

本章学习目标

- 熟练掌握 ASP.NET 中视图区域控件的使用方法
- 熟练掌握 ASP.NET 中文件上传控件的使用方法
- 熟练掌握 ASP.NET 中日历控件的使用方法
- 了解 ASP.NET 中广告控件的使用方法
- 了解 ASP.NET 中向导控件的使用方法

本章首先介绍 ASP.NET 服务器部分高级控件应用,对 ASP.NET 中的视图区域控件进行详细讲解,然后通过应用讲解文件上传控件、日历控件和向导控件,最后添加.xml广告文件,并在广告控件中进行应用。

3.1 简介视图区域控件

视图区域控件是页面分组、布局、外观设置常使用的控件,通过该控件可以将部分控件 作为一个单元进行管理,也可以为其他控件提供容器,或者为页面上特定区域设置独特外 观。视图区域控件主要包含面板控件(Panel)、占位符控件(PlaceHolder)、视图控件(View) 和多视图控件(MultiView)等,本节将对这几种控件进行详细讲解。

3.1.1 面板控件 Panel

□\$***** ●. UI. □***** 子校 ^{扫一扫}

Panel 控件又称面板控件,在工具箱中图标为"**Panel**",封装在 System. Web. UI. Control. WebControl 命名空间的 Panel 类中。可以在其内部动态地向该控件添加其他子控件和标签,也可以使用该控件将页面拆分为独立排版显示的若干部分。Panel 控件的常用属性如表 3.1 所示。

属性名	说明
BackImageUrl	获取或设置面板控件的背景图像 URL
DefaultButton	获取或设置面板控件的默认按钮
GroupingText	获取或设置面板控件中包含的控件组的标题
Controls	获取面板控件中包含的子控件集合

表 3.1 Panel 控件的常用属性

说明:

- (1) 默认按钮属性是指焦点在面板控件内按下 Enter 键时或鼠标"单击"时处理。
- (2) Controls 属性中包含 Add()方法,通过该方法可以动态地向面板控件中添加新的

Pane1控件中添加元素 添加□■个标签控件 添加□■个文本框控件 動态添加控件 □显示Pane1 」动态添加的控件

图 3.1 Panel 控件应用页面布局

子控件。

【示例 3-1】 在 E 盘的 ASP.NET 项目代码目录 中创建 chapter3 子目录,将其作为网站根目录,创建 名为 example3-1 的页面,添加面板按钮及其他控件, 实现动态添加控件以及隐藏 Panel 功能。

(1) 在页面 example3-1 中添加相应控件,具体如 图 3.1 所示。

(2)设置控件的相关属性,如下列源代码。

```
<form id="form1" runat="server">
     在 Panel 控件中添加元素<br/>>
       添加<asp:DropDownList ID="ddlLbl" runat="server">
          <asp:ListItem>0</asp:ListItem>
          <asp:ListItem>1</asp:ListItem>
          <asp:ListItem>2</asp:ListItem>
          <asp:ListItem>3</asp:ListItem>
          <asp:ListItem>4</asp:ListItem>
       </asp:DropDownList>
       个标签控件<br/>>
       添加<asp:DropDownList ID="ddlTxt" runat="server">
          <asp:ListItem>O</asp:ListItem>
          <asp:ListItem>1</asp:ListItem>
          <asp:ListItem>2</asp:ListItem>
          <asp:ListItem>3</asp:ListItem>
          <asp:ListItem>4</asp:ListItem>
       </asp:DropDownList>
       个文本框控件<br/>>
       <asp:Button ID="btnAdd" runat="server" Text="动态添加控件" OnClick="
btnAdd Click"/>
      <asp:CheckBox ID="chkShow" runat="server" AutoPostBack="True" Text="显
ar{\pi} Panel" OnCheckedChanged="chkShow CheckedChanged"/>
      <asp:Panel ID="Panel1" runat="server" GroupingText="动态添加的控件">
       </asp:Panel>
   </form>
```

(3) 为控件添加事件代码如下。

3.8

```
protected void btnAdd Click(object sender, EventArgs e)
{
   //设置变量 m 的值为 ddlLbl 控件中选中的数值
   int m=int.Parse(ddlLbl.SelectedItem.Value);
   //循环次数为要添加的 Label 控件的个数
   for (int i=1; i <=m; i++)</pre>
   {
      //实例化一个 Label 控件对象 1b1
      Label lbl=new Label();
      //赋值 1b1 对象的 ID 属性为:"1b1"+i
      lbl.ID="lbl"+i;
      //赋值 1b1 对象的 Text 属性为:"标签控件"+i
      lbl.Text="标签控件"+i.ToString()+"<br>";
      //添加 lbl 控件到 Panel 中
      Panel1.Controls.Add(lbl);
   }
   //设置变量 n 的值为 ddlTxt 控件中选中的数值
   int n=int.Parse(ddlTxt.SelectedItem.Value);
   //循环次数为要添加的 TextBox 控件的个数
   for (int i=1; i<=n; i++)</pre>
   {
      //实例化一个 TextBox 控件对象 txt
      TextBox txt=new TextBox();
      //赋值 txt 对象的 ID 属性为:"txt"+i
      txt.ID="txt"+i;
      //赋值 txt 对象的 Text 属性为:"文本框控件"+i
      txt.Text="文本框控件"+i.ToString();
      //添加 txt 控件到 Panel 中
      Panel1.Controls.Add(txt);
   }
}
protected void chkShow CheckedChanged(object sender, EventArgs e)
{
   //如果当前状态是没选中
   if (chkShow.Checked)
    ł
                                  //就把 Panel 选中
      Panel1.Visible=true;
      chkShow.Text="显示 Panel";
                                  //CheckBox1 的文本为"显示 Panel"
   }
   else
   {
      Panel1.Visible=false;
      chkShow.Text="隐藏 Panel";
    }
}
```

