

第 5 章 ASP.NET 控件介绍

使用 ASP.NET 开发 Web 程序之所以方便、快捷,是因为它有一组强大的控件库,包括 Web 服务器控件、Web 用户控件、HTML 服务器控件等。

5.1 ASP.NET 网页介绍

ASP.NET 网页基于开发动态网页的 Microsoft ASP.NET 技术,其网页文件保存为 *.aspx,属于服务器端动态页面,并由 IIS Web 服务器管理。ASP.NET 网页可看作 Web 应用程序的可编程用户接口。

ASP.NET 网页具有以下特点:

(1)使用服务器端代码来实现应用程序逻辑,并在浏览器或客户端设备中向用户提供信息。

(2)自动为样式、布局等功能呈现正确的、符合浏览器要求的 HTML,兼容所有浏览器或移动设备,也可以将 ASP.NET 网页设计为在特定浏览器上运行并利用浏览器特定的功能。

(3)兼容 .NET 公共语言运行库所支持的任何语言,其中包括 Microsoft Visual Basic、Microsoft Visual C#、Microsoft J# 和 Microsoft JScript .NET。

(4)基于 Microsoft .NET Framework 生成。它提供了 Framework 的所有优点,包括托管环境、类型安全性和继承。

(5)具有灵活性,可以向它们添加用户创建的控件和第三方控件。

5.1.1 表单的基本概念

在 ASP.NET 中,一个网页或窗口被看成是一个 Web Form(网页表单)。Web Form 把 Web 应用程序的用户界面分割成两部分:可视化用户组件界面(Visual Component)部分和接口逻辑(User Interface Logic)部分。其中,前者指包含 HTML 标记及服务器端控件声明的部分(也就是浏览器中看到的部分),这部分好像是一个大容器,开发者可以在其中放入各种 ASP.NET 控件;后者是指由开发者编写的用于与网页表单交互的代码。如果使用普通的文本编辑器进行设计,则上述两个部分放在同一个文件中,并以 .aspx 作为后缀;如果使用的是 Visual Studio .NET 进行设计,则可视化组件与接口逻辑将分别处于不同的文件中。

当用户请求包含服务器端控件的 Web 网页时,被请求的网页首先在服务器端执行,然后生成 HTML 后才送至客户端,浏览器再将结果显示给用户。需要注意的是,Web Form 的某些高级功能必须在特定的浏览器(如 IE 5.5 以上)才能正常运行。

5.1.2 回发和往返行程

ASP.NET 页面作为代码在服务器上运行。因此,要得到处理,页面必须配置为当用户单击按钮(或者当用户选中复选框或与页面中的其他控件交互)时提交到服务器。每次页面都会提交回自身,以便它可以再次运行其服务器代码,然后向用户呈现其自身的新版本。

具体的处理循环如下:

(1)用户请求页面。使用 HTTP GET 方法请求页面,页面第一次运行,执行初步处理(如果已通过编程让它执行初步处理)。

(2)页面将标记动态呈现到浏览器,用户看到的网页类似于其他网页。

(3)用户输入信息或从可用选项中进行选择,然后单击按钮(如果用户单击链接而不是按钮,页面可能仅仅定位到另一页,而第一页不会被进一步处理)。

(4)页面发送到 Web 服务器。

浏览器执行 HTTP POST 方法,该方法在 ASP.NET 中称为“回发”。确切地说,页面发送回其自身。例如,用户正在使用 Default.aspx 页面,则单击该页上的某个按钮可以将该页发回服务器,发送的目标则是 Default.aspx。

(5)在 Web 服务器上,该页再次运行,并且可在页面上使用用户输入或选择的信息。

(6)页面执行通过编程所要执行的操作。

(7)页面将其自身呈现回浏览器。

只要用户在该页面中工作,此循环就会继续。用户每次单击按钮时,页面中的信息会发送到 Web 服务器,然后该页面再次运行。每个循环称为一次“往返行程”。由于页面处理发生在 Web 服务器上,因此页面可以执行的每个操作都需要一次到服务器的往返行程。

需要注意的是,ASP.NET 网页可以运行客户端脚本。客户端脚本不需要到服务器的往返行程,这对于用户输入验证和某些类型的 UI 编程十分有用。

此外,在某些情况下,用户可能需要将某页发送到其他页,而不是发送到该页自身,这种行为称为“跨页发送”。例如,创建一系列用来处理客户订单的页面,每页都可以发送到序列中的下一页。

5.1.3 页面生存期

ASP.NET 网页与桌面应用程序中的窗体不同,ASP.NET 网页在用户使用窗体时不会启动或运行,仅当用户单击“关闭”按钮时才会卸载。这是由于 Web 具有断开链接的特性。浏览器从 Web 服务器请求页面时,浏览器和服务器相连的时间仅够处理请求。Web 服务器将页面呈现到浏览器之后,链接即终止。如果浏览器对同一 Web 服务器发出另一个请求,即使是对同一个页面发出的,该请求仍会作为新请求来处理。

Web 这种断开链接的特性决定了 ASP.NET 页的运行方式。用户请求 ASP.NET 网页时,将创建该页的新实例。该页执行其处理,将标记呈现到浏览器,然后该页被丢弃。如果用户单击按钮以执行“回发”,将创建该页的新实例;该页执行其处理,然后再次被丢弃。这样,每个回发和往返行程都会导致生成该页的一个新实例。正因为如此,网页如同人一样有了“生存期”,它始于服务器对其进行的解释,随着解释的完毕,生存期也就结束,即当在浏览器上看到 ASP.NET 网页时它的生存期已经结束了。

5.2 服务器控件概述

服务器控件是动态网页技术的一大进步,它真正地将后台程序和前端网页融合在了一起。服务器控件的广泛应用简化了应用程序的开发,提高了工作效率。

5.2.1 服务器控件的定义

服务器控件是指在服务器上执行程序逻辑的组件,常常具有一定的用户界面。服务器控件包含在 ASP.NET 页面中,当运行页面时,用户与控件发生交互行为,当页面被提交时,控件可在服务器端引发事件,根据相关事件处理程序来进行事件处理。

