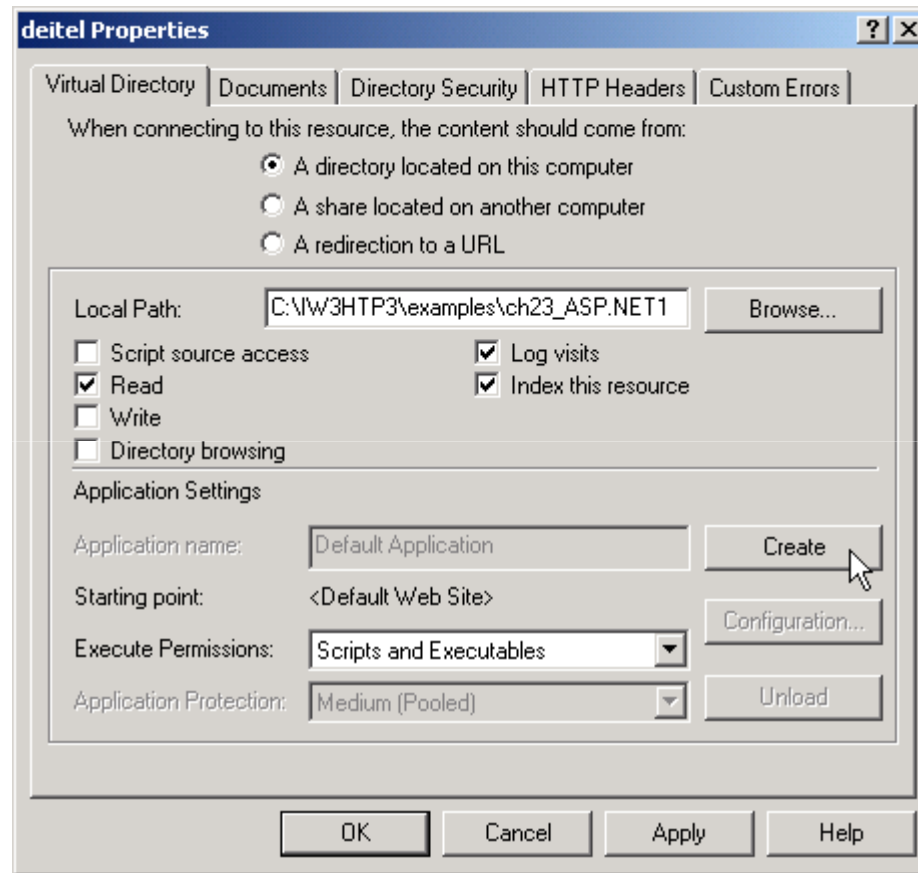


Fig. 23.2 Configuring a virtual directory as an application in IIS.

## 23.4 JScript .NET

- Truly object-oriented language
- Backward compatible with JScript
- Adheres to ECMA 262 Edition 4 standard
- Provides classes, packages, typed variables and access to .NET Framework

## 23.5 A Simple ASP.NET Example

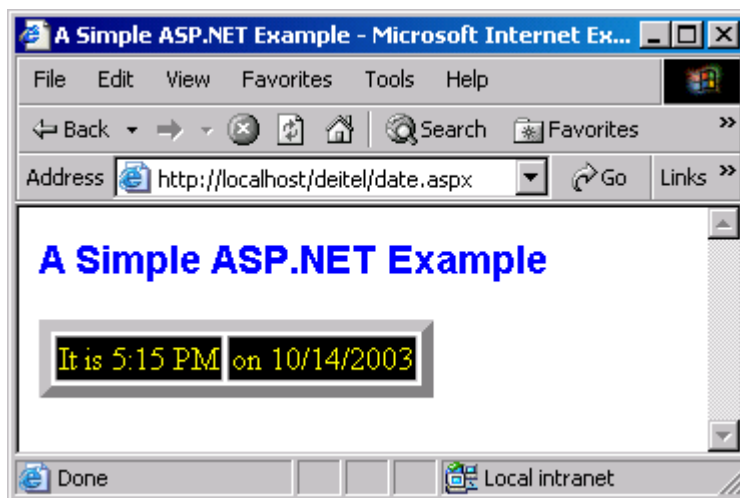
- Scripting delimiters <% and %>
  - Wrapped around Jscript.NET code
  - Compiled and executed on the server
  - @ Page directive
    - Specifies information needed by CLR to process file
  - Language attribute
    - Specifies JScript.NET as scripting language
  - runat attribute with value “server”
    - Indicates script should be processed on server

```
1 <%@ Page Language="JScript" %>
2
3 <!-- Fig. 23.3: date.aspx -->
4 <!-- A simple ASP.NET example -->
5
6 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
7     "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
8
9 <html xmlns = "http://www.w3.org/1999/xhtml">
10
11     <head>
12         <title>A Simple ASP.NET Example</title>
13         <style type = "text/css">
14             td        { background-color: black;
15                         color: yellow }
16             strong    { font-family: arial, sans-serif;
17                         font-size: 14pt; color: blue }
18             p          { font-size: 14pt }
19         </style>
20         <script runat = "server" language = "JScript">
21             var dayAndTime : DateTime = DateTime.Now;
22         </script>
23     </head>
24
```

**date.aspx**  
**(1 of 2)**

```
25 <body>
26 <strong>A Simple ASP.NET Example</strong>
27 <p>
28 <table border = "6">
29 <tr>
30 <td> It is
31 <% Response.Write( dayAndTime.ToShortTimeString() ); %>
32 </td>
33
34 <td> on
35 <% Response.Write( dayAndTime.ToShortDateString() ); %>
36 </td>
37 </tr>
38 </table>
39 </p>
40 </body>
41 </html>
```

**date.aspx**  
**(2 of 2)**



```
1 <!-- Fig. 23.3: date.aspx -->
2 <!-- A Simple ASP.NET example -->
3
4 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
5 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
6
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8
9     <head>
10         <title>A Simple ASP.NET Example</title>
11         <style type = "text/css">
12             td        { background-color: black;
13                       color: yellow }
14             strong    { font-family: arial, sans-serif;
15                       font-size: 14pt; color: blue }
16             p         { font-size: 14pt }
17         </style>
18
19     </head>
20
21     <body>
22         <strong>A Simple ASP.NET Example</strong>
23         <p>
24             <table border = "6">
25                 <tr>
```

**HTML generated by  
date.aspx  
(1 of 2)**

```
26         <td>
27             It is 5:15 PM
28         </td>
29
30         <td>
31             on 10/14/2003
32         </td>
33     </tr>
34 </table>
35 </p>
36 </body>
37 </html>
```

**HTML generated by  
date.aspx  
(2 of 2)**

## 23.5 A Simple ASP.NET Example

Object Name	Description
Request	Used to access information passed by an HTTP request.
Response	Used to control the information sent to the client.
Server	Used to access methods and properties on the server.

Fig. 23.5 Commonly used ASP.NET objects.



## 23.6 Web Forms

- `<form>` tag
  - Designate ASP.NET Web Form
- Web controls
  - HTML server controls
    - Programmable HTML elements run on the server
  - Web server controls
    - Form-like controls such as drop-down lists and text boxes
  - Validation controls (validators)
    - Required field validator
    - Range validator
  - User controls
    - Created by programmer

## 23.6 Web Forms

HTML Server Control	Description
HtmlAnchor	Navigation link.
HtmlButton	Customizable input button.
HtmlTable	Programmatically built table.
HtmlInputFile	Handles uploading of files from client to server.
HtmlImage	Renders images.
HtmlForm	User-input form.

Fig. 23.6 HTML server controls.

Web Server Control	Description
AdRotator	Presents ad images and ad banners.
DataGrid	Displays tabular data and supports selecting, sorting and editing data.
TextBox	Enables user to enter text.
HyperLink	Creates a link to another document.
DropDownList	Provides a single-select drop-down list.
Calendar	Displays a month calendar from which users can select dates.

Fig. 23.7 Web server controls.

## 23.6 Web Forms

Server Control	Description
RequiredFieldValidator	Checks that the user does not leave a field blank.
CompareValidator	Compares an input value with another value. The value being compared to may be another control's input value.
RangeValidator	Checks that a user's entry is within a specified range.
RegularExpressionValidator	Checks that the entry matches a regular expression pattern.
ValidationSummary	Displays the validation errors for all the validation controls on a page.

Fig. 23.8 Validation server controls.

```
1 <%@ Page Language="JScript" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
4 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
5
6 <!-- Fig. 23.9: name.aspx -->
7 <!-- Another Simple ASP.NET example -->
8
9 <html>
10 <head>
11 <title>Name Request</title>
12
13 <script language = "JScript" runat = "server">
14
15     function submitButton_Click(
16         sender : Object, events : EventArgs ) : void
17     {
18         if ( IsPostBack )
19         {
20             if ( iceCream.SelectedItem == "Yes" )
21             {
22                 message.Text = name.Text + " likes ice cream.";
23             }
24             else
25             {
```

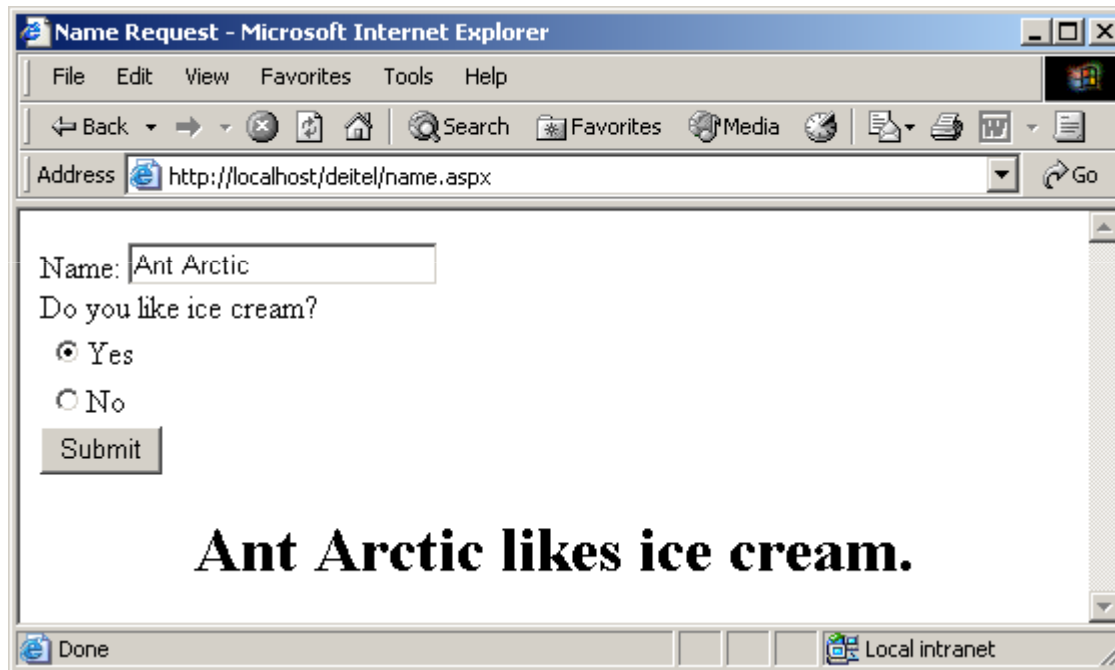
**name.aspx**  
**(1 of 3)**

```
26         message.Text = name.Text + " does not like ice cream.";
27     }
28 }
29
30     } // end submitButton_Click
31 </script>
32 </head>
33
34 <body>
35     <form action = "name.aspx" method = "post" runat = "server">
36
37         Name: <asp:TextBox id = "name" runat = "server"/>
38
39         <br />
40         Do you like ice cream?
41
42         <asp:RadioButtonList id = "iceCream" runat = "server">
43             <asp:ListItem>Yes</asp:ListItem>
44             <asp:ListItem>No</asp:ListItem>
45         </asp:RadioButtonList>
46
47         <asp:Button text = "submit" onClick = "submitButton_Click"
48             runat = "server"/>
49
50         <br />
```

**name.aspx**  
**(2 of 3)**

```
51     <center>
52         <h1> <asp:Label id = "message" runat = "server"/> </h1>
53     </center>
54
55     </form>
56 </body>
57 </html>
```