(4)网站运行后,根据列表数值,动态向面板控件中加相应数量的标签和文本框控件, 单击复选按钮可以显示或隐藏面板控件,部分功能如图 3.2 所示。

3.1.2 占位符控件 PlaceHolder

PlaceHolder 控件又称占位符控件,在工具箱中图标为" ☑ PlaceHolder ",封装在 System.Web.UI.Control.WebControl 命名空间的 PlaceHolder 类中。该控件的功能与面板

	× 🗅	- 6∂ ☆	× \$\$ @
Panet经件甲添加元素 添加3 个标签控件 添加2 个文本框控件			
<u>动态添加控件</u> □显示Panel 动态添加的控件 标签控件1 标签控件2 标签控件2			
小亚注叶3 文本框控件1 文本框控件2		෪ 100%	•:

图 3.2 Panel 控件应用页面演示

控件的功能类似,也可以作为页面内的一个容器动态地向其内部添加其他控件,不同的是, PlaceHolder 控件没有基于 HTML 的输出,只是为其他控件标记一个位置。

【示例 3-2】 在 E 盘 ASP.NET 项目代码的 chapter3 目录下创建名为 example3-2 的网页,分别向 Panel 控件和 PlaceHolder 控件内动态添加控件,对比 HTML 源代码的差异。 (1) 在页面中添加相应控件,具体如图 3.3 所示。

P1a< [P1	ceHolden LaceHold	·控件: ler ″Pl	laceHo	lder	1″]	 	 	
Pane	ョ1控件:					 	 	
添	加控件							

图 3.3 PlaceHolder 控件应用页面布局

(2) 设置控件的相关属性,如下列源代码。

```
<form id="form1" runat="server">
<div>
PlaceHolder 控件:<br/><asp:PlaceHolder ID="PlaceHolder1" runat="server"></asp:PlaceHolder>
<br/><br/>Panel 控件:<asp:Panel ID="Panel1" runat="server">
</asp:Panel>
<br/><br/><asp:Button ID="btnAdd" runat="server" Text="添加控件" OnClick="btnAdd_
Click"/>
</div>
</form>
```

(3) 添加事件代码,如下所示。

```
protected void btnAdd_Click(object sender, EventArgs e)
{
    Button btn1=new Button();
```

```
₿₿₿₫⁄2
   btn1.Text="PlaceHolder内的按纽";
   btn1.ID="btn1";
   PlaceHolder1.Controls.Add(btn1);
   Button btn2=new Button();
  btn2.Text="Panel内的按纽";
  btn2.ID="btn1";
  Panel1.Controls.Add(btn2);
}
```

Web高级控件

(4) 网站运行后单击"添加控件"按钮,如图 3.4 所示。

← ⊕ + @ https://loc ♀ + ♣ ℃ @ localhost	- × [*	_ ☆ ☆	× © @
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)			
PlaceHolder控件: PlaceHolder内的按纽			
Panel控件: Panel内的按组			
添加控件			
		€, 100%	•