5.2.2 服务器控件分类

ASP.NET 提供了多种服务器控件,根据定义的方式可分为两大类。

(1)HTML 服务器控件。HTML 服务器控件由普通 HTML 控件转换而来,外观上基本与普通 HTML 控件一致。

(2)ASP.NET 标准服务器控件。ASP.NET 标准服务器控件比 HTML 服务器控件具有更多内置功能,可以说这些控件是构建 ASP.NET Web 应用的主力军。服务器控件不仅包括窗体控件(如按钮和文本框),而且还包括特殊用途的控件(如日历、树视图和菜单控件)。Web 服务器控件与 HTML 服务器控件相比更为抽象,因为其对象模型不一定反映 HTML 语法。

在 ASP.NET 中,除了利用上述两类控件外,还可以使用与 Web 网页相同的编程技巧来定义自己的控件,即用户控件。用户控件是完全由开发人员自行设计开发的控件,可自定义其外观、功能、属性、方法、事件等特征,这是用户控件与 ASP.NET 标准服务器控件的本质区别。

ASP.NET 用户控件可以嵌入到其他 ASP.NET 网页中,这是一种创建工具栏和其他可重用元素的捷径。可以这样简单理解,当一个 Web 窗体被当作 Server 控件使用时,这个 Web 窗体便是用户控件。

5.2.3 服务器控件的创建

控件是.aspx 页面上重要的元素,服务器控件的创建通常可以使用以下两种方法。

(1)使用鼠标创建。在“工具箱”窗口中选择需要的控件后,通过使用鼠标拖动到页面上或直接双击,即可在页面上创建相应控件。此种方法非常简单,在工具箱中选择 HtmlInput (Button)控件后双击,即可在网站默认的 Default.aspx 设计页面上创建一个 HtmlButton 控件,如图 5-1 所示。

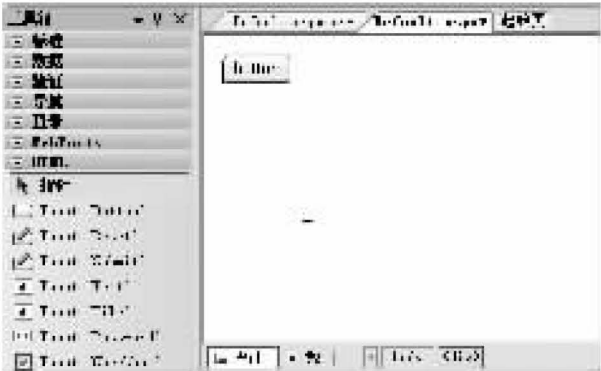


图 5-1 在设计页中创建 HtmlButton 控件

(2)在页面 HTML 视图中,通过编辑代码实现。同 HTML 中的控件不同,服务器控件的特点是具有 `Runat="Server"` 属性。当 ASP.NET 网页执行时,.NET 会检查页面上的标记有无 `Runat="Server"` 属性,如果没有就会被直接发送到客户端的浏览器进行解析,如果有则表示这个控件可以被.NET 程序所控制,需要等到程序执行完毕再将 HTML 控件的执行结果发送到客户端浏览器。

下面给出一个简单的例子,以使对第二种方法有一个更加直观的了解。

首先创建一个 ASP.NET Web 应用程序,取名为“Ex_control”。然后打开自动添加的页面 `Default.aspx`,在 `<div>` 和 `</div>` 标记之间输入以下代码:

HTML 输入框:`<input type="text" value="abc" runat="server" id="TextBox_Html"><br>`

HTML 按钮:`<input type="button" value="OK" runat="server" id="Button_Html"><br>`

Web 输入框:`<asp:TextBox ID="TextBox_Web" runat="Server">abc</asp:TextBox><br>`

Web 按钮:`<asp:Button ID="Button_Web" runat="Server" Text="OK"></asp:Button><br>`

当转到设计页,可以发现这里定义了四个控件,如图 5-2 所示。



图 5-2 通过编辑代码创建的控件

5.2.4 服务器控件的属性、事件和方法

ASP.NET Web 程序设计采用了面向对象的编程思想,服务器控件即为一系列的类,如

Button 控件类等。每一个具体的服务器控件便是这个类的一个具体实例,称之为对象,如在页面上新创建的按钮控件 Button1。

属性描述对象“静”的特性,如控件的背景色 BackColor,如图 5-3 所示的 HtmlButton 控件的属性窗口。属性窗口可通过执行 Visual Studio 2005 主窗口中的“视图”→“属性窗口”命令打开或关闭。



图 5-3 HtmlButton 控件的属性窗口

对象“动”的特性即对象的行为,如单击按钮 Button1 后在页面上出现文字信息,由事件和方法来实现。具体过程为在 ASP.NET 网页中,与服务器控件关联的事件在客户端(浏览器)上引发,ASP.NET 网页在 Web 服务器上处理。对于在客户端引发的事件,ASP.NET Web 控件事件模型要求在客户端捕获事件信息,并通过 HTTP POST 将事件消息传送到服务器。网页需要解释该 POST 已确定所发生的事件,然后执行在要处理事件的服务器中的相应方法。

下面介绍如何将事件绑定到方法。

通过服务器控件的事件,可以响应 ASPX 页面的动作。可以说一个事件就是一条消息,如某按钮已被单击。为了做出动作,在服务器端的代码中必须将消息转换成代码中的方法调用。

对于在页上声明的控件,可以通过在控件的标记中设置属性将事件绑定到方法。例如下面的代码将 Button 控件 Button1 的单击(Click)事件绑定到名为“Button1_Click”的方法上。在运行时,当按钮 Button1 受到单击动作时,ASP.NET 将查找名为“Button1_Click”的方法,并执行该方法。

```
<asp:Button ID="Button1" runat="Server" OnClick=" Button1_Click"
Text="练习"/>
```

下面的方法就是一个典型的服务器控件事件绑定的方法。

```
protected void Button1_Click(object sender, EventArgs e)
{
}

}
```