**name.aspx  
(3 of 3)**



```
1 <%@ Page Language="JScript" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
4   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
5
6 <!-- Fig. 23.10: validation.aspx -->
7 <!-- ASP.NET validation example -->
8
9 <html>
10   <head>
11     <title>Validate Fields</title>
12
13     <script language = "JScript" runat = "server">
14
15         function submitButton_Click(
16             sender : Object , events : EventArgs ) : void
17         {
18             if ( IsPostBack )
19             {
20                 if ( iceCream.SelectedItem == "Yes" )
21                 {
22                     message.Text = name.Text + " likes ice cream.";
23                 }
24                 else
25                 {
```

**validation.aspx**  
**(1 of 4)**

```
26         message.Text = name.Text + " does not like ice cream.";
27     }
28 }
29
30     } // end submitButton_Click
31 </script>
32 </head>
33
34 <body>
35     <form action = "validation.aspx" method = "post" runat = "server">
36
37         <table>
38             <tr>
39                 <td>
40                     Name: <asp:textbox id = "name" runat = "server"/>
41                 </td>
42                 <td>
43                     <asp:RequiredFieldValidator id = "required"
44                         ControlToValidate = "name"
45                         Display = "Static"
46                         runat = "server">
47                         Please enter your name.
48                     </asp:RequiredFieldValidator>
49                 </td>
50             </tr>
```

**validation.aspx**  
**(2 of 4)**



```
51     </table>
52
53     <br />
54     Do you like ice cream?
55
56     <asp:RadioButtonList id = "iceCream" runat = "server">
57         <asp:ListItem>Yes</asp:ListItem>
58         <asp:ListItem>No</asp:ListItem>
59     </asp:RadioButtonList>
60
61     <br />
62     How many scoops would you like? (0-45)
63
64     <asp:TextBox id = "scoops" runat = "server" />
65
66     <br />
67     <asp:button text = "Submit" onClick = "submitButton_Click"
68         runat = "server"/>
69
70     <asp:RangeValidator
71         ControlToValidate = "scoops"
72         MinimumValue = "0"
73         MaximumValue = "45"
74         Type = "Integer"
75         EnableClientScript = "false"
```

**validation.aspx**  
**(3 of 4)**

```
76         Text = "We cannot give you that many scoops."  
77         runat = "server" />  
78  
79     <center>  
80         <h1> <asp:label id = "message" runat = "server"/> </h1>  
81     </center>  
82  
83 </form>  
84 </body>  
85 </html>
```

**validation.aspx**  
**(4 of 4)**

## 23.6 Web Forms

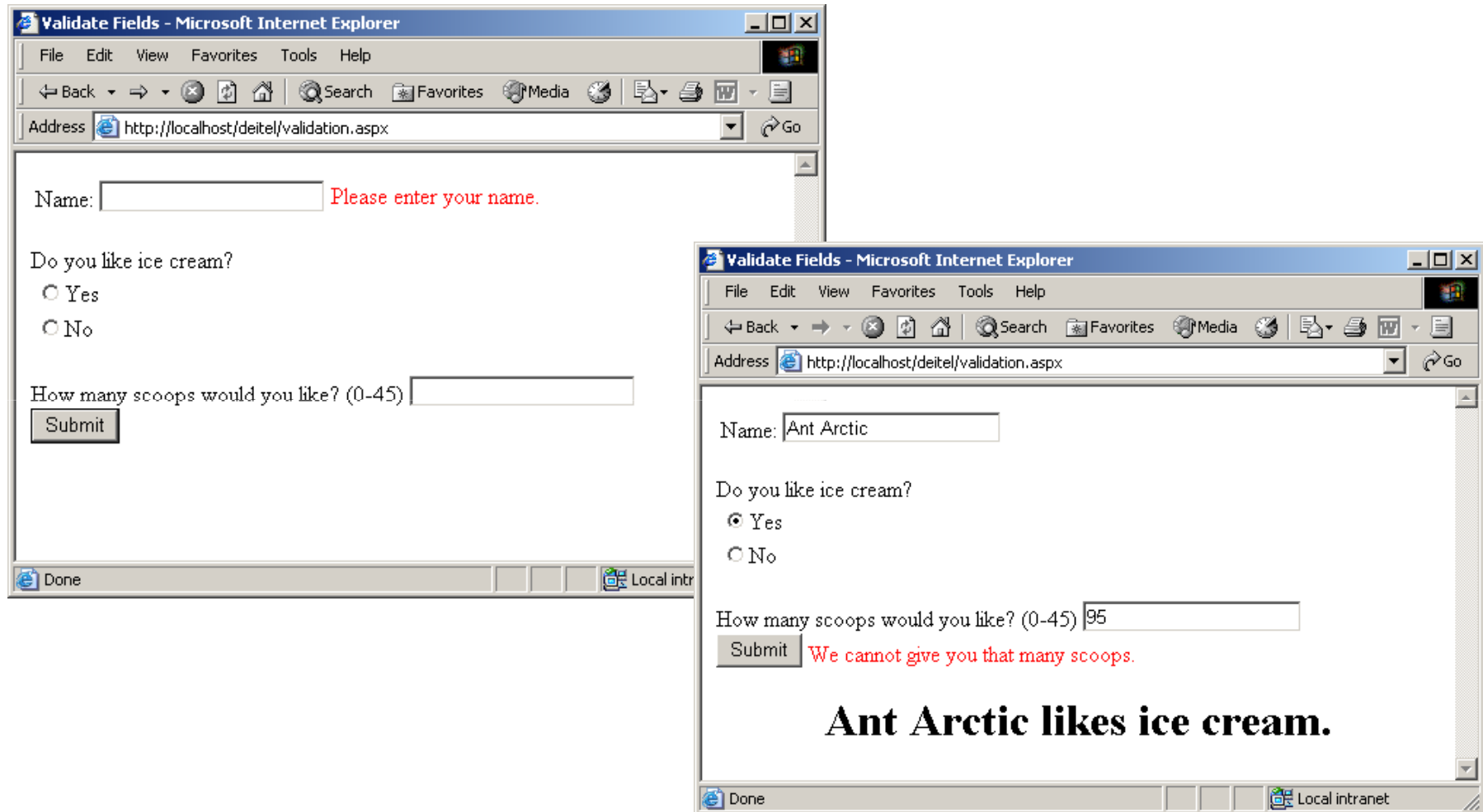


Fig. 23.11 Validation error output.

## 23.6 Web Forms

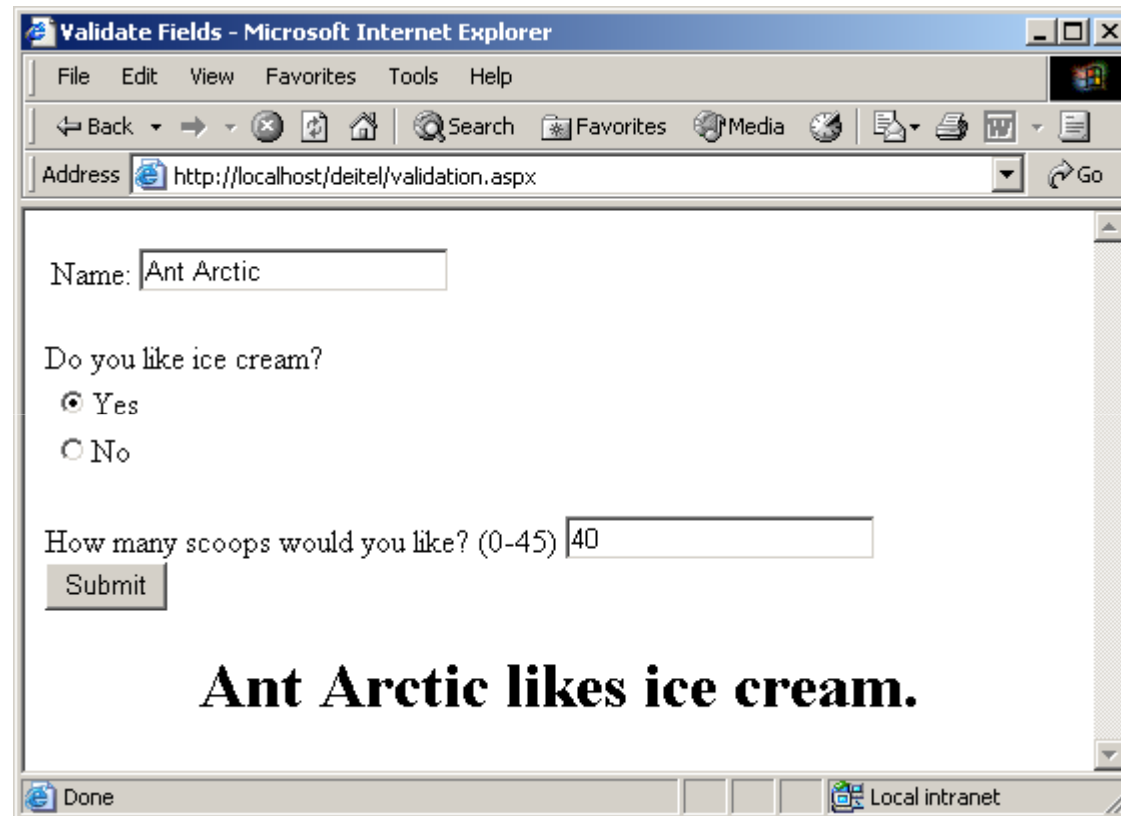


Fig. 23.12 Valid page without validation errors.

```
1 <%@ Page Language="JScript" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
4   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
5
6 <!-- Fig. 23.13: adRotator.aspx -->
7 <!-- ASP.NET AdRotator example -->
8
9 <html>
10   <head>
11     <title>Using An AdRotator</title>
12
13     <script language = "JScript" runat = "server">
14
15         function submitButton_Click(
16             sender : Object, events : EventArgs ) : void
17         {
18             if ( IsPostBack )
19             {
20                 if ( iceCream.SelectedItem == "Yes" )
21                 {
22                     message.Text = name.Text + " likes ice cream.";
23                 }
24                 else
25                 {
```

**adRotator.aspx**  
**(1 of 4)**

```
26         message.Text = name.Text + " does not like ice cream.";
27     }
28 }
29
30     } // end submitButton_Click
31 </script>
32 </head>
33
34 <body>
35     <form action = "adRotator.aspx" method = "post" runat = "server">
36
37         <asp:AdRotator AdvertisementFile = "ads.xml"
38             BorderColor = "black" Borderwidth = "1" runat = "server"/>
39
40         <table>
41             <tr>
42                 <td>
43                     Name: <asp:textbox id = "name" runat = "server"/>
44                 </td>
45                 <td>
46                     <asp:RequiredFieldValidator id = "required"
47                         ControlToValidate = "name"
48                         Display = "Static"
49                         runat = "server">
50                         Please enter your name.
```

**adRotator.aspx**  
**(2 of 4)**

```
51         </asp:RequiredFieldValidator>
52     </td>
53 </tr>
54 </table>
55
56 <br />
57 Do you like ice cream?
58
59 <asp:RadioButtonList id = "iceCream" runat = "server">
60     <asp:ListItem>Yes</asp:ListItem>
61     <asp:ListItem>No</asp:ListItem>
62 </asp:RadioButtonList>
63
64 <br />
65 How many scoops would you like? (0-45)
66
67 <asp:TextBox id = "scoops" runat = "server" />
68
69 <br />
70 <asp:button text = "Submit" onClick = "submitButton_Click"
71     runat = "server"/>
72
73 <asp:RangeValidator
74     ControlToValidate = "scoops"
75     MinimumValue = "0"
```

**adRotator.aspx**  
**(3 of 4)**

```
76         MaximumValue = "45"
77         Type = "Integer"
78         EnableClientScript = "false"
79         Text = "We cannot give you that many scoops."
80         runat = "server" />
81
82     <center>
83         <h1> <asp:label id = "message" runat = "server"/> </h1>
84     </center>
85
86 </form>
87 </body>
88 </html>
```