图 3.4 PlaceHolder 控件应用网页演示

(5) 对比网页运行后的 HTML 源代码,比较 PlaceHolder 控件和 Panel 控件的客户端 脚本,具体如图 3.5 所示。

	П	i G. G. C. V. O. X 🔠 🍺	
-	ex	ample3-2.aspx 🗙 🔄 🖏 🕼 🕛	
	1		~
	2		
	з	html	
	4		
	5	<html xmlns="http://www.w3.org/1999/xhtml"></html>	
	6	<head><meta content="text/html; charset=utf-8" http-equiv="Content-Type"/><title></title></head>	
	7		
	8		
	9	 body>	
	10	<form action="./example3-2.aspx" id="form1" method="post"></form>	
	11	<div class="aspNetHidden"></div>	
	12	<input aspnethidden"="" id="_VIEWSTATE" name="_VIEWSTATE" type="hidden" value="18AQ+Y5Wnm68Tj/xdPmZpVzcpUrcUTQGiNQptM68KQF2HieNBPFfSvu1XE</td><td>YR</td></tr><tr><td></td><td>13</td><td></div></td><td></td></tr><tr><td></td><td>14</td><td></td><td></td></tr><tr><td></td><td>15</td><td><div class="/>	
	16		
	17	<pre>cinput type="hidden" name="VIEWSTATEGENERATOR" id="VIEWSTATEGENERATOR" value="71166648" /></pre>	
	18	<pre><input id="even(valida(i0N" name="even(valida(i0N" panel1"="" type="hidden" value="+Nrnow81+/skszsanPBYM5/odxQ804Jryd2C4Wbp/+</pre></td><td>. 15</td></tr><tr><td></td><td>19</td><td></div</td><td></td></tr><tr><td></td><td>20</td><td></td><td></td></tr><tr><td></td><td>21</td><td>riacenoideriy(); <pre></td><td></td></tr><tr><td></td><td>22</td><td>che /s</td><td></td></tr><tr><td></td><td>24</td><td></td><td></td></tr><tr><td></td><td>25</td><td>マローノン
Panel1谷住: <div id="/></pre>	
	26		
	27	<input id="btn1" name="btn1" type="submit" yalue="Panel内的妹叔"/>	
	28	<td></td>	
	29	<pr></pr> <td></td>	
	30	<input id="btnAdd" name="btnAdd" type="submit" value="添加控件"/>	
	31		
	32		
	33		
	34		~
	35	()	1

图 3.5 PlaceHolder 控件和 Panel 控件的客户端脚本比较

在网页的 HTML 源文件中, Panel 控件有输出客户端脚本, 会产生 DIV 的 HTML 代码, 而 PlaceHolder 控件仅在服务器端起分组的作用, 不会产生额外的 HTML 代码。在页面中使用控件有进行分组的情况时, 如果客户端有对分组进行显示或隐藏, 以及改变颜色等操作需求时, 应使用 Panel 控件, 否则使用 PlaceHolder 控件。



3.1.3 视图控件 View 与多视图控件 MultiView

View 控件又称视图控件,在工具箱中图标为"**p** View",封装在 System. Web. UI. Control. WebControl 命名空间的 View 类中,用于将页面拆分为独立显示的若干部分。视图必须包含在 MultiView 控件内,可以作为其他控件的容器,没有外观,只有被激活时才会显示。

MultiView 控件又称多视图控件,在工具箱中图标为" MultiView",封装在 System. Web.UI.Control.WebControl 命名空间的 MultiView 类中。作为使用 View 控件的必选控件,MultiView 控件实际上是作为一个容器使用。在 MultiView 控件中可以包含若干个 View 控件,通过设置 ActiveViewIndex 属性确定在 MultiView 控件内显示的 View 视图,当 ActiveViewIndex=-1 时,所有 View 视图均不显示。MultiView 控件的常用属性如表 3.2 所示。MultiView 控件的常用事件如表 3.3 所示。

表	3.2	MultiView	控件的	常用属性
---	-----	-----------	-----	------

属性名	说明
ActiveViewIndex	获取或设置 MultiView 控件的 ActiveView 控件的索引

表 3.3 MultiView 控件的常用事件

事件名	说明
ActiveViewChanged	当 MultiView 控件的 ActiveView 控件发生变化时触发

说明:

MultiView 控件内各 View 控件的索引不需要设置,按从上至下位置自动赋值,初始索引值为 0。

【示例 3-3】 在 chapter3 网站根目录下创建名为 example3-3 的网页,页面内包含一个 MultiView 控件和 4 个 View 控件以及若干其他控件,练习使用视图的切换,以及按钮的 CommandName 和 CommanArgument 属性。

(1)在 chapter3 网站根目录上新建"上传文件"文件夹,在"上传文件"文件夹添加卡通数字图片 1.jpg、2.jpg、3.jpg、4.jpg。

(2) 按图 3.6 所示添加相应文本及控件。

(3) 设置控件的相关属性,如源文件所示。

```
<form id="form1" runat="server">
<div>
<asp:RadioButtonList ID="RadioButtonList1" runat="server" AutoPostBack=
```

部3章 Web高级控件

c视图1 c视图2 c试图3 c视图4			
			MultiView1
			View1
	回尾视图	玉一视图	
20 m -	1 2000		View2
PULEI2	上一视图	<u>下一视度</u>	
	1 +0.000		View3
Averal s	<u>{WB</u>		
		_ \/ \	View4
70/1814	上一视图	回自视图	