那么如何创建这种绑定呢?在 Visual Studio 2005 中非常简单,只需要在设计视图中双击要为其创建默认事件处理程序的页面或控件即可。此时将自动创建默认事件的处理程序,并打开代码编辑器,光标此时位于事件处理程序中,如图 5-4 所示。



图 5-4 Button1 控件的代码编辑器窗口

此外,也可以在属性窗口中创建事件处理程序。步骤如下:

- (1)在设计视图中选择控件。
- (2)在“属性”对话框中单击“事件符号”,将显示所选控件的事件列表。
- (3)在事件旁边的单元格中执行下列其中一个操作:
 - 双击。
 - 输入要创建的事件处理程序的名称。
 - 从下拉列表中选择已有的事件处理程序名称。

下面了解常用的页面加载事件。当 ASPX 页面加载时就触发了这个事件。通过在设计视图中双击页面就可以创建该事件,默认的事件名称为“Page_Load”。下面的例子为页面的加载事件实现了方法,该方法实现输出文字“页面加载”,运行后如图 5-5 所示。



图 5-5 页面的加载事件

当页面出现时,便会自动执行该方法,代码如下:

```
protected void Page_Load(object sender, EventArgs e)
{
    Page.Response.Write("页面加载");
}
```

5.3 HTML 服务器控件

在 ASP.NET 中引入 HTML 服务器控件是 ASP.NET 对 ASP 的一次重大改良,对页面设计的方法和提交数据的方式都发生了根本性的改变。简单地说,HTML 服务器控件是从普通 HTML 标记衍生来的,基于 CLR 的可直接在服务器端使用的 ASP.NET 控件。

5.3.1 HTML 服务器控件概述

HTML 服务器控件运行在服务器上,是 .NET 程序可以操作的页面对象,如按钮、输入框、图像,HTML 服务器端控件由标准的 HTML 标记衍生而来,在外形上与普通的 HTML 元素非常相似,几乎与 HTML 标记有一一对应的关系,因此两者的属性也基本相同。将 HTML 标记转化为 HTML Server 控件的方法为:在需要转换为 HTML 服务器端控件的 HTML 标记中加入 `runat="server"` 及 `id="XXX"` 属性即可。有些还需要在 HTML 标记结尾处添加上一个 `/` 作为结束标记。HTML 服务器端控件与 HTML 标记的区别在于 HTML Server 控件的功能更为强大,且在服务器端运行。

使用 HTML 服务器端控件具有以下优点:

- (1) 页面显示代码和页面控制代码分离,便于页面的编写和维护。
- (2) 具有“回送”功能。
- (3) 既支持服务器端事件,又支持客户端事件。

HTML 服务器控件都包含在 `System.Web.UI.HtmlControls`,该空间主要包含三个子空间。这里的三个子空间分别是:

- (1) `HtmlInputControl` 子空间:用作定义所有 HTML 输入控件,如 `<input type="text">`、`<input type="submit">` 和 `<input type="file">` 元素。
- (2) `HtmlContainerControl` 子空间:定义内部可以包含另外的控件或其他内容的控件,这些控件必须带结束标记。
- (3) `HtmlImage` 子空间:用作定义图像控件。

HTML 服务器控件具有一些通用属性。所谓通用属性,就是指存在于每一个 HTML 服务器控件中的相同的属性名称与属性设定。例如,控件显示在浏览器上的样式 `Style` 属性就是一个通用属性。

1. Attributes 属性

`Attributes` 属性获取或设置 HTML Server 控件的所有属性值。

语法格式:

对象名称.`Attributes.Add`("属性名称","值")

使用说明:属性名称依控件所支持的属性名称而定;对于不同控件可能含有不同的属性名称。

2. Style 属性

`Style` 属性获取或设置 HTML Server 控件的 CSS 样式属性设定值。

语法格式:

对象名称.`Style.Add`("CSS 属性","值")

使用说明:CSS 属性依控件所支持的 CSS 属性名称而定;对于不同控件可能含有不同的属性名称。CSS 样式所支持的属性的确非常多,但是具有一定的规律。用户可以参考 CSS 样式表手册(如在 VS.NET 中以“使用 CSS 样式”为名的联机帮助)。

例如,通过对 `Style` 属性进行设定,改变 `HtmlInputText` 文本框控件 `text1` 的字体名称属性 `FAMILY_NAME`、字体颜色属性 `COLOR`。需要注意的是,设置 `Style` 属性时,双引号中的属性名称是严格区分大小写的。

`Text1.Style.Add("FAMILY_NAME","隶书")`

```
Text1.Style.Add("Color","Red")
```

3. Parent 属性

Parent 属性引用在执行期间的父对象来源,也可以指向父对象。

在 Web Form 网页中设计 HTML 控件时,采用容器(Container)架构来创建对象,所以在 HTML Server 控件的引用过程中可以引用父对象。这样既可扩展程序的处理能力,又可在执行期间删除对象。

语法格式:

(1)引用父对象:对象名称.Parent

(2)获得父对象引用的类的来源:对象名称.Parent.ToString()

4. Controls 属性

Controls 属性引用指定的 HTML Server 控件在执行期间的子对象集合。

语法格式:

(1)引用某个子对象:对象名称.Controls[i]

(2)获得子对象引用的类的来源:对象名称.Controls[i].ToString()

5. Disabled 属性

Disabled 属性用于启用或停用 HTML Server 控件。

语法格式:

对象名称.Disabled=True | False

使用说明:Disabled 属性设为 True 时,将指定的 HTML Server 控件设定为停用状态。

6. Visible 属性

Visible 属性用于是否显示一个 HTML Server 控件。

语法格式:

对象名称.Visible=True | False

使用说明:Visible 属性设为 True 时,HTML Server 处于可见状态。

7. HTML 服务器输入控件的共有属性

输入控件所共有的属性主要有 Value 和 Type 属性。其中,Value 属性获取或设置与输入控件关联的值,Type 属性获取输入控件的类型。

8. HTML 服务器容器控件的共有属性

HTML 服务器容器控件的共有属性为 InnerHtml 和 InnerText。这两个属性用于设置<标记>和</标记>之间的 HTML 内容或纯文本。

语法格式:

对象名称.InnerHtml=字符串

对象名称.InnerText=字符串

InnerHtml 是用 HTML 格式来显示标记间的内容,InnerText 是用纯文本来显示标记间的内容,包括 HTML 标记。

5.3.2 HTML 输入控件

HTML 输入控件是用户使用最为广泛的控件,常见的应用如输入信息和上传文件功能等。HtmlInput 控件由七个控件组成,均由<input>标记衍生而来,彼此的差别也仅在于其 Type 属性,这与普通的 HTML 中的<input>标记是一样的。

1. HtmlInputButton 按钮控件

当需要使用按钮来提交数据时,可以使用 HtmlInputButton 控件。

1) 创建 HtmlInputButton 控件

有两种方法在页面上创建一个 HtmlInputButton 控件。

(1)在页面的源视图中通过添加代码实现。例如,添加一个 ID 为“Button1”的 HtmlInputButton 对象,可以通过下面的代码实现:

```
<div>  
    <input id="Button1" type="button" value=" Button1" runat="server" />  
</div>
```

把 input 的 type 属性改为 submit 或者 reset,就可以分别创建“提交”和“重置”按钮。

(2)在页面的设计视图添加。在工具箱中的 HTML 选项卡中,将如图 5-6 所示的 HtmlInputButton 对象中的某一个拖动到页面中(或双击),便可在页面上添加一个 HtmlInputButton 对象。

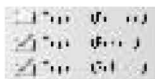


图 5-6 HtmlInputButton 对象

图 5-7 中的三个对象从上而下分别是普通按钮、提交按钮及重置按钮。默认情况下,它们是普通的 HTML 客户端控件,可通过右击控件,在弹出的快捷菜单中选择“作为服务器控件运行”命令使之成为服务器端控件,如图 5-7 所示。



图 5-7 设置 HtmlInputButton 对象作为服务器控件运行

2) 属性和事件

HtmlInputButton 控件的常用属性见表 5-1。

表 5-1 HtmlInputButton 控件的常用属性

属性名称	含 义
EnableViewState	获取或设置一个值,该值指示服务器控件是否向发出请求的客户端保持自己的视图状态以及它所包含的任何子控件的视图状态
Style	获取 HTML 服务器控件的所有级联样式表 (CSS) 属性的集合
Name	获取或设置 HtmlInputControl 的唯一标识符名
Value	获取或设置 HtmlInputControl 显示的文字
ServerClick 事件	当在 Web 页上单击 HtmlInputButton 控件时发生

3)应用

本应用所要实现的效果如图 5-8 所示,当用户单击“显示时间”按钮时,页面上将显示当前时间。

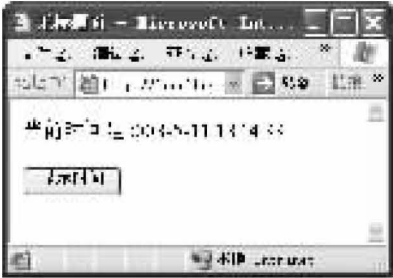


图 5-8 HtmlInputButton 的应用效果

步骤如下：

(1)新建网站 EX_HtmlInputButton, 在页面上添加一个 HtmlInputButton 控件,右击该按钮对象,在弹出的快捷菜单中选择“作为服务器控件运行”命令。

(2)右击该按钮对象,从弹出的快捷菜单中选择“属性”命令,在打开的属性对话框中修改其 id 属性为 ButtonTime,修改其 value 属性值为“显示时间”。

(3)双击按钮打开代码页,并编辑按钮的单击事件,代码如下：

```
protected void ButtonTime_ServerClick(object sender, EventArgs e)
{
    string currentTime = System.DateTime.Now.ToString();
    Page.Response.Write("当前时间是" + currentTime);
}
```

(4)完成后保存,按 Ctrl+F5 组合键运行程序。

2. 单行文本框 HtmlInputText

当需要在页面上输入单行信息时可以使用 HtmlInputText 对象。

在页面的源视图中可以通过添加下面的代码实现。例如,添加一个 ID 为“Text1”的 HtmlInputText 对象,代码如下：

```
<div>
    <input id=" Text1" type=" text" runat="server" />
</div>
```

把其 type 属性改为 password 就可以实现密码框控件。

1) 属性和事件

HtmlInputText 控件的常用属性及含义见表 5-2。

表 5-2 HtmlInputText 控件的常用属性

属性名称	含 义
MaxLength	获取或设置可以在文本框中输入的最大字符数
Size	获取或设置文本框的宽度
ServerChange 事件	在服务器上更改文本框中的值时发生

2) 应用

本应用所要实现的功能如图 5-9 所示,当用户单击“确定”按钮时,页面上将显示用户输入的用户名和密码。

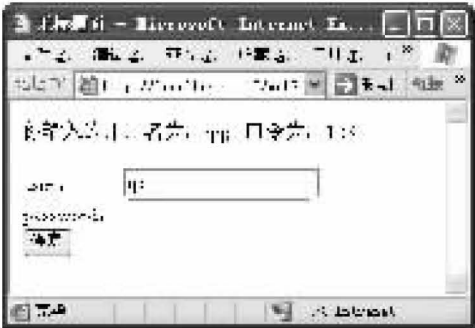


图 5-9 HtmlInputText 的应用效果

步骤如下:

(1)新建网站 EX_HtmlInputText, 在 default.aspx 页面上添加两个 InputText 对象和一个 InputButton 对象。分别右击这些对象,在弹出的快捷菜单中选择“作为服务器控件运行”命令。

(2)右击该按钮对象,在弹出的快捷菜单中选择“属性”命令,在打开的属性对话框中分别修改其 id 为 textInput、passwordInput 和 Button1。控件的 HTML 代码如下:

```
<div>
    <input id="TextInput" type="text" runat="server" />
    <input id="PasswordInput" type="password" runat="server" />
    <input id="Button1" type="button" value="确定" runat="server"
</div>
```

(3)双击按钮对象打开代码页,编辑按钮的单击事件,代码如下:

```
protected void Button1_ServerClick(object sender, EventArgs e)
{
    Page.Response.Write("你输入的用户名为:" + TextInput.Value);
    Page.Response.Write(" ; 口令为:" + PasswordInput.Value);
}
```