**adRotator.aspx**  
**(4 of 4)**



```
1 <?xml version = "1.0" ?>
2
3 <!-- Fig. 23.14: ads.xml -->
4 <!-- Flag database -->
5
6 <Advertisements>
7
8   <Ad>
9     <ImageUrl>images/unitedstates.png</ImageUrl>
10    <NavigateUrl>http://www.usa.worldweb.com/</NavigateUrl>
11    <AlternateText>US Tourism</AlternateText>
12    <Impressions>80</Impressions>
13  </Ad>
14
15  <Ad>
16    <ImageUrl>images/germany.png</ImageUrl>
17    <NavigateUrl>http://www.germany-tourism.de/</NavigateUrl>
18    <AlternateText>German Tourism</AlternateText>
19    <Impressions>80</Impressions>
20  </Ad>
21
22  <Ad>
23    <ImageUrl>images/spain.png</ImageUrl>
24    <NavigateUrl>http://www.tourspain.es/</NavigateUrl>
25    <AlternateText>Spanish Tourism</AlternateText>
```

ads.xml  
(1 of 2)

```
26     <Impressions>80</Impressions>
27   </Ad>
28
29 </Advertisements>
```

**ads.xml**  
**(2 of 2)**

## 23.6 Web Forms

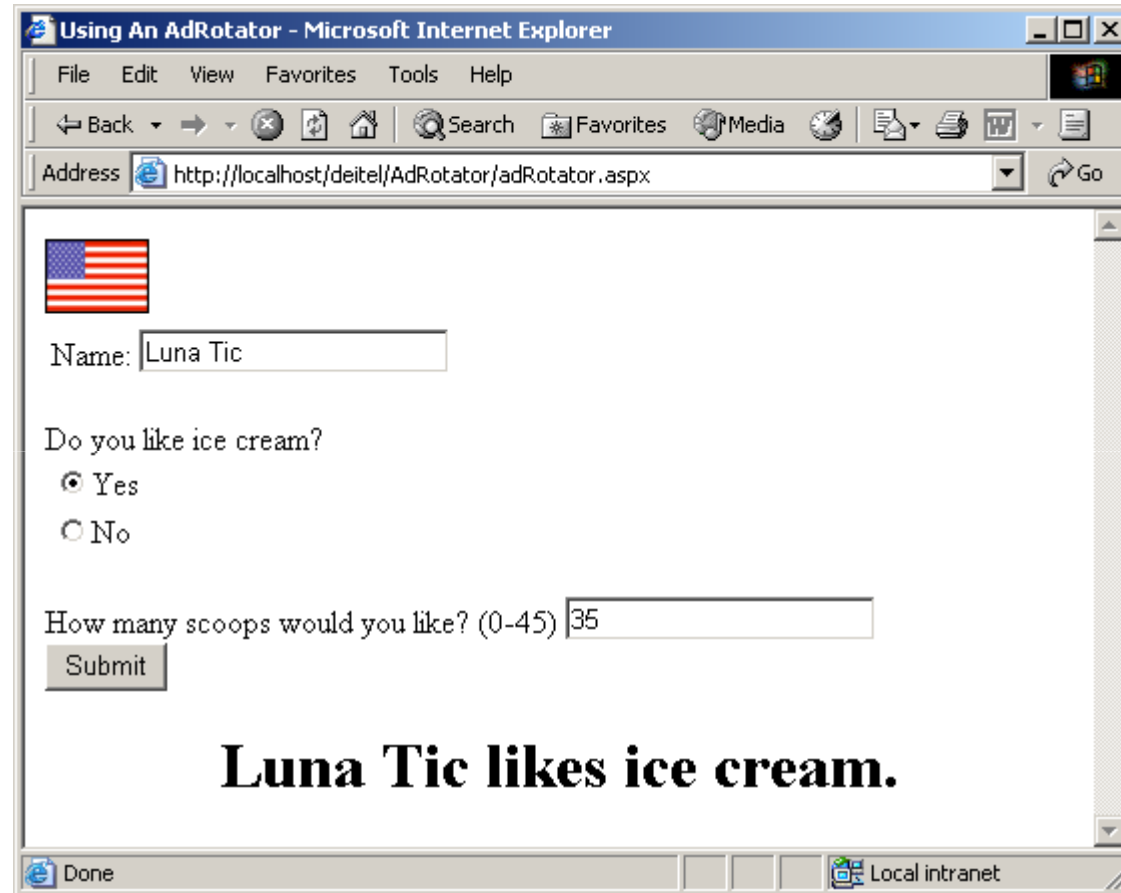


Fig. 23.15 ASPX page with an AdRotator.

## 23.7 Session Tracking

- Personalization
- Protection of privacy
- Cookies
- .NET's `HttpSessionState` object
- Use of input form elements of type “hidden”
- URL rewriting

## 23.7.1 Cookies

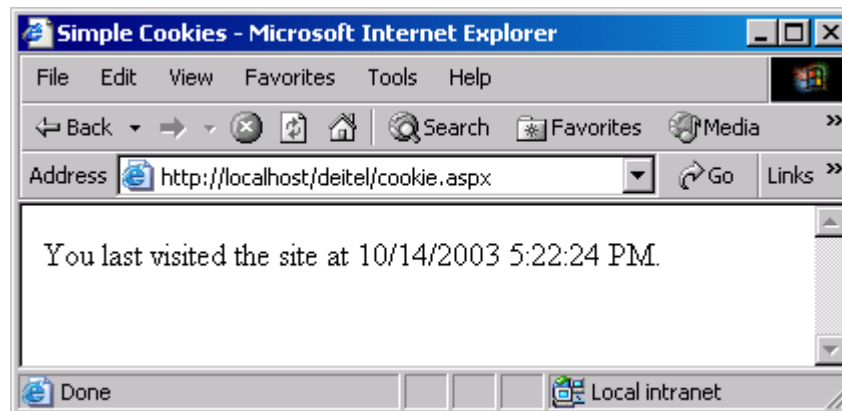
- Customize interactions with Web pages
- Stored by a Web site on an individual's computer
- Reactivated each time the user revisits site

```
1 <%@ Page Language="JScript" Debug="true" %>
2
3 <!-- Fig. 23.16: cookie.aspx -->
4 <!-- Records last visit -->
5
6 <html>
7   <head>
8     <title> Simple Cookies </title>
9
10    <script runat = "server">
11
12      function Page_Load( object : Object, events : EventArgs )
13      {
14        var lastVisit : String;
15
16        if ( Request.Cookies( "visit" ) == null )
17        {
18          welcome.Text = "This is the first time that " +
19            "you have visited this site today";
20        }
21        else
22        {
23          lastVisit = Request.Cookies( "visit" ).Value;
24          welcome.Text = "You last visited the site at " +
25            lastVisit + ".";
```

cookie.aspx  
(1 of 2)

```
26     }
27
28     var time : DateTime = DateTime.Now;
29     Response.Cookies( "visit" ).Value = time.ToString();
30     Response.Cookies( "visit" ).Expires = time.AddDays( 1 );
31
32     } // end Page_Load
33 </script>
34 </head>
35 <body>
36     <form runat = "server">
37         <asp:label id = "welcome" runat = "server"/>
38     </form>
39 </body>
40 </html>
```

**cookie.aspx**  
**(2 of 2)**



## 23.7.1 Cookies

Property	Description
Domain	Returns a <b>String</b> containing the cookie's domain (i.e., the domain of the Web server from which the cookie was downloaded). This determines which Web servers can receive the cookie. By default, cookies are sent to the Web server that originally sent them to the client.
Expires	Returns a <b>DateTime</b> object indicating when the browser can delete the cookie.
Name	Returns a <b>String</b> containing the cookie's name.
Path	Returns a <b>String</b> containing the URL prefix for the cookie. Cookies can be "targeted" to specific URLs that include directories on the Web server, enabling the programmer to specify the location of the cookie. By default, a cookie is returned to services operating in the same directory as the service that sent the cookie or a subdirectory of that directory.
Secure	Returns a <b>Boolean</b> value indicating whether the cookie should be transmitted through a secure protocol. The value <b>True</b> causes a secure protocol to be used.
Value	Returns a <b>String</b> containing the cookie's value.
Fig. 23.17	<b>HttpCookie</b> properties.



## 23.7.2 Session Tracking with HttpSessionState

Property	Description
Count	Specifies the number of key-value pairs in the <code>Session</code> object.
IsNewSession	Indicates whether this is a new session (i.e., whether the session was created during loading of this page).
IsReadOnly	Indicates whether the <code>Session</code> object is read only.
Keys	Returns an object containing the <code>Session</code> object's keys.
SessionID	Returns the session's unique ID.
Timeout	Specifies the maximum number of minutes during which a session can be inactive (i.e., no requests are made) before the session expires. By default, this property is set to 20 minutes.
Fig. 23.18	<code>HttpSessionState</code> properties.

```

1  <%@ Page Language="JScript" %>
2  <%@ Import Namespace="System" %>
3
4  <!-- Fig. 23.19: optionsPage.aspx          --%>
5  <!-- Page that presents a list of language options. --%>
6
7  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
8     "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
9
10 <html>
11   <head>
12     <title>Options Page</title>
13
14     <script runat = "server">
15
16         // event handler for Load event
17         var books : Hashtable = new Hashtable();
18
19         function Page_Load( sender : Object, events : EventArgs ) : void
20         {
21             // if page is loaded due to postback, load session
22             // information, hide language options from user
23             books.Add( "C#", "0-13-062221-4" );
24             books.Add( "C++", "0-13-089571-7" );
25             books.Add( "C", "0-13-089572-5" );

```

optionsPage.aspx  
(1 of 6)

```
26     books.Add( "Python", "0-13-092361-3" );
27
28     if ( IsPostBack )
29     {
30         // display components that contain
31         // session information
32         welcomeLabel.Visible = true;
33         languageLink.Visible = true;
34         recommendationsLink.Visible = true;
35
36         // hide components
37         submitButton.Visible = false;
38         promptLabel.Visible = false;
39         languageList.Visible = false;
40
41         // set labels to display session information
42         if ( languageList.SelectedItem != null )
43         {
44             welcomeLabel.Text +=
45                 languageList.SelectedItem.ToString() + ".";
46         }
47         else
48         {
49             welcomeLabel.Text += "no language.";
50         }

```

optionsPage.aspx  
(2 of 6)

```
51         idLabel.Text += "Your unique session ID is: " +
52             Session.SessionID;
53
54
55         timeoutLabel.Text += "Timeout: " +
56             Session.Timeout + " minutes";
57     } // end if
58 } // end Page_Load
59
60 // when user clicks submit button,
61 // store user's choice in session object
62 function submitButton_Click (
63     sender : Object, events : EventArgs ) : void
64 {
65     if ( languageList.SelectedItem != null )
66     {
67         var language : String =
68             languageList.SelectedItem.ToString();
69
70         // note: must use ToString method because the hash table
71         // stores information as objects
72         var ISBN : String = books[ language ].ToString();
73
74         // store in session as name-value pair
75         // name is language chosen, value is
```

optionsPage.aspx  
(3 of 6)

```
76         // ISBN number for corresponding book
77         Session.Add( language, ISBN );
78     } // end if
79 } // end submitButton_Click
80
81 </script>
82 </head>
83 <body>
84     <form id = "recommendationsPage" method = "post" runat = "server">
85         <P>
86             <asp:Label id = "promptLabel" runat = "server"
87                 Font-Bold = "True">Select a programming language:
88             </asp:Label>
89             <asp:Label id = "welcomeLabel" runat = "server"
90                 Font-Bold = "True" Visible = "False">
91                 welcome to Sessions! You selected
92             </asp:Label>
93         </P>
94         <P>
95             <asp:RadioButtonList id = "languageList" runat = "server">
96                 <asp:ListItem value = "C#">C#</asp:ListItem>
97                 <asp:ListItem value = "C++">C++</asp:ListItem>
98                 <asp:ListItem value = "C">C</asp:ListItem>
99                 <asp:ListItem value = "Python">Python</asp:ListItem>
100            </asp:RadioButtonList></P>
```