图 3.6 View 控件和 MultiView 控件应用页面布局

```
"True" OnSelectedIndexChanged= "RadioButtonList1 SelectedIndexChanged"
RepeatDirection="Horizontal">
          <asp:ListItem>视图 1</asp:ListItem>
          <asp:ListItem>视图 2</asp:ListItem>
          <asp:ListItem>视图 3</asp:ListItem>
          <asp:ListItem>视图 4</asp:ListItem>
       </asp:RadioButtonList>
        < asp: MultiView ID="MultiView1" runat="server" ActiveViewIndex="0"</pre>
OnActiveViewChanged= "MultiView1 ActiveViewChanged">
          <asp:View ID="View1" runat="server">
              视图 1       
<asp: LinkButton ID = "LinkButton7" runat = "server" CommandArgument = "View4"</pre>
CommandName="SwitchViewByID">回尾视图</asp:LinkButton>&nbsp;
<asp:LinkButton ID="LinkButton1" runat="server" CommandName="NextView">下一视
图</asp:LinkButton><br/>
<asp:Image ID="Image1" runat="server" ImageUrl="~/上传文件/1.jpg" Width="200px"/>
           </asp:View>
           <asp:View ID="View2" runat="server">
           视图 2       
<asp:LinkButton ID="LinkButton4" runat="server" CommandName="PrevView">上一视
图</asp:LinkButton>&nbsp;
<asp:LinkButton ID="LinkButton2" runat="server" CommandName="NextView">下一视
图</asp:LinkButton><br/>
<asp:Image ID="Image2" runat="server" ImageUrl="~/上传文件/2.jpg" Width="200px"/>
           </asp:View>
```

```
<asp:View ID="View3" runat="server">
视图 3       
<asp:LinkButton ID="LinkButton9" runat="server" CommandName="PrevView">上一视
图</asp:LinkButton>&nbsp;
视图</asp:LinkButton> <br/>
<asp: Image ID="Image3" runat="server" ImageUrl="~/上传文件/3.jpg" Width="
200px"/>
         </asp:View>
         <asp:View ID="View4" runat="server">
视图 4       
<asp:LinkButton ID="LinkButton6" runat="server" CommandName="PrevView">上一视
图</asp:LinkButton>&nbsp;
<asp: LinkButton ID = "LinkButton8" runat = "server" CommandArgument = "View1"</pre>
CommandName="SwitchViewByID">回首视图</asp:LinkButton>&nbsp;<br/>
<asp:Image ID="Image4" runat="server" ImageUrl="~/上传文件/4.jpg" Width="
200px"/>
         </asp:View>
     </asp:MultiView>
    </div>
   </form>
```

(4)为 RadioButtonList1 控件和 MultiView1 控件的事件添加如下代码。



(5)运行网站,可以通过单选按钮列表切换视图,也可以通过每个视图中的链接按钮切换视图,部分运行页面如图 3.7 所示。



图 3.7 View 控件和 MultiView 控件应用网页演示

通过设置按钮控件的 CommandName 和 CommandArgument 属性可以直接实现视图 的切换,CommandName 属性可设置的值如下。

- · PrevView: 表示切换到上一视图(当前视图索引不可为 0)。
- NextView: 表示切换到下一视图(当前视图不可为索引值最大的视图)。
- SwitchViewByID: 表示切换到指定 ID 的视图,具体的 ID 属性值需在 CommandArgument 属性中设置,具体见视图 4"回首视图"按钮中属性值的设置。

3.2 文件上传控件 FileUpload

FileUpload 控件又称文件上传控件,在工具箱中图标为" **1** FileUpload",封装在 System.Web.UI.Control.WebControl 命名空间的 FileUpload 类中,在网页中显示为一个文 本框控件和一个"浏览"按钮的组合,用于在网页中通过单击按钮完成待上传文件的选择。 FileUpload 控件的常用属性如表 3.4 所示。



Web高级控件

ßé

属性名	说 明
FileName	获取客户端使用 FileUpload 控件上传文件的名称
HasFile	获取一个值,该值指示 FileUpload 控件是否包含文件
FileBytes	获取 FileUpload 控件中所包含文件的字节数
PostedFile	获取 FileUpload 控件上传文件的 HttpPostedFile 对象

表 3.4 FileUpload 控件的常用属性

说明:

(1) HasFile 属性用于验证 FileUpload 控件是否包含文件,若返回值为 True,则表示控件包含文件;若返回值为 False,则表示控件不包含文件。

(2) PostedFile 属性包含若干二级属性,常用的属性如下。

- ContentLength 属性: 获取上传文件的大小(以字节为单位)。
- ContentType 属性:获取上传文件的 MIME 类型。
- FileName 属性:获取上传文件的完整路径及文件名。 FileUpload 控件的常用方法如表 3.5 所示。

表 3.5 FileUpload 控件的常用方法

方法名	说明
SaveAs()	将使用 FileUpload 控件上传文件的内容保存到服务器的指定路径

说明:

使用 FileUpload 控件时,可以通过单击"浏览"按钮,在"选择文件"对话框中选择文件, 选择文件后使用 HasFile 属性确定是否选择了文件,调用文件上传控件的 SaveAs()方法执 行文件的上传。 【示例 3-4】 在 chapter3 网站根目录下创建名为 example3-4 的网页,页面内包含一个 文件上传控件和若干其他控件,练习使用文件上传控件的相关属性及事件。