(4)完成后保存,按 Ctrl+F5 组合键运行程序。

3. TextArea 控件

当需要在页面上输入多行信息时可以使用多行文本框 HtmlTextArea 对象。例如,添加一个 ID 为 TextArea1,高和宽分别为 6mm 和 20mm 的 HtmlTextArea 对象,代码如下:

```
<div>
    <textarea id="TextArea1" runat="server" cols="20" rows="6"></textarea>
</div>
```

1)属性

HtmlTextArea 的许多属性与 HtmlInputText 相似,这里介绍其独有的属性,见表 5-3。

表 5-3 HtmlTextArea 的部分属性

属性名称	含 义
Cols	获取或设置 HtmlTextArea 控件的宽度
Rows	获取或设置 HtmlTextArea 控件的高度

2)应用

HtmlTextArea 的应用与 HtmlInputText 类似,效果如图 5-10 所示。

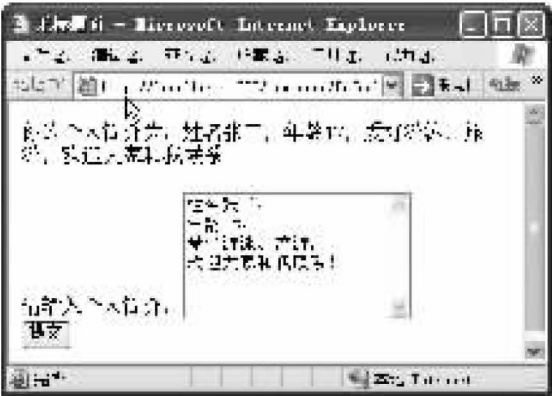


图 5-10 HtmlTextArea 的应用效果

步骤如下:

(1)新建网站 EX_HtmlTextArea, 在页面上添加一个 TextArea 对象和一个 InputButton 对象。分别右击这些对象,在弹出的快捷菜单中选择“作为服务器控件运行”命令。

(2)右击该按钮对象,在弹出的快捷菜单中选择“属性”命令,在打开的属性对话框中分别修改其 id 为 textArea1、Button1。textArea1 控件的 rows 属性值为 6,控件的 HTML 代码如下:

```
<div>
    <textarea id="TextArea1" runat="server" rows="6"></textarea>
    <input id="Button1" runat="server" type="button" value="提交" />
</div>
```

(3)双击 InputButton 对象打开代码页,编辑其单击事件的触发方法,代码如下:

```
protected void Button1_ServerClick(object sender, EventArgs e)
{
```

```

Page.Response.Write("你的个人简介为:" + TextArea1.Value);
}

```

(4)完成后保存,按 Ctrl+F5 组合键运行程序。

4. 隐藏文本框 HtmlInputHidden

当需要在页面上实现一个隐藏的输入框时可以使用 HtmlInputHidden 对象,它是存在的,但不在窗体上显示,经常与 HtmlInputText 和 HtmlInputButton 一起使用,用于存储临时信息或中间结果。例如,添加一个 ID 为“Hidden1”的 HtmlInputHidden 对象,其代码如下:

```

<div>
    <input id="Hidden1" type="Hidden" runat="server" />
</div>

```

HtmlInputHidden 的常用属性是 value 属性,用来获取或设置 HtmlInputHidden 上显示的文字。

下面学习 HtmlInputHidden 控件的应用。本应用所要实现的功能如图 5-11 所示,用户连续输入一系列数字,每次输入后单击“+”按钮,输入完成后单击“结果”按钮,浏览器上将显示所有的输入及结果构成的等式,如“1+2+4+6=13”。



图 5-11 HtmlInputHidden 的应用效果

步骤如下:

(1)新建网站 EX_HtmlInputHidden,在页面上添加两个 HtmlInputButton 对象、一个 HtmlInputText 对象(用来接受用户输入的数据)和两个 HtmlInputHidden 对象(分别用来临时存储累加的数字和最终结果)。定义控件的 HTML 代码如下:

```

<div>
    <input id="Text1" type="text" runat="server" />
    <input id="Button1" type="button" value="+" runat="server" />
    <input id="Button2" type="button" value="结果" runat="server" />
    <input id="HiddenNumbers" type="hidden" runat="server" />
    <input id="HiddenResult" type="hidden" runat="server" />
</div>

```

(2)双击“+”按钮打开代码页,编辑按钮“+”的单击事件触发方法,代码如下:

```

protected void Button1_ServerClick(object sender, EventArgs e)
{
    //用 HiddenNumbers 保存所累加的数
    if (HiddenNumbers.Value == "")

```

```
        HiddenNumbers.Value += Text1.Value;
    else
        HiddenNumbers.Value += "+" + Text1.Value;
    //用 HiddenResult 保存中间结果
    int temp = 0;
    if (HiddenResult.Value == "")
        temp = Convert.ToInt32(Text1.Value);
    else
        temp=Convert.ToInt32(Text1.Value) + Convert.ToInt32(HiddenResult.Value);
    HiddenResult.Value = temp.ToString();
}
```

(3)双击“结果”按钮,编辑其单击事件的触发方法,代码如下:

```
protected void Button2_ServerClick(object sender, EventArgs e)
{
    Page.Response.Write(HiddenNumbers.Value + "=" + HiddenResult.Value);
    HiddenNumbers.Value = "";
    HiddenResult.Value = "";
}
```

(4)完成后保存,按 Ctrl+F5 组合键运行程序。

5. 文件上传框 HtmlInputFile

当用户需要从浏览器向服务器上传文件时,可以使用 HtmlInputFile 对象。上传的文件可以是二进制文件或文本文件,这时需将包含控件的表单的 Enctype 属性设置为“multipart/form-data”。

例如,添加一个 ID 为“File1”的 HtmlInputFile 对象,代码如下:

```
<div>
    <input id="File1" type="file" runat="server" />
</div>
```

需要注意的是,HtmlInputFile 控件本身并没有把文件发送到服务器的内置方法,需要结合使用 PostedFile 属性中包含的 System. web. PostedFile 对象的 SaveAs 方法。