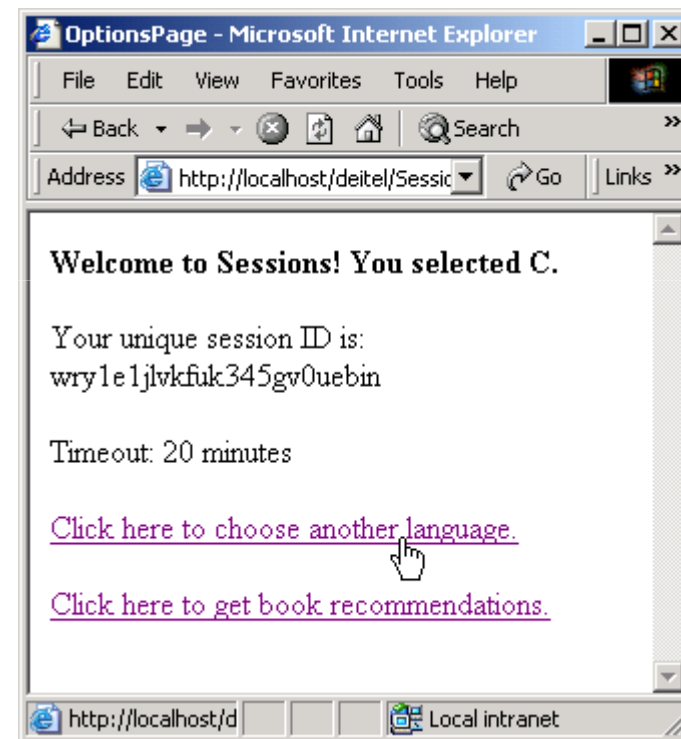
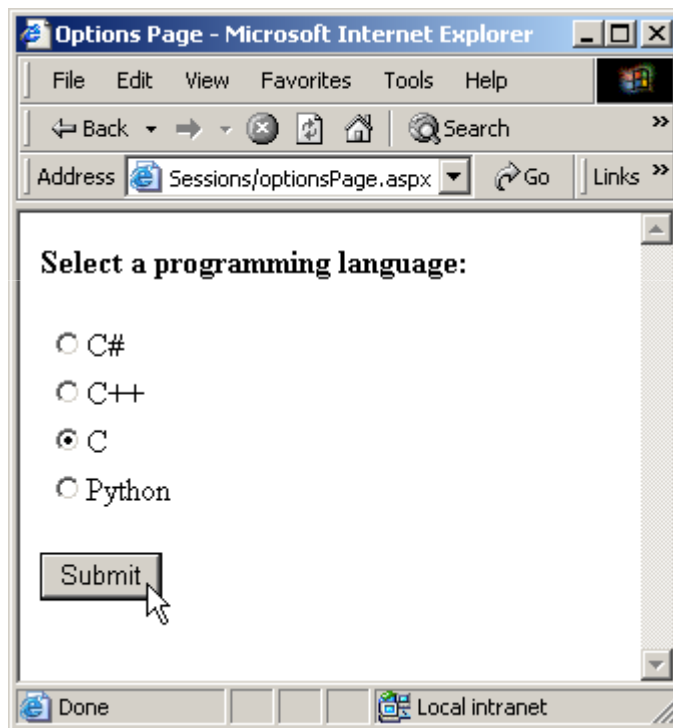
**optionsPage.aspx  
(4 of 6)**

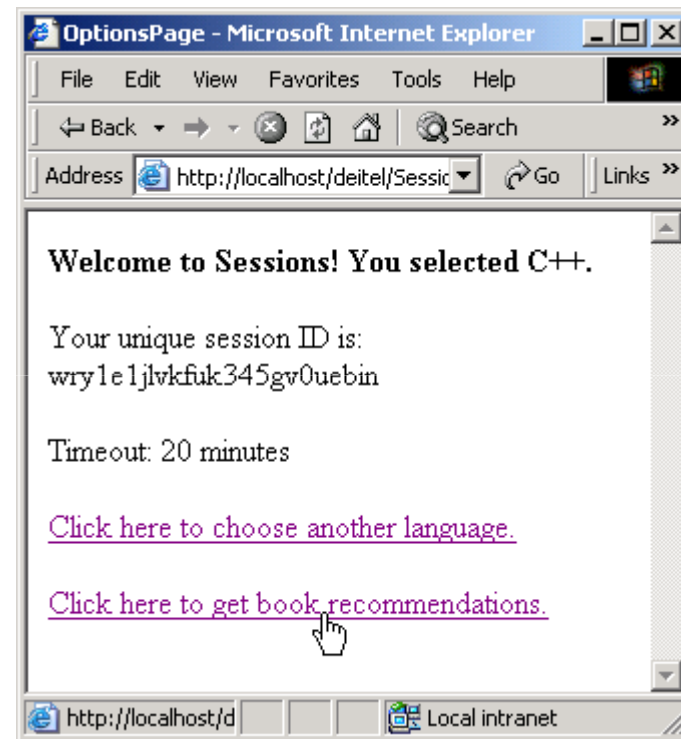
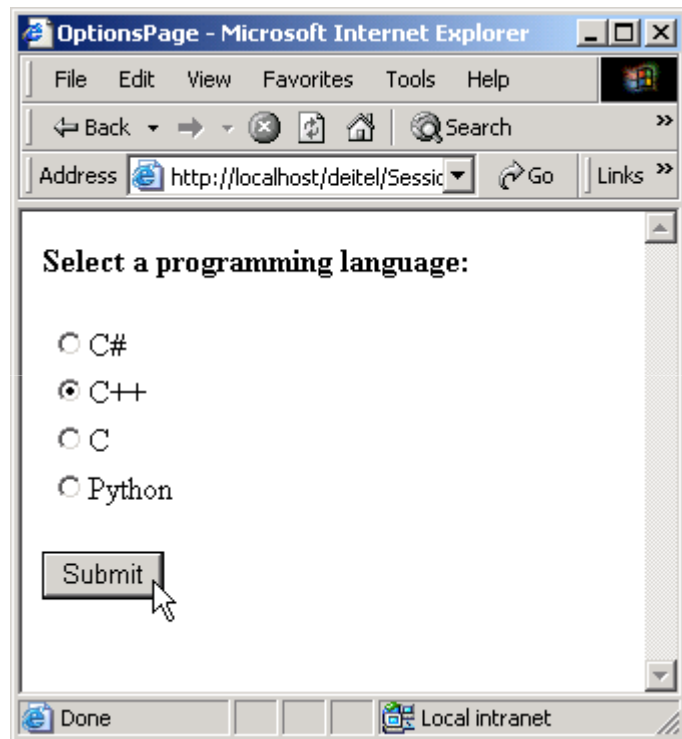
```
101     <P>
102         <asp:Button id = "submitButton" runat = "server"
103             Text = "Submit" onClick = "submitButton_Click">
104     </asp:Button>
105 </P>
106 <P>
107     <asp:Label id = "idLabel" runat = "server">
108 </asp:Label>
109 </P>
110 <P>
111     <asp:Label id = "timeoutLabel" runat = "server">
112 </asp:Label>
113 </P>
114 <P>
115     <asp:Label id = "newSessionLabel" runat = "server">
116 </asp:Label>
117 </P>
118 <P>
119     <asp:HyperLink id = "languageLink" runat = "server"
120         NavigateUrl = "optionsPage.aspx" visible = "False">
121         Click here to choose another language.
122     </asp:HyperLink>
123 </P>
124 <P>
125     <asp:HyperLink id = "recommendationsLink" runat = "server">
```

**optionsPage.aspx  
(5 of 6)**

```
126         NavigateUrl = "recommendationsPage.aspx"
127         visible = "False">
128         Click here to get book recommendations.
129     </asp:HyperLink>
130 </P>
131 </form>
132 </body>
133 </html>
```

**optionsPage.aspx  
(6 of 6)**







```

1  <%@ Page Language="JScript" %>
2
3  <!-- Fig. 23.20: recommendationsPage.aspx --%>
4  <!-- Read the users session data.          --%>
5
6  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
7     "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
8
9  <html>
10     <head>
11         <title>Recommendations Page</title>
12
13         <script runat = "server">
14
15             protected function OnInit( events : EventArgs ) : void
16             {
17                 // determine if session contains information
18                 if ( Session != null )
19                 {
20                     // iterate through session values,
21                     // display in ListBox
22                     for ( var i : int = 0; i < Session.Count; i++ )
23                     {
24                         // store current key in sessionName
25                         var keyName : String = Session.Keys[ i ];

```

recommendationsPage  
.aspx  
(1 of 3)

```

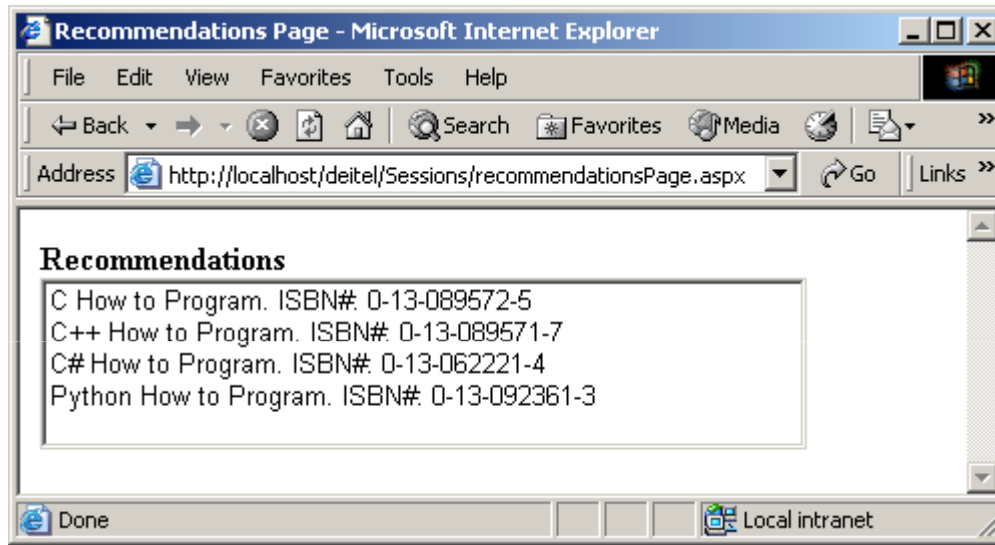
26
27         // use current key to display
28         // Session's name/value pairs
29         booksListBox.Items.Add( keyName +
30         " How to Program. ISBN#: " + Session[ keyName ] );
31     } // end for
32 }
33 else
34 {
35     recommendationsLabel.Text = "No Recommendations";
36     booksListBox.Visible = false;
37 }
38 } // end OnInit
39
40 </script>
41 </head>
42 <body>
43     <form id = "Form1" method = "post" runat = "server">
44         <asp:Label id = "recommendationsLabel"
45             runat = "server" Font-Bold = "True">
46             Recommendations
47         </asp:Label>
48         <br />
49         <asp:ListBox id = "booksListBox" runat = "server"
50             width = "383px" height = "91px">

```

recommendationsPage  
.aspx  
(2 of 3)

```
51     </asp:ListBox>
52     </form>
53 </body>
54 </html>
```

**recommendationsPage  
.aspx  
(3 of 3)**



## 23.8 ASP.NET and XML

- ASP.NET provides classes and built-in functionality to manipulate XML files
- Guest book stores posts in XML and uses an XSL output posts

```
1 <?xml version = "1.0" ?>
2
3 <!-- Fig. 23.21: posts.xml -->
4 <!-- Guest book posts      ->
5
6 <?xml-stylesheet type = "text/xsl" href = "formatting.xsl"?>
7
8 <guestbook>
9     <post timestamp = "8/1/2003 9:41:49 AM">
10         <name>anter</name>
11         <email>ic@deitel.com</email>
12         <text>Hello! How are you today? I am fine! </text>
13     </post>
14     <post timestamp = "8/1/2003 9:40:40 AM">
15         <name>Pro Gram Fly</name>
16         <email>PGF@deitel.com</email>
17         <text>XML and ASP.NET! what a combination!</text>
18     </post>
19 </guestbook>
```

**posts.xml**  
**(1 of 1)**

```

1  <%@ Page Language="JScript" Debug="true" %>
2  <%@ Import Namespace="System.Data" %>
3  <%@ Import Namespace="System.Xml" %>
4
5  <!-- Fig. 23.22: guestbook.aspx          -->
6  <!-- Web Form for guest book application -->
7
8  <html>
9      <head>
10         <title>Guest Book</title>
11
12         <link rel = "stylesheet" type = "text/css"
13             href = "style.css" />
14
15         <script language = "JScript" runat = "server">
16
17             function guestBookPost_Click(
18                 sender : Object, events : EventArgs ) : void
19             {
20                 // Open an XML document.
21                 var myDocument : XmlDocument = new XmlDocument();
22                 myDocument.Load( Server.MapPath( "posts.xml" ) );
23