(1) 在 example3-4 网页添加相应文件上传控件和其他控件,具体如图 3.8 所示。



图 3.8 FileUpload 控件应用页面布局

(2) 设置控件相关属性,如源文件所示。

```
<form id="form1" runat="server">
```

(3)为按钮控件的事件添加如下代码。

```
protected void btnOK Click(object sender, EventArgs e)
{
   //判断文件上传控件中是否包含文件
   if (FileUpload1.HasFile)
   {
      //赋值 fileName 变量为文件上传控件中选择的文件名
      string fileName=FileUpload1.FileName;
      //赋值 lastDotIndex 变量为文件名中最后一个'.'的索引值
      int lastDotIndex=fileName.LastIndexOf('.');
      //赋值 lastName 变量为文件的后缀名
      string lastName=fileName.Remove(0, lastDotIndex);
      //判断选择的文件后缀名是否为 .jpg|.bmp|.jpeg|.gif 中的一种
      if (lastName==".jpg" || lastName==".bmp" || lastName==".jpeg" || lastName==".gif")
      {
         //判断选择的文件是否大于 1MB
         if (FileUpload1.PostedFile.ContentLength<1 * 1024 * 1024)
             //"年年年年月月日日时时分分秒秒毫毫毫"格式的时间字符串
             string strTime=DateTime.Now.ToString("yyyyMMddhhmmssfff");
             //赋值 newFileName 变量为原文件名加时间字符串构成的新文件名
             string newFileName=fileName.Insert(lastDotIndex, "("+strTime+")");
             //赋值 path 变量为"上传文件"对应的绝对路径
             string path=Server.MapPath("上传文件");
             //将选中文件保存到指定路径
```

3.章

```
FileUpload1.SaveAs(path+@""+newFileName);
             //赋值 imgUrl 变量为新上传的文件名
             string imgUrl=@"上传文件"+newFileName;
             //为 Image1 控件的 ImageUrl 属性赋值
             Image1.ImageUrl=imgUrl;
             //为 lblInfo 控件的 Text 属性赋值,显示上传文件的文件名、大小、类型属性
             lblInfo.Text="选择的文件名:"+FileUpload1.FileName+"<br>文件大
小:"+ FileUpload1. PostedFile. ContentLength +" b < br > 文件类型:" + FileUpload1.
PostedFile.ContentType;
         }
         else
          {
             lblInfo.Text="图像大小必须小于 1MB";
          }
      }
      else
      1
         lblInfo.Text="请选择正确图像文件(.jpg|.bmp|.jpeg|.gif)";
   }
   else
   {
      lblInfo.Text="请选择图像";
}
```

(4)运行网站,如没选择文件时单击"上传"按钮,会提示"请选择图像";若选择了非图像文件,则提示"请选择正确图像文件(.jpg|.bmp|.jpeg|.gif)";若选择的文件大于 1MB,提示"图像大小必须小于 1MB";若选择小于 1MB 的.gif 图像,则可实现图片文件以新文件名

上传,如" Catjpg",在 image1 控件中显示图像,在 lblInfo 控件中显示相关属性。部分执行效果如图 3.9 所示。

说明:

(1)在上传文件名中添加时间字符串可有效防止文件同名覆盖问题。

(2)采用以文件后缀名进行类型验证的方法,不 会对文件内容进行检测,对于被恶意修改后缀名的文件无法识别判断。

(3)路径中使用到的"@"符可以使后续字符串 内容不做转义字符处理,如@"//"等价于"////"字 符串。

(4) Server.MapPath("上传文件"); Server 内置 对象中的 MapPath()方法可将相对路径转化为实际 物理路径,本书第5章将会详细讲解。

(5)系统默认上传文件大小为 4096KB,如果要 上传超过此大小的文件,会出现错误界面。但可在



图 3.9 FileUpload 控件应用页面演示

web.config 文件中进行配置,设置文件大小。在解决方案中找到配置文件 Web.config,添加如下代码。

configuration>
<system.web></system.web>
<httpruntime executiontimeout="120" maxrequestlength="4096"></httpruntime>

其中,maxRequestLength 属性限制文件上传的大小,以 KB 为单位,默认值为 4096KB, 而最大上限为 2097151KB,大约是 2GB; executionTimeout 属性限制文件上传的时间,以秒(s) 为单位,默认值为 90s,可设置该属性值以延长或缩短上传时间。



Calender 控件又称日历控件,在工具箱中图标为"**Calendar**",封装在 System. Web.UI.Control.WebControl 命名空间的 Calender 类中。用户可通过该日历控件导航到某年的某一天,且具有自动套用格式,方便用户选择使用。Calendar 控件是一个比较复杂的控件,具有大量的编程和格式选项,其常用属性如表 3.6 所示。Calender 控件的常用事件如表 3.7 所示。

属性名	说明
Caption	获取或设置与日历控件关联的标题
SelectedDate	获取或设置选定的在控件中突出显示的特定日期
ShowNextPrevMonth	获取或设置是否允许用户进行月份导航
SelectionMode	获取或设置日历控件的选择模式
VisibleDate	获取或设置日期用于确定日历中显示的月份

表 3.6 Calender 控件的常用属性

说明:

- (1) 通过 SelectionMode 属性可以设置日历控件的选择模式,具体有如下枚举值。
- None: 表示禁用所有日期选择。
- Day: 表示用户可以选择一天,每个日期都将包含带有日期编号的链接。
- DayWeek:表示用户可以选择一天或者一周,除日期编号的链接外,日历的左侧会额外添加一个带有周选择链接的列。
- DayWeekMonth: 表示用户可以选择一天、一周或一月,除日期编号的链接外,日历的左侧会额外添加一个带有周和月选择链接的列。

(2) 如果 SelectionMode 属性值为 DayWeek 或者 DayWeekMonth,则 SelectedDate 属性值为选中日期中的第一天。

	表 3.7 Calender 控件的常用事件
事件名	说明
SelectionChanged	当更改选择的日期时发生

【示例 3-5】 在 E 盘 ASP.NET 项目代码的 chapter3 目录下,创建名为 example3-5 的 网页,实现 Calender 控件选择模式的设置以及日期的选择。

(1) 在页面中添加相应控件,具体如图 3.10 所示。

7月		2017年8月 9月				
周日	周一	周二	周三	周四	周五	周六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9
3 4 5 6 7 8 9 日历选择模式: ^{[不选择} 当前选中的日期是: [1b1Date] 选中的天是: [1b1Day] 选中的月是: [1b1Month] 进句的月是: [1b1Month]						

图 3.10 Calender 控件应用页面布局

(2) 单击 Calender 控件右上角的"票"图标,选择【自动套用格式...】中的"专业型 2",其他控件的属性设置如下列源文件。

```
<form id="form1" runat="server">
    <div>
         < asp: Calendar ID = "Calendar1" runat = "server" BackColor = "White"</pre>
BorderColor="Black" Font-Names="Verdana" Font-Size="9pt" ForeColor="Black"
Height = " 250px " OnSelectionChanged = " Calendar1 SelectionChanged " Width = "
330px" BorderStyle="Solid" CellSpacing="1" NextPrevFormat="ShortMonth">
           <SelectedDayStyle BackColor="#333399" ForeColor="White"/>
           <TodayDayStyle BackColor="#999999" ForeColor="White"/>
           <OtherMonthDayStyle ForeColor="#999999"/>
           <DayStyle BackColor="#CCCCCC"/>
            < NextPrevStyle Font - Size = "8pt" ForeColor = "White" Font - Bold = "
True"/>
            < DayHeaderStyle Font-Bold="True" Height="8pt" Font-Size="8pt"
ForeColor="#333333"/>
           <TitleStyle BackColor="#333399" Font-Bold="True" Font-Size="12pt"
ForeColor= "White" BorderStyle= "Solid" Height= "12pt"/>
       </asp:Calendar>
       <br/>br/>
       日历选择模式:<asp:DropDownList ID="ddlMode" runat="server" AutoPostBack=
"True" OnSelectedIndexChanged="ddlMode SelectedIndexChanged">
           <asp:ListItem Value="None">不选择</asp:ListItem>
           <asp:ListItem Value="Day">天</asp:ListItem>
```

(3)为各控件添加事件代码如下。

```
protected void ddlMode SelectedIndexChanged(object sender, EventArgs e)
{
   switch (ddlMode.SelectedValue)
   {
       case "None":
          Calendar1.SelectionMode=CalendarSelectionMode.None;
          break;
       case "DayWeekMonth":
          Calendar1.SelectionMode=CalendarSelectionMode.DayWeekMonth;
          break;
       case "DayWeek":
          Calendar1.SelectionMode=CalendarSelectionMode.DayWeek;
          break;
       case "Day":
          Calendar1.SelectionMode=CalendarSelectionMode.Day;
          break;
   }
}
protected void Calendar1 SelectionChanged(object sender, EventArgs e)
{
   lblDate.Text =Calendar1.SelectedDate.ToShortDateString();
   lblDay.Text=Calendar1.SelectedDate.Day.ToString() + "H ";
   lblMonth.Text=Calendar1.SelectedDate.Month.ToString()+"月";
   lblYear.Text=Calendar1.SelectedDate.Year.ToString()+"年";
}
```

(4) 网站运行效果如图 3.11 所示。

			2022	年8月			<u>9月</u>		
1	周一	周二	周三	周四	周五	周六	周日		
	<u>25</u>	<u>26</u>	27	<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>		
	1	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	Z		
	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>		
	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>		
	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>		
	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	2	<u>3</u>	<u>4</u>		
i诜:	择模:		日	~					
			(//)						

Web高级控件

3.6

图 3.11 Calender 控件应用页面演示



AdRotator 控件又称广告控件,在工具箱中图标为" AdRotator ",封装在 System. Web.UI.Control.WebControl 命名空间的 AdRotator 类中,用于在页面上显示广告图像序列,并能通过单击广告图片实现链接跳转。AdRotator 控件的常用属性如表 3.8 所示。

表 3.8 AdRotator 控件的常用属性

属性	说明
AdvertisementFile	获取或设置包含 ad 信息的 XML 文件的路径
Target	获取或设置打开 URL 的位置

AdRotator 控件使用 XML 文件存储 ad 信息。XML 文件使用<Advertisements>开始和结束。在<Advertisements>标签内部,有若干个定义每条 ad 的<Ad>标签。<Ad>标签中预定义的元素如表 3.9 所示。