1)属性

HtmlInputFile 控件的常用属性及含义见表 5-4。

表 5-4 HtmlInputFile 的属性

属性名称	含 义
Accept	获取或设置用逗号分隔的文件类型列表,用于约束用户可以上传的文件类型
Value	获取或设置所要上传的文件的带路径名称

2)HtmlInputFile 对象的应用

本示例所要实现的功能如图 5-12 所示,用户选择一个待上传的文件,然后将其上传到服务器上程序所在目录的 Upload 文件夹中,步骤如下:

(1)新建网站 Ex_HtmlInputFile,并在项目文件夹中新建一个 Upload 文件夹,用于存储上

传的文件。在默认的 Default 页面上添加一个 HtmlInputFile 对象和一个 HtmlInputButton 对象。定义控件的 HTML 代码如下：

```
<div>  
    <input id="File1" runat="server" type="file" />  
    <input id="Button1" runat="server" type="button" value="上传" />  
</div>
```

(2) 双击“上传”按钮，其单击事件的代码如下：

```
protected void Button1_ServerClick(object sender, EventArgs e)  
{  
    //获取文件名  
    ArrayList arrFileName = new ArrayList();  
    arrFileName.AddRange(File1.Value.Split('\\'));  
    string fileName = arrFileName[arrFileName.Count - 1].ToString();  
    //获取文件所要保存的路径  
    string uploadPath = Page.Request.PhysicalApplicationPath + "upload\\";  
    //上传文件  
    File1.PostedFile.SaveAs(uploadPath + fileName);  
    Page.Response.Write("上传成功!");  
}
```

(3) 按 Ctrl+F5 组合键运行程序，到程序所在路径的 Upload 文件夹下，将会发现所上传的文件。

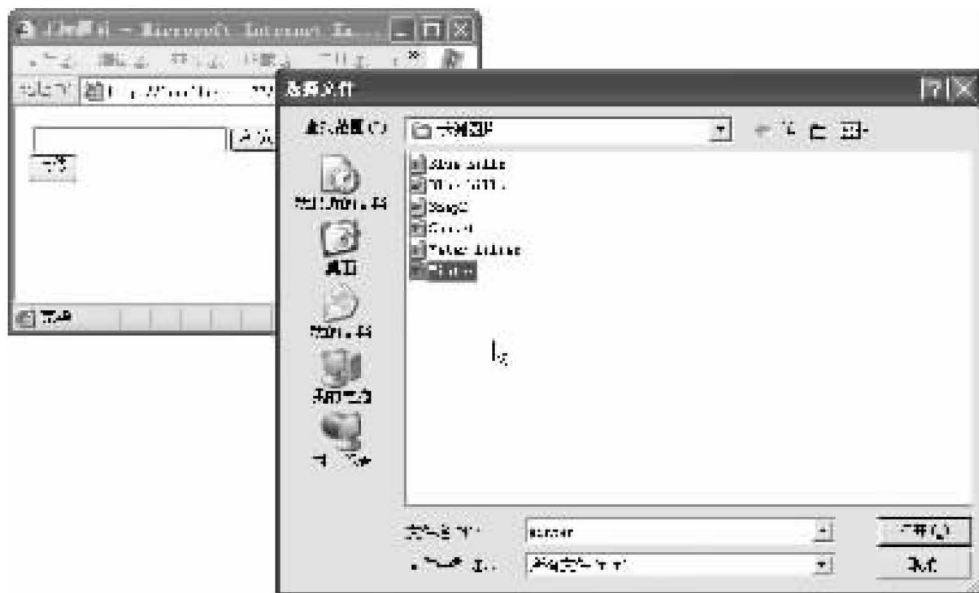


图 5-12 HtmlInputFile 的应用效果

6. 复选框 HtmlInputCheckBox

HtmlInputCheckBox 控件实现了复选框功能,当用户在页面上是否选中了某个选项,可以使用这个对象。

例如,添加一个 ID 为“CheckBox1”的 HtmlInputCheckBox 对象,代码如下:

```
<div>
    <input id=" CheckBox1" type=" CheckBox" runat="server" />
</div>
```

1)属性

这里介绍其特有的属性 Checked,用来获取或设置复选框是否被选中,取值为 true 或 false。

2)应用

本应用所要实现的功能如图 5-13 所示,用户可在多个爱好中进行选择,单击“提交”按钮后,页面上将显示用户所选择的内容。



图 5-13 HtmlInputCheckBox 的应用效果

步骤如下:

(1)新建网站 Ex_HtmlInputCheckBox,在 Default 页面上添加四个 HtmlInputCheckBox 对象和一个 HtmlInputButton 对象。定义控件的 HTML 代码如下:

```
<div>
    <input id="Checkbox1" type="checkbox" runat="server" />游泳
    <input id="Checkbox2" type="checkbox" runat="server" />篮球
    <input id="Checkbox3" type="checkbox" runat="server" />阅读
    <input id="Checkbox4" type="checkbox" runat="server" />电影
    <input id="Button1" type="button" value="提交" runat="server" />
</div>
```

(2)双击“提交”按钮,其单击事件的代码如下:

```
protected void Button1_ServerClick(object sender, EventArgs e)
{
    string result = "您选择的爱好有:";
    if (Checkbox1.Checked == true) result += "游泳";
    if (Checkbox2.Checked == true) result += "篮球";
    if (Checkbox3.Checked == true) result += "阅读";
    if (Checkbox4.Checked == true) result += "电影";
}
```

```

    Response.Write(result);
}

```

这段代码依次根据复选框的 Checked 属性判断用户作出的选择,用 Result 字符串进行记录,最后输出了结果。

(3)完成后保存,按 Ctrl+F5 组合键运行程序。

7. 单选按钮 HtmlInputRadioButton

单选按钮 HtmlInputRadioButton 与复选框 HtmlInputCheckBox 控件相比,单选按钮 HtmlInputRadioButton 也实现了选择功能,但对于同一组中的多个单选按钮,用户仅能选中其中一个。

例如,添加一个 ID 为“RadioButton1”的 HtmlInputRadioButton 对象,代码如下:

```

<div>
    <input id="RadioButton1" type="radio" runat="server" />
</div>