```

guestbook.aspx  
(1 of 4)

```
24 // Create XML element that will represent the post
25 var postNode : XElement =
26     myDocument.CreateElement( "post" );
27
28     postNode.SetAttribute( "timestamp", DateTime.Now.ToString() );
29
30     var nameNode : XElement =
31         myDocument.CreateElement( "name" );
32     var emailNode : XElement =
33         myDocument.CreateElement( "email" );
34     var messageNode : XElement =
35         myDocument.CreateElement( "text" );
36
37     nameNode.AppendChild(
38         myDocument.CreateTextNode( name.Text ) );
39     emailNode.AppendChild(
40         myDocument.CreateTextNode( email.Text ) );
41     messageNode.AppendChild(
42         myDocument.CreateTextNode( message.Text ) );
43
44     postNode.AppendChild( nameNode );
45     postNode.AppendChild( emailNode );
46     postNode.AppendChild( messageNode );
47
```

guestbook.aspx  
(2 of 4)

```

48         // Insert the new element into the XML tree and save
49         myDocument.DocumentElement.PrependChild( postNode );
50         myDocument.Save( Server.MapPath( "posts.xml" ) );
51
52         name.Text = "";
53         email.Text = "";
54         message.Text = "";
55
56         Response.Redirect ( "posts.xml" );
57     } // end guestBookPost_Click
58
59     </script>
60 </head>
61 <body>
62
63     <p>My Guest Book</p>
64
65     <hr />
66     <form Runat = "server">
67         <table>
68             <tr><td>Name:</td><td> <asp:TextBox ID = "name"
69                 Columns = "50" Runat = "server"/></td></tr>
70             <tr><td>E-mail:</td><td><asp:TextBox ID = "email"
71                 Columns = "50" Runat = "server"/></td></tr>

```

guestbook.aspx  
(3 of 4)



```
72     </table>
73     Message: <asp:TextBox ID = "message" TextMode = "MultiLine"
74         Columns = "50" Rows = "4" Runat = "server"/><br />
75
76     <asp:LinkButton ID = "guestBookPost"
77         OnClick = "guestBookPost_Click" Text = "Post"
78         Runat = "server" ForeColor = "Green" Font-Size = "20" />
79
80     <br />
81
82     <asp:HyperLink ID = "link" NavigateUrl = "posts.xml"
83         Text = "view" Runat = "server" ForeColor = "Green"
84         Font-Size = "20" />
85 </form>
86 </body>
87 </html>
```

**guestbook.aspx**  
**(4 of 4)**

```
1 <?xml version = "1.0" ?>
2
3 <!-- Fig. 23.23: formatting.xml -->
4 <!-- XSL document that transforms XML data to HTML -->
5
6 <xsl:stylesheet version = "1.0"
7   xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
8
9   <xsl:output method = "html" omit-xml-declaration = "no"
10     doctype-system =
11       "http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd"
12     doctype-public = "-//W3C//DTD XHTML 1.1//EN" />
13
14   <xsl:template match = "/">
15
16     <html xmlns = "http://www.w3.org/1999/xhtml">
17       <xsl:apply-templates select = "*" />
18     </html>
19
20   </xsl:template>
21
22   <xsl:template match = "guestbook">
23
24     <head>
25       <title><xsl:value-of select = "name"/></title>
```

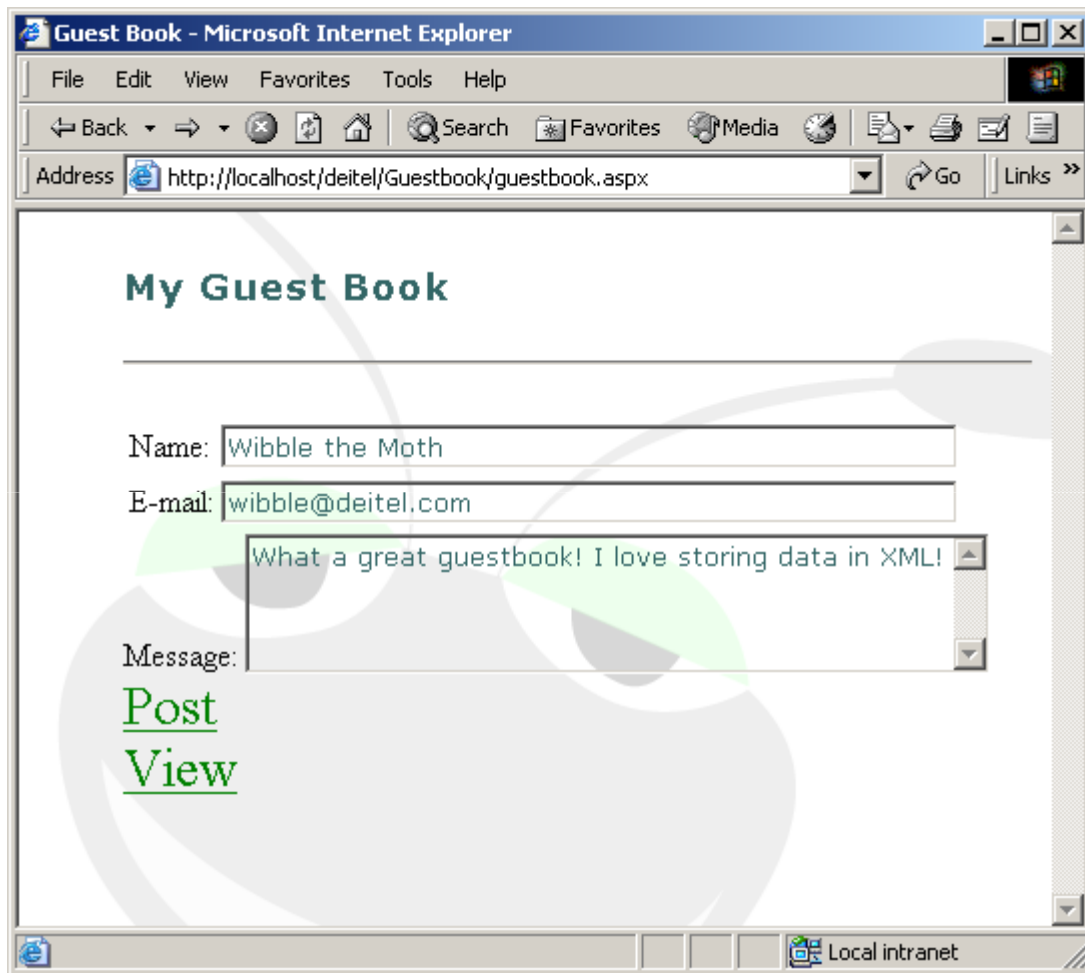
**formatting.xml**  
**(1 of 3)**

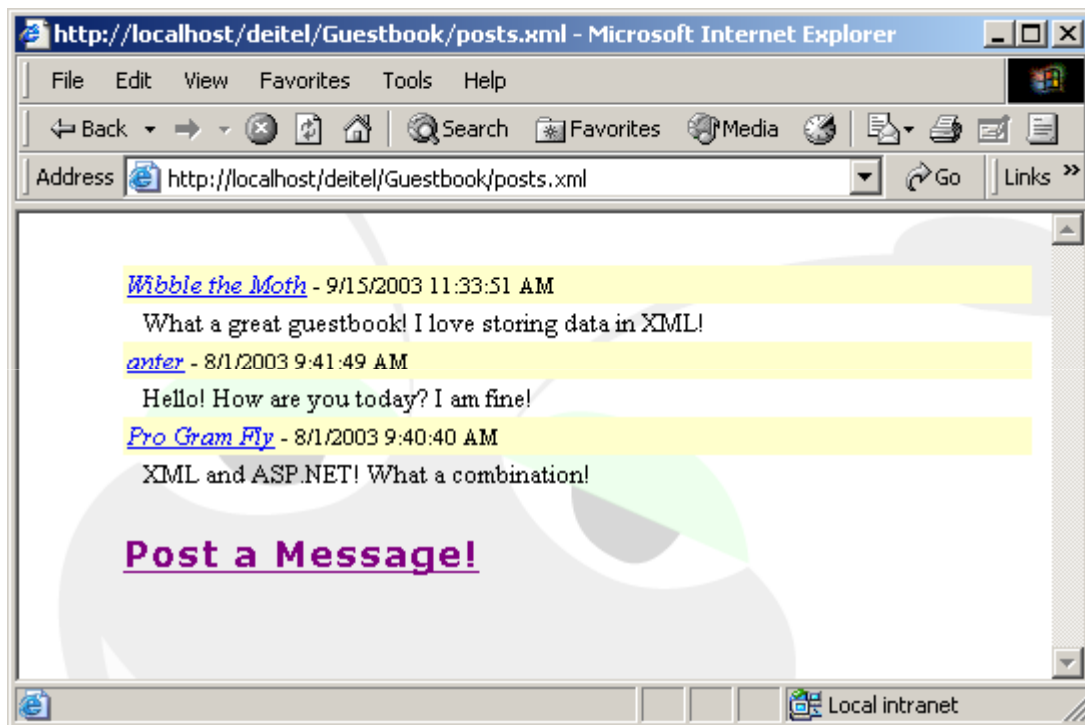
```
26     <link rel = "stylesheet" type = "text/css"
27         href = "style.css" />
28 </head>
29
30 <body>
31
32     <table width = "100%" cellspacing = "0"
33         cellpadding = "2">
34         <xsl:apply-templates
35             select = "post" />
36     </table>
37
38     <p>
39         <a href = "guestbook.aspx">Post a Message!</a>
40     </p>
41
42 </body>
43
44 </xsl:template>
45
46 <xsl:template match = "post">
47
48     <tr>
49         <td class = "msgInfo">
```

**formatting.xml  
(2 of 3)**

```
51
52     <em>   <a href = "mailto:{email}">
53     <xsl:value-of select = "name" /> </a> </em>
54     -
55
56     <span class = "date">
57         <xsl:value-of select = "@timestamp" />
58     </span>
59 </td>
60 </tr>
61
62 <tr>
63     <td class = "msgText">
64         <xsl:value-of select = "text" />
65     </td>
66 </tr>
67
68 </xsl:template>
69
70 </xsl:stylesheet>
```

**formatting.xml**  
**(3 of 3)**





## 23.9 Reading and Writing Text Files

- `System.IO` namespace
  - Reading and writing to files and streams
  - `FileInfo` and `Directory`
    - Methods and properties to retrieve information and perform basic operations on files and directories
  - `FileStream`, `StreamReader` and `StreamWriter`
    - Read and write content to and from a file

## 23.9 Reading and Writing Text Files

Methods/Properties	Description
<i>Methods</i>	
AppendText	Creates a stream object for appending to current file.
CopyTo	Copies a file.
Create	Creates a file.
Delete	Deletes a file.
Open	Opens a file.
ToString	Returns a string that represents the full path of a file.
<i>Properties</i>	
CreationTime	The creation time of the current <code>FileInfo</code> object.
Directory	An instance of the parent directory.
extension	A string that represents the extension of the file.
Exists	Boolean indicating whether the file exists.
Length	The size of the current file.
Fig. 23.25	<b>FileInfo</b> class methods and properties.



## 23.9 Reading and Writing Text Files

Methods/Properties	Description
<i>Methods</i>	
<code>Exists</code>	Determines whether the specified path refers to an existing directory on disk.
<code>GetLastAccessTime</code>	Returns the date and time the specified file or directory was last accessed.
<code>GetLastWriteTime</code>	Returns the last date and time when the specified file or directory was written.
<code>GetLogicalDrives</code>	Retrieves the names of the drives on the computer.
<code>GetParent</code>	Retrieves the parent directory of the specified path.
Fig. 23.26	<b>Directory</b> class methods and properties.

## 23.9 Reading and Writing Text Files

Methods/Properties	Description
<i>Methods</i>	
<code>Close</code>	Closes the <code>StreamReader</code> and the underlying stream, then releases any system resources associated with the reader.
<code>Peek</code>	Returns the next available character but does not consume it.
<code>Read</code>	Reads the next character or next set of characters from the input stream.
<code>ReadLine</code>	Reads a line of characters from the current stream and returns the data as a string.
<i>Properties</i>	
<code>BaseStream</code>	The underlying stream.
<code>CurrentEncoding</code>	The current character encoding of the current stream.
Fig. 23.27	<b>StreamReader</b> class methods and properties.