标签名	说明
<imageurl></imageurl>	表示图像文件的路径
<navigateurl></navigateurl>	表示该 ad 时所链接的 URL

表 3.9 <Ad>标签中预定义的元素

	· · · · · · · · · · · · · · · · · · ·
标签名	说 明
<alternatetext></alternatetext>	表示图像的替换文本
<keyword></keyword>	表示 ad 的类别
<Impressions $>$	表示广告显示权重,所有的 Impressions 之和不能超过 2048,000,000-1

(法主

ASP.NET网站设计教程

【示例 3-6】 在 chapter3 网站根目录下创建名为 example3-6 的网页,页面内包含一个 广告控件,练习使用广告控件和广告文件。

(1) 在 chapter3 网站根目录添加"广告文件"文件夹,其中包含方正电脑.gif、广东证券.gif、 雀巢咖啡.gif、招商银行.gif 4 张图片。

(2)在 chapter3 网站根目录添加 XML 文件,并命名为 AdRotatorFile.xml,编写文件内 容如下。

```
<? xml version="1.0" encoding="utf-8" >
<Advertisements>
 < Ad >
   <ImageUrl>广告图片/方正电脑.gif</ImageUrl>
   <NavigateUrl>http://www.founderpc.cn</NavigateUrl>
   <AlternateText>欢迎访问方正电脑</AlternateText>
   <Keyword>方正</Keyword>
   <Impressions>20</Impressions>
 </Ad>
 <Ad>
   <ImageUrl>广告图片/广东证券.gif</ImageUrl>
   <NavigateUrl>http://www.gzs.com.cn</NavigateUrl>
   <AlternateText>欢迎访问广东证券</AlternateText>
   <Keyword>广东证券</Keyword>
   <Impressions>80</Impressions>
 </Ad>
 <Ad>
   <ImageUrl>广告图片/雀巢咖啡.gif</ImageUrl>
   <NavigateUrl>http://www.nescafe.com.cn</NavigateUrl>
   <AlternateText>欢迎访问雀巢咖啡</AlternateText>
   <Keyword>雀巢咖啡</Keyword>
   <Impressions>30</Impressions>
 </Ad>
 <Ad>
   <ImageUrl>广告图片/招商银行.gif</ImageUrl>
   <NavigateUrl>http://www.cmbchina.com</NavigateUrl>
   <AlternateText>欢迎访问招商银行</AlternateText>
   <Keyword>招商银行</Keyword>
   <Impressions>70</Impressions>
 </Ad>
</Advertisements>
```

(3) 在 example3-6 的网页添加 AdRotator 控件,并设置相关属性,如下列源文件。

```
<form id="form1" runat="server">
        < asp: AdRotator ID="AdRotator1" runat="server" AdvertisementFile="~/
AdRotatorFile.xml"/>
</form>
```

Web高级控件

3 é

(4)运行页面,将按照权重随机显示一个广告。广告在每次页面载入时更改,每一个广告出现的频率通过 < Impressions>权重 </ Impressions>属性确定。如本示例中"广东证券"的权重为 80,所有广告的权重和为 200,则每刷新页面 200 次,"广东证券"的广告大约将 以 80 次的频数进行显示。部分页面如图 3.12 所示。



图 3.12 AdRotator 控件应用页面演示



Wizard 控件又称向导控件,在工具箱中图标为"** Wizard",封装在 System. Web.UI.Control.WebControl 命名空间的 Wizard 类中。Wizard 控件提供了一种简单的机制,能够轻松地生成步骤、添加新步骤或重新安排步骤顺序,无须编写代码即可生成线性或非线性的导航,也可实现自定义控件的用户导航。Wizard 控件的常用属性如表 3.10 所示。Wizard 控件的常用事件如表 3.11 所示。

表 3.10	Wizard	控件的常	常用属性

属性名	说明
ActiveStep	获取 WizardSteps 集合中当前显示给用户的步骤
ActiveStepIndex	获取或设置当前向用户显示的步骤
CancelButtonImageUrl	获取或设置为"取消"按钮显示的图像的 URL
CancelButtonText	获取或设置为"取消"按钮显示的文本标题
CancelButtonType	获取或设置呈现为"取消"按钮的按钮类型
CompleteStep	获取对最终用户账户创建步骤的引用
CreateUserButtonText	获取或设置在"创建用户"按钮上显示的文本标题
CreateUserButtonType	获取或设置呈现为"创建用户"按钮的按钮类型