```

1) 属性

HtmlInputRadioButton 最常用的属性也是 Checked,指示是否被选中。HtmlInputRadioButton 通过 name 属性进行分组,具有相同 name 的多个单选按钮只能被选中一个,即如果其中一个的 Checked 属性被设置为 true,其余的该属性将自动变为 false。

2) 应用

本应用所要实现的功能如图 5-14 所示,用户选择自己的性别,单击“显示”按钮后,页面上将显示用户所选择的内容。

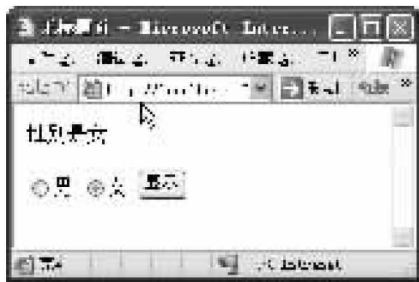


图 5-14 HtmlInputRadioButton 的应用效果

步骤如下:

(1)新建网站 Ex_HtmlInputRadioButton,在默认的 Default 页面上添加两个 HtmlInputRadioButton 对象和一个 HtmlInputButton 对象。定义控件的 HTML 代码如下:

```

<div>
    <input id="RadioMale" type="radio" name="RadioGroup" runat="server" />男
    <input id="RadioFemale" type="radio" name="RadioGroup" runat="server" />女
    <input id="Button1" type="button" value="显示" runat="server" />
</div>

```

(2)双击“提交”按钮,编辑其单击事件的代码如下:

```
protected void Button1_ServerClick(object sender, EventArgs e)
```

```
{
    if (RadioMale.Checked == true)
        Response.Write("性别是男");
    else
        Response.Write("性别是女");
}
```

(3)完成后保存,按 Ctrl+F5 组合键运行程序。

5.3.3 Select 控件

当用户需要在多个选项中作出选择时,除了使用前面介绍过的 HtmlInputCheckBox 和 HtmlInputRadioButton 对象之外,还可以使用 HtmlSelect 控件,即列表框或下拉列表框控件。

例如,添加一个 ID 为“Select1”的 HtmlSelect 对象,代码如下:

```
<div>
    <select id="Select1" runat="server" >
        <option>选项 1</option>
        <option>选项 2</option>
    </select>
</div>
```

每个列表框或下拉列表框都具有多个 option 选项,若要设置其文本,可修改该项的 ListItem. Text 属性,或直接将文本放入<option>和</option>中间。

1)属性

HtmlSelect 控件用作列表框还是下拉框控件,由其 Multiple 属性决定。如果允许多重选择可将其 Multiple 属性设置为 true 或将 Size 属性设置为大于 1 的值,控件将显示为列表框。Select 控件的常用属性及含义见表 5-5。

表 5-5 HtmlSelect 的常用属性

属性名称	含 义
DataMember	获取或设置要绑定到 HtmlSelect 控件的数据集
DataSource	获取或设置要绑定到 HtmlSelect 控件的信息源
DataTextField	获取或设置数据源中要绑定到 HtmlSelect 控件中各项的 ListItem. Text 属性的字段
DataValueField	获取或设置数据源中要绑定到 HtmlSelect 控件中各项的 ListItem. Value 属性的字段
Items	获取一个集合,该集合包含在 HtmlSelect 控件中列出的项
SelectedIndex	属性设置为具有指定值的列表中的第一个项的索引
Value	获取 HtmlSelect 控件中选定项的值
Size	获取或设置 HtmlSelect 控件的高度(以行为单位)
SelectedIndex	获取或设置 HtmlSelect 控件中的选定项的序号索引

其中前四个属性将结合 ADO.NET 中的数据对象,根据从数据库中查询出的结果显示。本书将在后续章节中讲述。

2) 应用

本应用所要实现的功能如图 5-15 所示,用户选择自己的性别(单选)和兴趣信息(多选),单击“提交”按钮后,页面上将显示用户所选择的内容。



图 5-15 HtmlSelect 的应用效果

步骤如下:

(1)新建网站 Ex_HtmlSelect,在默认的 Default 页面上添加两个 HtmlSelect 对象和一个按钮对象,“性别”下拉列表框的 Multiple 属性值为 false,“兴趣”下拉列表框的 Multiple 属性值为 true。定义控件的 HTML 代码如下:

```
<div>
    性别:<select id="Select1" runat="server">
        <option>男</option>
        <option>女</option>
    </select>
    兴趣:<select id="Select2" multiple="true" runat="server">
        <option>篮球</option>
        <option>舞蹈</option>
        <option>旅游</option>
        <option>钢琴</option>
        <option>电影</option>
    </select>
    <input id="Button1" type="button" value="提交" runat="server" />
</div>
```

(2)双击“提交”按钮,其单击事件的代码如下:

```
protected void Button1_ServerClick(object sender, EventArgs e)
{
    //根据“性别”下拉列表框得到性别信息
    string str = "您的性别为:";
```

```

for (int i = 0; i <= Select1.Items.Count - 1; i++)
{
    if (Select1.Items[i].Selected)
    {
        str += Select1.Items[i].Text + "<br>";
        break;    //由于只能选择一项,因此可以跳出循环
    }
}
//根据“兴趣”列表框得到兴趣信息
str += "您的兴趣有:";
for (int i = 0; i <= Select2.Items.Count - 1; i++)
{
    if (Select2.Items[i].Selected)
        str += Select2.Items[i].Text + " ";
}
//输出
Response.Write(str);
}

```

(3)完成后保存,按 Ctrl+F5 组合键运行程序。

5.3.4 Table 控件

Table 控件属于容器控件。所谓容器控件,是指其内部可以包含控件或其他内容的控件,常见的有表单(Form)和表格(Table)。此类控件的特征在于其 HTML 标记具有结束标记,如<Form></Form>。

网页元素的布局是非常重要的工作,为了将各个控件、图片合理放置,常常需要使用 HTMLTable 控件。每一个 HTMLTable 表格都具有行(TR)和列(TD)。