## 23.9 Reading and Writing Text Files

Methods/Properties	Description
<i>Methods</i>	
<code>Close</code>	Closes the current <code>StreamWriter</code> and any underlying stream.
<code>Flush</code>	Clears all buffers for the current writer.
<code>Write</code>	Writes data to the stream.
<code>WriteLine</code>	Writes data to the stream data followed by a line terminator.
<i>Properties</i>	
<code>AutoFlush</code>	Gets or sets a value indicating whether the <code>StreamWriter</code> will flush its buffer after every <code>Write</code> call.
<code>Encoding</code>	Gets the encoding in which the output is written.
Fig. 23.28	<b>StreamWriter</b> class methods and properties.

```

1  <%@ Page Language="JScript" %>
2  <%@ Import Namespace="System" %>
3  <%@ Import Namespace="System.Data" %>
4  <%@ Import Namespace="System.Web.UI.WebControls" %>
5  <%@ Import Namespace="System.IO" %>
6
7  <%
8      // Fig. 23.29: mailinglist.aspx
9      // A Web Form mailing list.
10 %>
11
12 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
13     "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
14
15 <html>
16     <head>
17         <title>Join our Mailing List</title>
18         <script language = "JScript" runat = "server">
19
20             var dataView : DataView;
21
22             function Page_Load (
23                 sender : Object, events : EventArgs ) : void
24             {
25                 dataView = new DataView ( new DataTable() );

```

mailinglist.aspx  
(1 of 8)

```
26     }
27
28     function clearButton_Click(
29         sender : Object, events : System.EventArgs ) : void
30     {
31         nameTextBox.Text = "";
32         emailTextBox.Text = "";
33         cityTextBox.Text = "";
34         stateTextBox.Text = "";
35         addressTextBox.Text = "";
36     }
37
38     function FillMessageTable() : void
39     {
40         var table : DataTable = dataView.Table;
41         table.Columns.Add( "Name" );
42         table.Columns.Add( "Address" );
43         table.Columns.Add( "City" );
44         table.Columns.Add( "State" );
45         table.Columns.Add( "E-mail" );
46
47         // open mailing list for reading
48         var reader : StreamReader = new StreamReader(
49             Request.PhysicalApplicationPath + "/Mailing/" +
50             "mailinglist.txt" );
```

mailinglist.aspx  
(2 of 8)

```
51
52     var separator : char[] = [ '\t' ];
53
54     // read in line from file
55     var message : String = reader.ReadLine();
56
57     while ( message != null ) {
58         // split the string into its four parts
59         var parts : String[] = message.Split( separator );
60
61         // load data into table
62         table.LoadDataRow( parts, true );
63
64         // read in one line from file
65         message = reader.ReadLine();
66     } // end while
67
68     // update grid
69     dataGrid.DataSource = table;
70     dataGrid.DataBind();
71
72     reader.Close();
73 } // end FillMessageTable
74
```

mailinglist.aspx  
(3 of 8)

```

75     function submitButton_Click (
76         sender : Object, events: System.EventArgs ): void
77     {
78         // open stream for appending to file
79         var list : StreamWriter =
80             new StreamWriter( Request.PhysicalApplicationPath
81                 + "/Mailing/" + "mailinglist.txt", true );
82
83         // write new address to file
84         list.WriteLine(
85             nameTextBox.Text + "\t" + addressTextBox.Text + "\t"
86             + cityTextBox.Text + "\t" + stateTextBox.Text + "\t" +
87             emailTextBox.Text );
88
89         // clear textboxes and close stream
90         nameTextBox.Text = "";
91         emailTextBox.Text = "";
92         cityTextBox.Text = "";
93         stateTextBox.Text = "";
94         addressTextBox.Text = "";
95         list.Close();
96
97         FillMessageTable();
98     } // end submitButton_Click
99

```

mailinglist.aspx  
(4 of 8)

```
100     </script>
101 </head>
102 <body>
103     <form id = "Form1" runat = "server">
104         <asp:Label id = "promptLabel" runat = "server"
105             ForeColor = "Blue" Font-Size = "X-Large">
106             Fill in the fields below to join our mailing list:
107         </asp:Label>
108         <br />
109
110         <asp:Label id = "nameLabel" runat = "server"
111             style = "position: absolute; left:107px">
112             Name:</asp:Label>
113         <asp:TextBox id = "nameTextBox" runat = "server"
114             style = "position: absolute; left:150px">
115         </asp:TextBox>
116         <br />
117
118         <asp:Label id = "addressLabel" runat = "server"
119             style = "position: absolute; left:93px">
120             Address:</asp:Label>
121         <asp:TextBox id = "addressTextBox" runat = "server"
122             style = "position: absolute; left:150px">
123         </asp:TextBox>
124         <br />
```

**mailinglist.aspx**  
**(5 of 8)**



```
125
126     <asp:Label id = "cityLabel" runat = "server"
127         style = "position: absolute; left:119px">
128         City:</asp:Label>
129     <asp:TextBox id = "cityTextBox" runat = "server"
130         style = "position: absolute; left:150px">
131     </asp:TextBox>
132     <br />
133
134     <asp:Label id = "stateLabel" runat = "server"
135         style = "position: absolute; left:56px">
136         State/Province:
137     </asp:Label>
138     <asp:TextBox id = "stateTextBox" runat = "server"
139         style = "position: absolute; left:150px">
140     </asp:TextBox>
141     <br />
142
143     <asp:Label id = "emailLabel" runat = "server"
144         style = "position: absolute; left:105px">
145         E-mail:
146     </asp:Label>
147     <asp:TextBox id = "emailTextBox" runat = "server"
148         style = "position: absolute; left:150px">
149     </asp:TextBox>
```

**mailinglist.aspx**  
**(6 of 8)**

```
150     <br />
151     <br />
152
153     <asp:Button id = "clearButton" runat = "server" width = "57px"
154         style = "position: absolute; left:150px"
155         Text = "Clear" OnClick = "clearButton_Click">
156 </asp:Button>
157 <asp:Button id = "submitButton" runat = "server"
158     style = "position: absolute; left:220px"
159     Text = "Submit" OnClick = "submitButton_Click">
160 </asp:Button>
161 <br />
162 <br />
163
164     <asp:DataGrid id = "dataGrid" runat = "server"
165         HorizontalAlign = "Left" BorderColor = "#E7E7FF"
166         GridLines = "Horizontal" BackColor = "White"
167         BorderStyle = "None" CellPadding = "3">
168         <SelectedItemStyle Font-Bold = "True" ForeColor = "#F7F7F7"
169             BackColor = "#738A9C">
170         </SelectedItemStyle>
171         <AlternatingItemStyle BackColor = "#F7F7F7">
172         </AlternatingItemStyle>
173         <ItemStyle HorizontalAlign = "Left" ForeColor = "#4A3C8C"
174             BackColor = "#E7E7FF">
```

mailinglist.aspx  
(7 of 8)

```

175         </ItemStyle>
176         <HeaderStyle Font-Bold = "True" ForeColor = "#F7F7F7"
177             BackColor = "#4A3C8C">
178         </HeaderStyle>
179         <FooterStyle ForeColor = "#4A3C8C" BackColor = "#B5C7DE">
180         </FooterStyle>
181     </asp:DataGrid>
182 </form>
183
184 </body>
185 </html>

```

mailinglist.aspx  
(8 of 8)

Join our Mailing List - Microsoft Internet Explorer

Address: http://localhost/deitel/Mailing/mailingList.aspx

Fill in the fields below to join our mailing list:

Name:

Address:

City:

State/Province:

E-mail:

Clear Submit

Name	Address	City	State	E-mail
CeCe Sharp	54321 Ant Way	Antown	AV	cecesharp@deitel.com
Luna Tic	32415 Tic Ave	Antville	VA	lunatic@deitel.com

Done Local intranet

## 23.9 Reading and Writing Text Files

Property	Description
PhysicalApplicationPath	The file system path of the currently executing server application's root directory.
IsSecureConnection	Boolean. Determines if the request came in through SSL (Secure Sockets Layer).
UserHostAddress	Client's IP address.
UserHostName	Client's DNS host name.
HttpMethod	Request method (i.e., GET or POST).
UserAgent	Provides information about the client making the request.
Cookies	Retrieves the cookies residing on the client.
ServerVariables	Retrieves the variables about the server.

Fig. 23.30 Request object properties.

## 23.10 Connecting to a Database in ASP.NET

- OleDbDataReader object
  - Reads data from a database
- OleDbConnection object
  - Represents connection to database
- OleDbCommand object
  - Two parameters
    - queryString
      - Contains SQL to execute
    - Database connection
- ExecuteReader
- DataSet
  - Set of data and includes the tables that contain and order it
- OleDbDataAdapter
  - Retrieve information from database and place resulting information in DataSet

```

1 <%@ Page Language="JScript" %>
2 <%@ Import Namespace="System" %>
3 <%@ Import Namespace="System.Data" %>
4 <%@ Import Namespace="System.Data.OleDb" %>
5 <%@ Register TagPrefix="Header" TagName="ImageHeader"
6   Src="imageHeader.ascx" %>
7
8 <!-- Fig 23.31: authors.aspx -->
9 <!-- This page allows a user to choose an -->
10 <!-- author and display that authors name -->
11
12 <html>
13   <body>
14     <script language = "JScript" runat = "server">
15
16       function Page_Load( sender : Object, events : EventArgs )
17       {
18         if ( !IsPostBack )
19         {
20           var dataBaseConnection : OleDbConnection = new
21             OleDbConnection( ConfigurationSettings.AppSettings(
22               "ConnectionString" ) );
23
24           var queryString : System.String =
25             "SELECT lastName, firstName FROM Authors";

```

authors.aspx  
(1 of 5)

```
26
27     dataBaseConnection.Open();
28
29     var dataBaseCommand : OleDbCommand =
30         new OleDbCommand( queryString, dataBaseConnection );
31
32     var dataReader = dataBaseCommand.ExecuteReader();
33
34     // while we read a row from result of
35     // query, add first item to drop down list
36     while ( dataReader.Read() )
37         nameList.Items.Add( dataReader.GetString( 0 ) +
38             ", " + dataReader.GetString( 1 ) );
39
40     // close database connection
41     dataBaseConnection.Close();
42 }
43 else
44 {
45     dataGrid.DataSource = GetData();
46     dataGrid.DataBind();
47 }
48 } // end Page_Load
49
```

authors.aspx  
(2 of 5)

```
50 // Read a database and return the DataView
51 function GetData() : ICollection
52 {
53     var set : DataSet = new DataSet();
54
55     // establish a connection, and query the database
56     var dataBaseConnection: OleDbConnection = new
57         OleDbConnection( ConfigurationSettings.AppSettings(
58             "ConnectionString" ) );
59
60     var authorID : int = nameList.SelectedIndex + 1;
61
62     var queryString : String =
63         "SELECT Titles.Title, Titles.ISBN, " +
64         "Publishers.PublisherName FROM AuthorISBN " +
65         "INNER JOIN Titles ON AuthorISBN.ISBN = " +
66         "Titles.ISBN, Publishers WHERE " +
67         "(AuthorISBN.AuthorID = " + authorID + ")";
68
69     var dataBaseCommand : OleDbCommand =
70         new OleDbCommand( queryString, dataBaseConnection );
71
72     var dataAdapter : OleDbDataAdapter =
73         new OleDbDataAdapter( dataBaseCommand );
74
```

**authors.aspx**  
**(3 of 5)**



```
75     dataAdapter.Fill( set );
76
77     // close database connection
78     dataBaseCommand.Connection.Close();
79
80     var dataView : DataView = new DataView( set.Tables[ 0 ] );
81     dataView.Sort = "Title";
82
83     return dataView;
84 } // end GetData
85
86 </script>
87
88 <form runat = "server">
89
90     <Header:ImageHeader id = "head" runat = "server">
91     </Header:ImageHeader>
92     <br />
93
94     Authors:
95     <asp:DropDownList id = "nameList" runat = "server"
96         width = "158px" Height = "22px">
97     </asp:DropDownList>
98     <asp:button id = "button" text = "select" runat = "server">
99     </asp:button>
```

**authors.aspx**  
**(4 of 5)**

```

100     <p>
101         <asp:DataGrid id = "dataGrid" runat = "server">
102             </asp:DataGrid>
103     </p>
104
105 </form>
106 </body>
107 </html>

```


**authors.aspx  
(5 of 5)**

http://localhost/deitel/Database/authors.aspx - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address http://localhost/deitel/Database/authors.aspx Go Links >>

 **Bug2Bug.com**

Authors:

Title	ISBN	PublisherName
Advanced Java 2 Platform How to Program	0130895601	Prentice Hall
Advanced Java 2 Platform How to Program	0130895601	Prentice Hall PTG
C How to Program	0131180436	Prentice Hall
C How to Program	0131180436	Prentice Hall PTG
C How to Program	0130895725	Prentice Hall
C How to Program	0130895725	Prentice Hall PTG

Done Local intranet

```
1 <!-- Fig. 23.32: imageHeader.ascx      --%>
2 <!-- Listing for the header user control --%>
3
4 <asp:Image id = "Image1" runat = "server"
5     ImageUrl = "bug2bug.png">
6 </asp:Image>
```

**imageHeader.ascx**  
**(1 of 1)**

```
1 <!-- Fig 23.33: Web.config -->
2 <!-- Web.Config Configuration File -->
3
4 <configuration>
5   <appSettings>
6     <add key = "ConnectionString"
7       value = "Provider=Microsoft.Jet.OLEDB.4.0;
8       Data Source=C:\Inetpub\wwwroot\Database\Books.mdb" />
9   </appSettings>
10 </configuration>
```

**Web.config  
(1 of 1)**

## 23.11 Code-Behind Approach

- Alternative method of adding scripts to make static content dynamic
- Not included in the .aspx file
- Must be compiled first

```
1 <%@ Page Language="JScript"
2     Src="adRotator.aspx.js" Inherits="MyCodeBehind" %>
3
4 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
5     "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
6
7 <!-- Fig. 23.34: adRotator.aspx -->
8 <!-- ASP.NET AdRotator example -->
9
10 <html>
11     <head>
12         <title>Using An AdRotator</title>
13     </head>
14
15     <body>
16         <form action = "adRotator.aspx" method = "post" runat = "server">
17
18             <asp:AdRotator AdvertisementFile = "ads.xml"
19                 BorderColor = "black" Borderwidth = "1" runat = "server"/>
20
21             <table>
22                 <tr>
23                     <td>
24                         Name: <asp:TextBox id = "name" runat = "server"/>
25                     </td>
```

## adRotator.aspx (1 of 3)

```
26         <td>
27             <asp:RequiredFieldValidator id = "requiredCheck"
28                 ControlToValidate = "name"
29                 Display = "Static"
30                 runat = "server">
31                 Please enter your name.
32             </asp:RequiredFieldValidator>
33         </td>
34     </tr>
35 </table>
36
37 <br />
38 Do you like ice cream?
39
40 <asp:RadioButtonList id = "iceCream" runat = "server">
41     <asp:ListItem>Yes</asp:ListItem>
42     <asp:ListItem>No</asp:ListItem>
43 </asp:RadioButtonList>
44
45 <br />
46 How many scoops would you like? (0-45)
47
48 <asp:TextBox id = "scoops" runat = "server" />
49
50 <br />
```

**adRotator.aspx**  
**(2 of 3)**

```
51     <asp:button text = "Submit" OnClick = "submitButton_Click"
52         runat = "server"/>
53
54     <asp:RangeValidator id = "rangeCheck"
55         ControlToValidate = "scoops"
56         MinimumValue = "0"
57         MaximumValue = "45"
58         Type = "Integer"
59         EnableClientScript = "false"
60         Text = "We cannot give you that many scoops."
61         runat = "server" />
62
63     <center>
64         <h1> <asp:label id = "message" runat = "server"/> </h1>
65     </center>
66
67 </form>
68 </body>
69 </html>
```

**adRotator.aspx**  
**(3 of 3)**



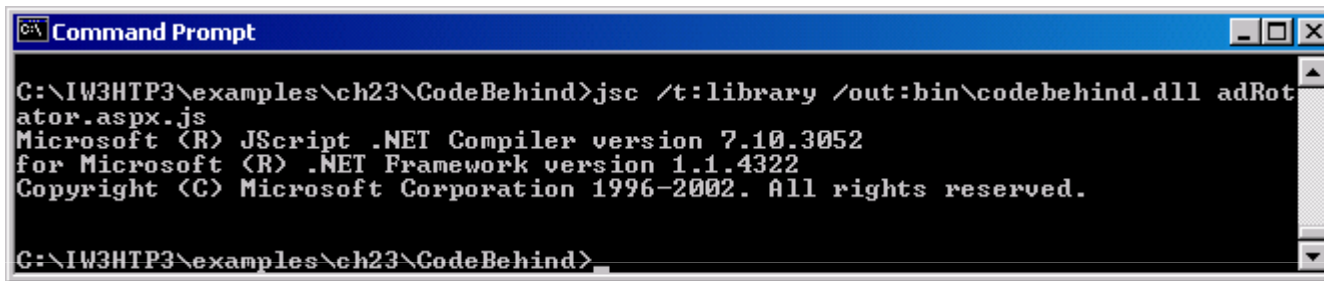
```
1 // Fig 23.35: adRotator.aspx.js
2 // Code-behind for the adRotator.aspx
3 // user interface.
4
5 import System;
6 import System.Web.UI;
7 import System.Web.UI.WebControls;
8
9 public class MyCodeBehind extends Page
10 {
11     protected var name : TextBox;
12     protected var scoops : TextBox;
13     protected var iceCream : RadioButtonList;
14     protected var message : Label;
15
16     public function submitButton_Click(
17         sender : Object, events : EventArgs ) : void
18     {
19         if ( IsPostBack )
20         {
21             if ( iceCream.SelectedItem == "Yes" )
22             {
23                 message.Text = name.Text + " likes ice cream.";
24             }
25         }
26     }
27 }
```

adRotator.aspx.js  
(1 of 2)

```
25     else
26     {
27         message.Text = name.Text + " does not like ice cream.";
28     }
29 }
30 } // end submitButton_Click
31 } // end class MyCodeBehind
```

**adRotator.aspx.js**  
**(2 of 2)**

## 23.11 Code-Behind Approach



```
C:\IW3HTP3\examples\ch23\CodeBehind>jsc /t:library /out:bin\codebehind.dll adRotator.aspx.js
Microsoft (R) JScript .NET Compiler version 7.10.3052
for Microsoft (R) .NET Framework version 1.1.4322
Copyright (C) Microsoft Corporation 1996-2002. All rights reserved.

C:\IW3HTP3\examples\ch23\CodeBehind>
```

Fig. 23.36 Using the command prompt to compile the code-behind.

## 23.11 Code-Behind Approach

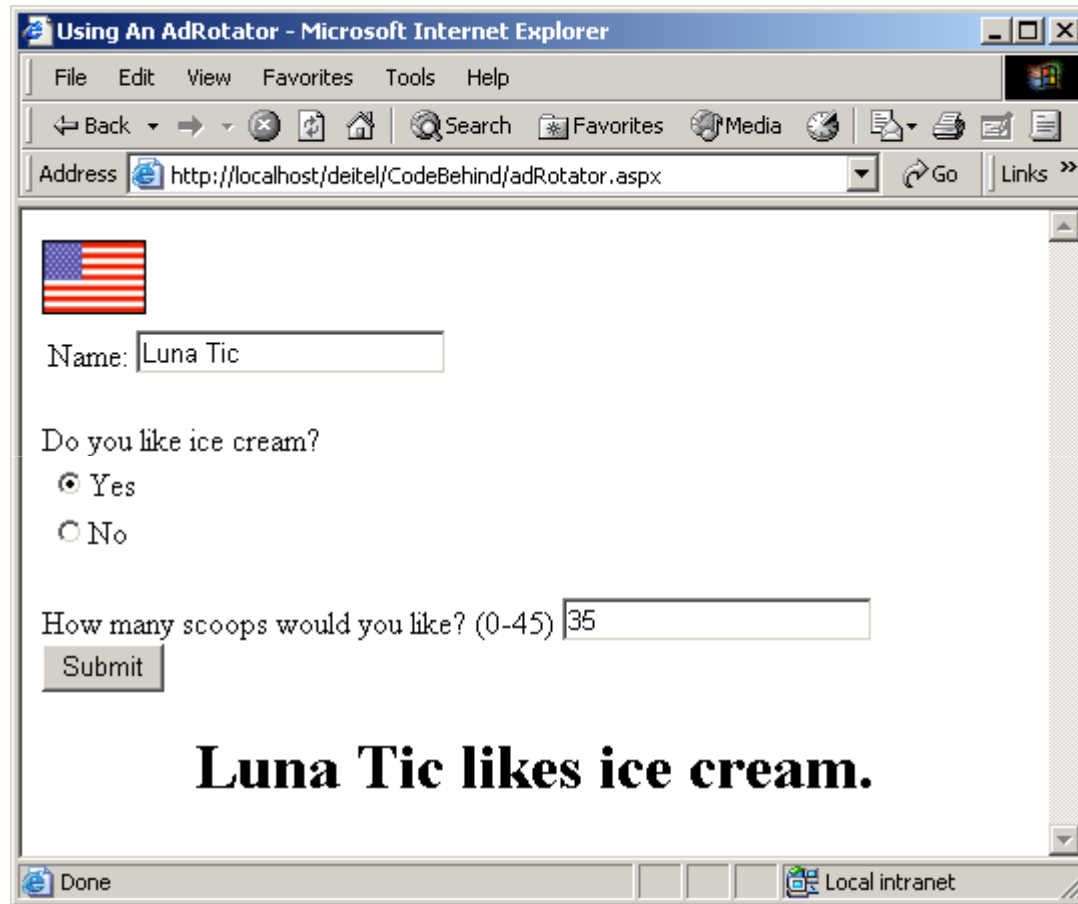


Fig. 23.37 Output of AdRotator.aspx using the code-behind method.

## 23.12 ASP.NET Web Services

- Enables distributed computing by allowing one machine to call methods on other machines
- Defining the Web Service
  - Remote machine
    - Store Web service
    - Web-service methods
      - Remote Procedure Call (RPC)
  - Business-to-business (B2B) transactions
- Viewing the Web Service
  - Service Description
    - Web Service Description Language (WSDL)

```
1 <%@ webService Language="JScript" Class="NumberService" %>
2
3 // Fig. 23.38: number.asmx
4 // A simple web service
5
6 import System;
7 import System.Web.Services;
8
9 public class NumberService extends WebService
10 {
11     // Determines whether all the characters in a string are digits
12     webMethod public function isNum( number : String ) : Boolean
13     {
14         var digitArray : char[];
15         digitArray = number.ToCharArray();
16
17         for ( var i : int = 0; i < digitArray.Length ; i++ )
18         {
19             if ( !Char.IsDigit( digitArray[ i ] ) )
20             {
21                 return false;
22             }
23         }
24
```

**number.asmx**  
**(1 of 2)**

```
25     return true;
26 } // end isNum
27
28 WebMethod public function Add( a : float, b : float) : float {
29     return a + b;
30 }
31 } // end class NumberService
```