属性名	说明
Email	获取或设置用户输入的电子邮件地址
FinishDestinationPageUrl	获取或设置当用户单击"完成"按钮时将重定向到的 URL
HeaderText	获取或设置为在控件上的标题区域显示的文本标题
Question	获取或设置用户输入的密码恢复确认问题
QuestionRequiredErrorMessage	获取或设置由于用户未输入密码确认问题而显示的错误信息
StepNextButtonImageUrl	获取或设置为"下一步"按钮显示的图像的 URL
StepNextButtonStyle	获取一个对 Style 对象的引用,该对象定义"下一步"按钮的设置
StepNextButtonText	获取或设置为"下一步"按钮显示的文本标题
StepNextButtonType	获取或设置呈现为"下一步"按钮的按钮类型
StepPreviousButtonImageUrl	获取或设置为"上一步"按钮显示的图像的 URL
StepPreviousButtonStyle	获取一个对 Style 对象的引用,该对象定义"上一步"按钮的设置
StepPreviousButtonText	获取或设置为"上一步"按钮显示的文本标题
StepPreviousButtonType	获取或设置呈现为"上一步"按钮的按钮类型
StepStyle	获取一个对 Style 对象的引用
Style	获取在 Web 服务器控件外部标记上呈现为样式属性的文本属性的集合
TabIndex	获取或设置 Web 服务器控件的选项卡索引
ToolTip	获取或设置当鼠标指针悬停在 Web 服务器控件上时显示的文本
Visible	获取或设置一个值,该值指示服务器控件是否作为 UI 呈现在页面上
WizardSteps	获取一个包含该控件定义的所有 WizardStepsBase 对象集合

- (1) Wizard 控件可用于下列工作。
- 收集多个步骤中的相关信息。
- 将大型输入页面分割成较小的逻辑步骤。
- 允许线性或非线性地导航各个步骤。
- (2) Wizard 控件可分成 4 大区域, 如图 3.13 所示。
- 向导步骤(WizardStep)区域:Wizard 控件使用 多个步骤描绘用户输入的不同部分。每个步骤 的内容添加在标记<asp:WizardStep>中,所有的<asp:WizardStep>又都包含在<WizardStep> 标记中。实际应用时,每次只能显示一个<asp: WizardStep>定义的内容。



续表

图 3.13 Wizard 控件区域划分

- 标题(Header)区域:用于在步骤顶部提供一致信息,此项是可选元素。
- 侧栏(Sidebar)区域:此项也是可选元素,通常显示在向导左边,包含所有步骤的列

68

表,并提供在各个步骤间的跳转。

• 导航(Navigation)按钮区域: 有 Wizard 内置导航功能,它会根据步骤类型 (StepType)设置值的不同,而呈现不同的导航按钮。

第−3-€∕∕

Web高级控件

表 3.11	Wizard	控件	的	常	用	事	件
--------	--------	----	---	---	---	---	---

事件名	说明
ActiveStepChanged	当切换控件中显示的步骤时发生
CancelButtonClick	当单击"取消"按钮时发生
FinishButtonClick	当单击"完成"按钮时发生
NextButtonClick	当单击"下一步"按钮时发生
PreviousButtonClick	当单击"上一步"按钮时发生
SiderBarButtonClick	当单击侧边栏中的按钮时发生

说明:

NextButtonClick、PreviousButtonClick和 SiderBarButtonClick 事件如不自主添加, 控件中也可自动实现"步骤"间的跳转。

【示例 3-7】 在 chapter3 网站根目录下创建名为 example3-7 的网页,页面内包含一个 Wizard 控件和若干其他基本控件,添加 Wizard 的 Steps 属性并进行应用。

(1)在 chapter3 网站根目录下添加 Wizard 控件,右击"》",在弹出的快捷菜单中选择 "添加/移除 Wizard Steps…",或者单击"属性"窗口 Wizard Steps 属性后面的"…"图标,出 现"WizardStep 集合编辑器"窗口,如图 3.14 所示。



图 3.14 "WizardStep 集合编辑器"窗口

(2)添加5个WizardStep默认类型Step,即5个步骤,最后一个"完成"Step设置其StepType属性为Complete(见图 3.15),其余均为Auto属性。

成员(M): ○ 基本信息 1 详细信息 2 兴趣委好 3 会员等级 4 完成 ClientIDMode Inherit EnableTheming True StepType Complete ValidateRequest Inherit

图 3.15 WizardStep 集合编辑器设置

每个 WizardStep 步骤都有一个 StepType 属性,作用是决定每个步骤中的导航按钮如何显示。StepType 的枚举值如下。

- Start:开始步骤。
- Step: 阶段步骤。
- Finish: 完成步骤。
- Complete: 结束步骤。
- Auto: 自动,系统自动识别其为何种 StepType 类型。

(3) 按照每个 Step 的界面,添加相应的控件,具体如图 3.16~图 3.20 所示。

基本信息	欢迎	您加入我们
<u> </u>	用户名: 🗌 密 码: 🗌	
		下一步

图 3.16 "基本信息"步骤布局

基本信息 <u>详细信息</u> 兴趣爱好 会员等级	欢迎您加入我们 □ 谈天 □ 说地 □ 军事 □ 做梦 □ 服饰
元叫 	上一步 下一步

图 3.18 "兴趣爱好"步骤布局



图 3.17 "详细信息"步骤布局

基本信息	欢迎您加入我们
<u> </u>	普通会员 👤
会员等级	上一步 完成
元以	

图 3.19 "会员等级"步骤布局