1) 创建 HTMLTable 对象

创建 HTMLTable 对象有两种方法:在工具箱的 HTML 选项卡中通过鼠标拖动或双击添加。默认的表格是 3×3 结构。在页面的源视图中,通过<table>标记来实现。

例如,添加一个 2×3 的表格,可以通过下面的代码实现。

```

<div>
    <table style="width: 271px; height: 123px" id="TABLE1" runat="server">
        <tr>
            <td></td>
            <td></td>
            <td></td>
        </tr>
        <tr>
            <td></td>
            <td></td>
            <td></td>
        </tr>
    </table>

```

```
</tr>
</table>
</div>
```

作为一个容器控件,HTMLTable 控件其内部可以定义新的控件如输入控件、其他容器控件或另一个 HTMLTable 控件。

2)属性

HTMLTable 控件最常用的属性是能控制其外观的几个属性,见表 5-6。

表 5-6 HTMLTable 控件的常用属性

属性名称	含 义
Align	获取或设置 HtmlTable 控件相对于网页上其他元素的对齐方式
Border	获取或设置 HtmlTable 控件的边框宽度(以像素为单位)
CellPadding	获取或设置 HtmlTable 控件中单元格内容和单元格边框之间的间距(以像素为单位)
CellSpacing	获取或设置 HtmlTable 控件中相邻单元格之间的间距(以像素为单位)
Height	获取或设置 HtmlTable 控件的高度
Rows	获取包含 HtmlTable 控件中所有行的 HtmlTableRowCollection 集合
Width	获取或设置 HtmlTable 控件的宽度
Visible	HtmlTable 控件边框是否可见

3)应用

本应用结合上面介绍的 HTMLInputButton 控件和 HTMLInputText 控件,将它们放入 HTMLTable 控件完成功能,所要实现的效果如图 5-16 所示。



图 5-16 HTMLTable 的应用效果

步骤如下：

- (1)新建网站 Ex_HtmlHTMLTable,在页面上添加一个 HtmlTable 对象,具有 2×2 结构。其 Border 属性值为 5,Visible 属性值为 true,enableviewstate 属性值为 true。
- (2)在 HtmlTable 控件的单元格内放入两个 HTMLInputButton 控件和一个 HTMLInputText 控件,分别实现“显示时间”和“显示姓名”功能。本例与前面例子的区别在于输出

信息不在页面上,而在表格中。为了实现这个功能,首先把需要显示内容的单元格变成服务器端控件,然后在输出时对其 InnerText 属性赋值。以“显示当前时间”按钮的单击事件方法为例,代码如下:

```
protected void Button1_ServerClick(object sender, EventArgs e)
{
    string currentTime = System.DateTime.Now.ToString();
    td1.InnerText = "当前时间是" + currentTime;
}
```

其中,td1 位所要显示文字的单元格 ID。“显示姓名”按钮的单击事件方法与前面类似,不再赘述。

(3)完成全部程序后保存,按 Ctrl+F5 组合键运行程序。

5.3.5 Image 控件

HtmlImage 控件可以在 Web 页上显示图像,并可以通过程序来更改图像、控制图像大小及图像对齐方式等。

1)创建 HtmlImage 对象

创建 HtmlImage 对象有两种方法:在工具箱的 HTML 选项卡中通过鼠标拖动或双击添加,并可通过 src 属性选择要显示的图像。在页面的源视图中,可以通过下面的代码实现:

```
<div>
    
</div>
```

2)属性

HTMLImage 控件的常用属性及含义见表 5-7。

表 5-7 HTMLImage 控件的常用属性

属性名称	含 义
align	获取或设置图像相对于其他网页元素的对齐方式
alt	获取或设置在图像不可用或当前正在下载且尚未完成的情况下浏览器显示的替换标题
border	获取或设置图像框的宽度
height/width	获取或设置图像的高度/宽度
src	获取或设置要显示的图像文件的源
visible	获取或设置一个值,该值指示服务器控件是否作为 UI 呈现在页上

3)应用

本应用所要实现的效果如图 5-17 所示,用户在下拉列表框中选择不同的背景,页面就显示不同的图片。

步骤如下:

(1)新建网站 Ex_HtmlImage,在程序所在目录 picture 下放置四幅图片:Winter.jpg、lilies.jpg、Sunset.jpg 和 Blue hills.jpg。在默认页面上添加一个 HTMLImage 控件、一个 HTMLSelect 控件和一个 HTMLInputButton 控件。其 HTML 代码如下:

```
<div>
    
```

```

<select id="Select1" runat="server">
    <option>Winter</option>
    <option>lilies</option>
    <option>Sunset</option>
    <option>Blue hills</option>
</select>
<input id="Button1" type="button" value="图片变换" runat="server" />
</div>

```

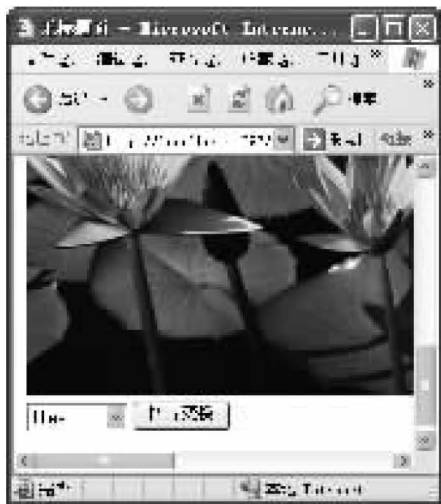


图 5-17 HtmlImage 的应用效果

(2) 双击“图片变换”按钮,编辑单击事件方法,代码如下:

```

protected void Button1_ServerClick(object sender, EventArgs e)
{
    //循环检查用户的选择,并根据用户的选择得到图像文件名
    string str = "";
    for (int i = 0; i <= Select1.Items.Count - 1; i++)
    {
        if (Select1.Items[i].Selected)
        {
            str += Select1.Items[i].Text + ".jpg";
            break;
        }
    }
    //改变图像来源
    IMG1.Source = ".\\Picture\\" + str;
}

```

(3) 完成程序后保存,按 Ctrl+F5 组合键运行程序。