**number.asmx**  
**(2 of 2)**

## 23.12 ASP.NET Web Services

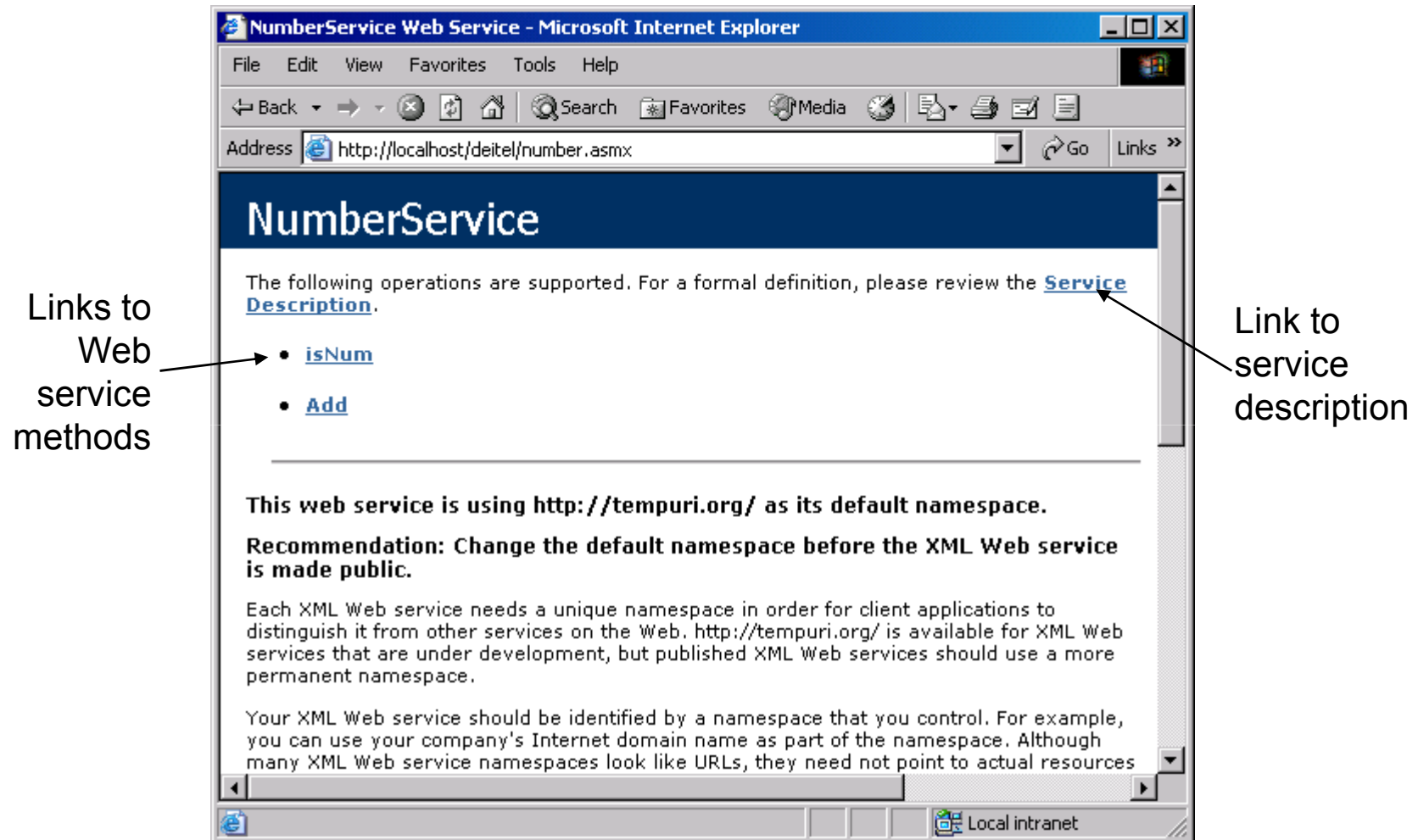


Fig. 23.39 ASMX file rendered in Internet Explorer.



## 23.12 ASP.NET Web Services

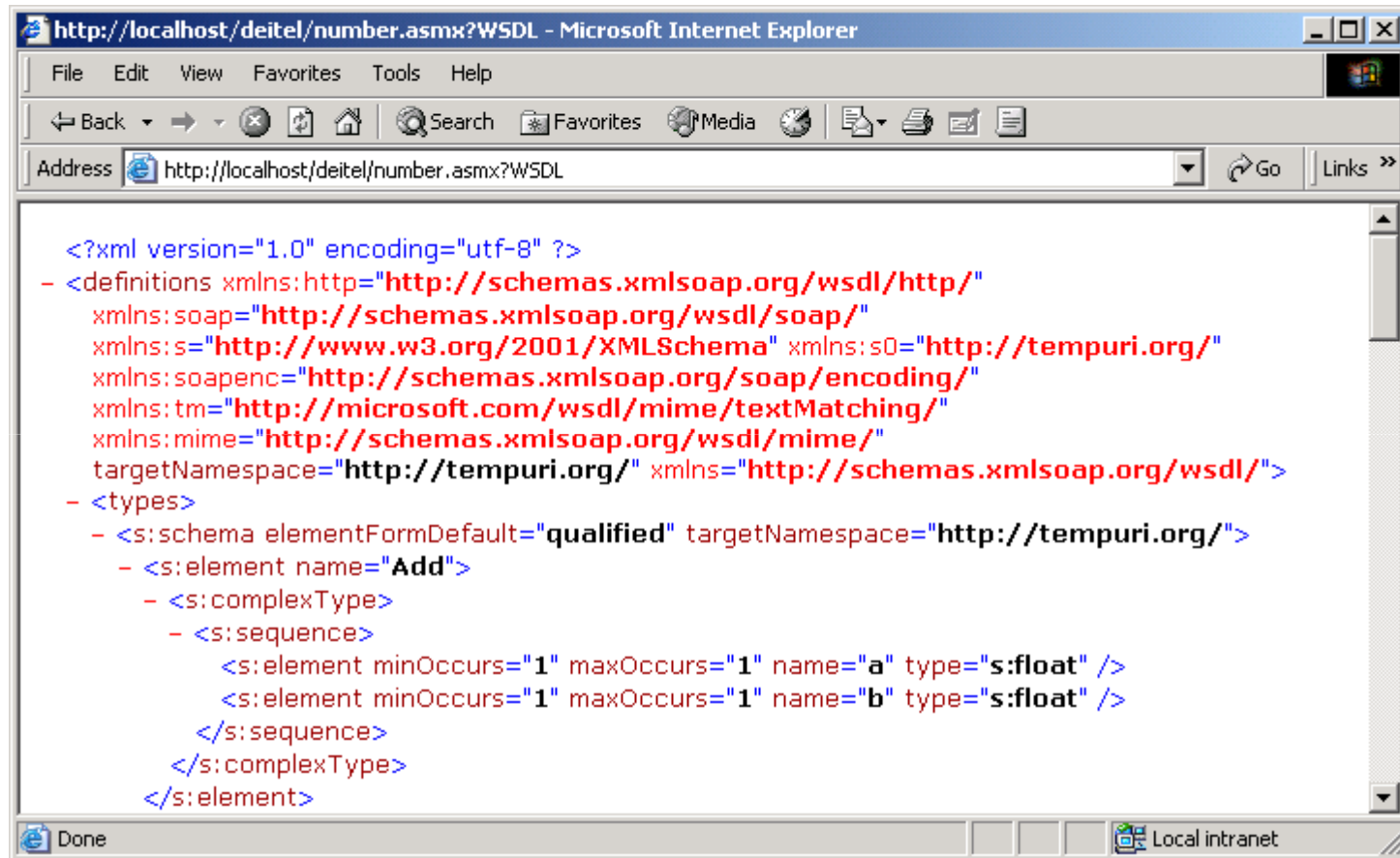


Fig. 23.40 Service description for a Web service.

## 23.12 ASP.NET Web Services

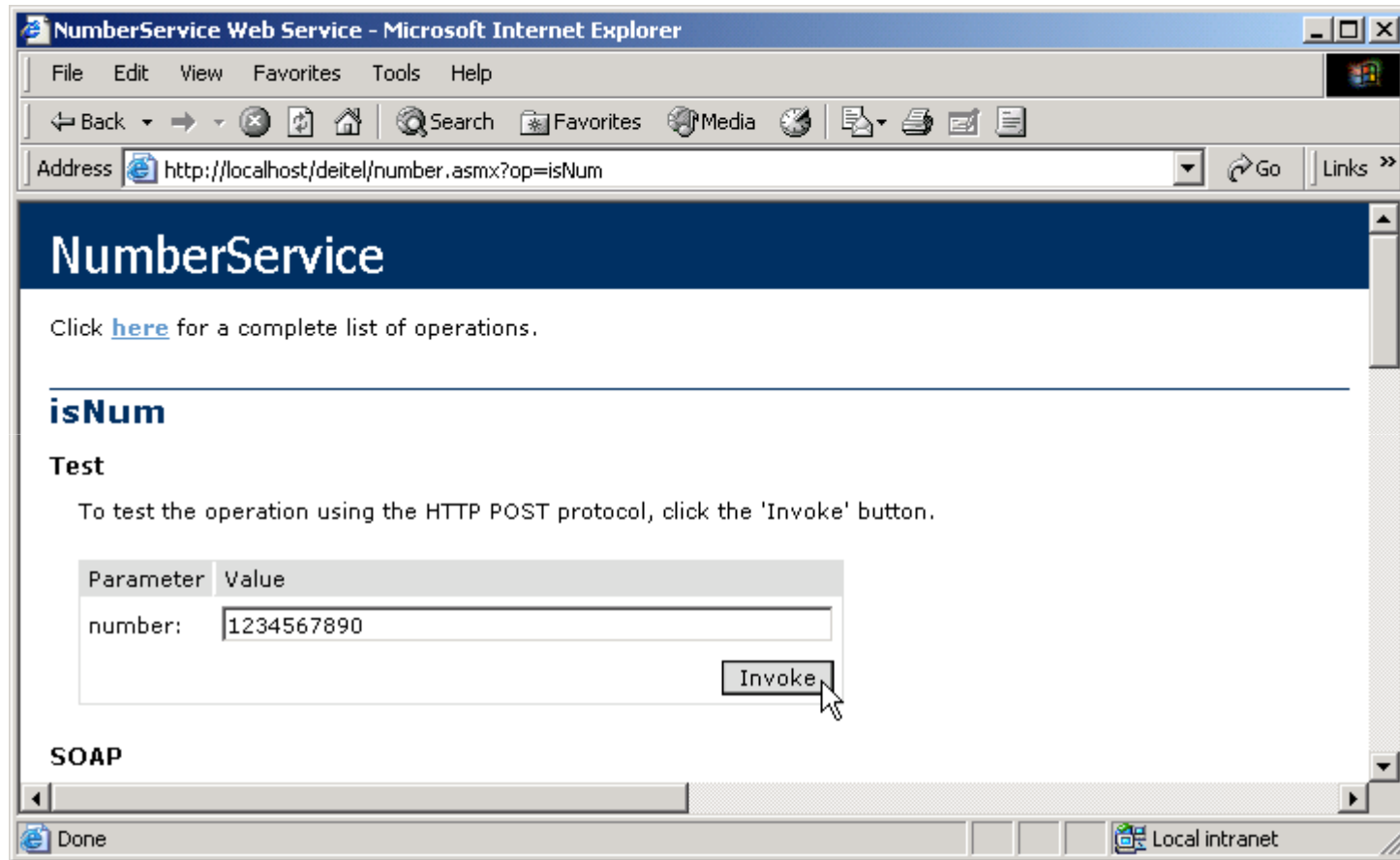


Fig. 23.41 Invoking a method of a Web service from a Web browser.

## 23.12 ASP.NET Web Services

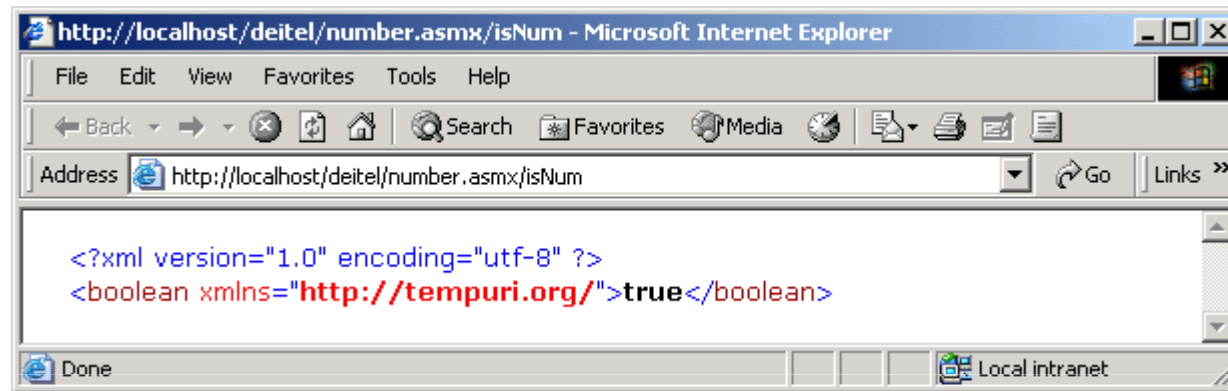


Fig. 23.42 Results of invoking a Web-service method from a Web browser.

## 23.13 Web Resources

- [www.asp.net](http://www.asp.net)
- [www.asp101.com/aspdotnet/aspplus/index.asp](http://www.asp101.com/aspdotnet/aspplus/index.asp)
- [www.411asp.net](http://www.411asp.net)
- [www.aspfree.com](http://www.aspfree.com)
- [www.aspng.com](http://www.aspng.com)
- [www.aspnetfaq.com](http://www.aspnetfaq.com)
- [www.123aspx.com](http://www.123aspx.com)
- [msdn.microsoft.com/netframework](http://msdn.microsoft.com/netframework)
- [www.xmlforasp.net](http://www.xmlforasp.net)
- [www.ondotnet.com/topics/dotnet/asp.net](http://www.ondotnet.com/topics/dotnet/asp.net)