



## 本章要点

- 母版页的使用
- web.sitemap 的使用
- SiteMapDataSource 的使用
- Menu 的使用
- SiteMapPath 的使用

本章安排了两个实例，即使用母版及页面导航，来重新构造第 3 章所建立的 3 个相册显示页面：Albums.aspx、Photos.aspx 和 Details.aspx 页面。

实例 5 主要创建一个母版页，并重新构造这 3 个页面来使用这个母版页，从而简化 3 个相册显示页面的制作与管理，母版页技术对于大、中型网站的制作与管理是非常重要的；实例 6 主要在新建的母版页中使用导航控件，实现页面的导航。

### 4.1 实例 5——使用母版页简化页面制作

在第 3 章中，通过实例 2、实例 3 和实例 4 建立了 3 个页面，分别是 Albums.aspx 页面、Photos.aspx 页面和 Details.aspx 页面，这 3 个页面主要用于相册显示的基本功能，即相册的显示、某一相册中照片的显示以及指定照片的显示。

分析这 3 个页面可以发现，这 3 个页面具有一致的页面外观和页面结构，即具有相同的头部和脚部，以及相同的页面外框。对于这 3 个页面来说，在每个页面中重复制作这些相同的部分还不会觉得劳累，但如果对于一个大、中型网站中的成百上千个页面来说，这种重复制作简直是一种灾难。通过使用 ASP.NET 2.0 中的母版页，对这 3 个页面重新改造，可以简化页面的制作，便于页面的集中修改与管理。制作后的母版页 Default.master 的设计视图如图 4-1 所示。

实例 5 是在第 3 章实例的基础上，为 3 个相册显示页面新建一个母版页。在进行实例 5 的设计过程中，应首先建立实例 5 的网站。

#### 4.1.1 相册显示页面的结构分析

查看第 3 章已经建立的 3 个页面：Albums.aspx、Photos.aspx 和 Details.aspx 页面，这里显示了 Albums.aspx 和 Photos.aspx 两个页面的运行页面，分别如图 4-2 和图 4-3 所示。从图 4-2

和图 4-3 中可以看出，这两个页面具有相同的外观和一致的界面，它们的页面结构主要由上、中、下三部分所组成。

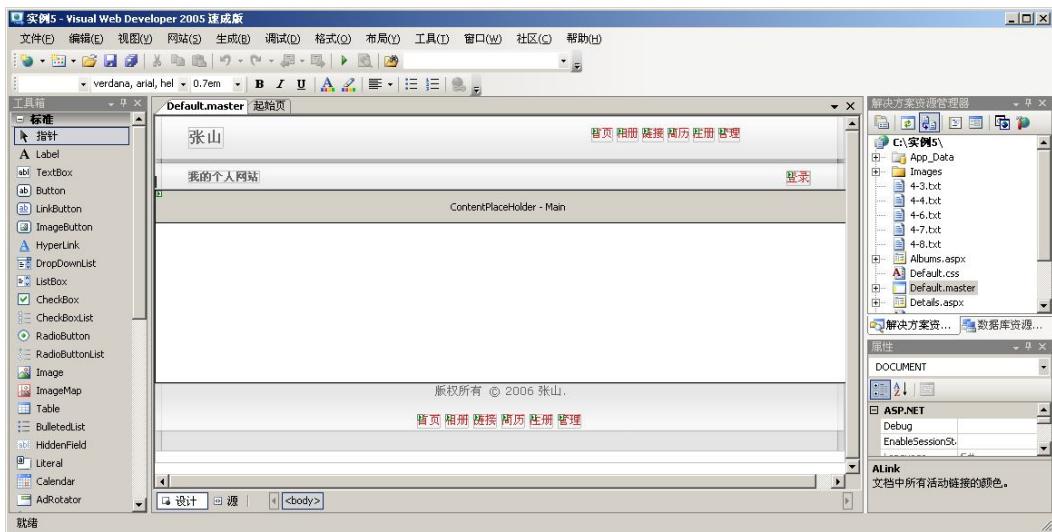


图 4-1 Default.master 的设计视图



图 4-2 Albums.aspx 的运行页面



图 4-3 Photos.aspx 的运行页面

上面部分称为页面头部 (Header)，主要是说明整个网站的名称，或者一个分类页面的名称，如可以在其中放置网站 Logo 和网站的页面地址链接等。可以看出，上述两个页面中的头部内容完全一样。

中间部分是内容部分 (Content)，不同的页面具有不同的内容，Albums.aspx 显示相册的内容，Photos.aspx 显示某一相册中的所有照片。很显然，在这两个页面中，该部分的内容是变化的。

下面部分是页面脚部 (Footer)，主要放置网站的版权说明、公司名称、地址和制作日期等。可以看出，上述两个页面中的脚部内容也完全一样。

实际上，对于 Albums.aspx、Photos.aspx 和 Details.aspx 三个页面来说，它们都具有相同的头部和脚部，只是内容部分不同而已。前面是分别在这 3 个页面中重复设计相同的头部和脚部来实现这 3 个页面，这种方法对于区区 3 个页面数量来说，工作量还不算大，但如果是一个大型网站，成百上千的页面需要这些重复工作，不仅工作量大，而且容易出错，特别是当需要修改这些头部和脚部的内容时，这种方法将带来巨大的工作量，不利于网页的制作和管理。

因此，如果构建一个模板页面，在其中设计出不同的页面部分，头部和脚部是多个页面的相同部分，中间部分随不同页面的内容不同而变化，那么其他页面只需使用这个模板，就可以节省重复设计头部和脚部的大量时间，如果需要修改页面的头部和脚部，只需修改这个模板中的相应部分即可，非常有利于网页的制作和管理。

VWD 提供了母版技术 (Master page) 来实现上述的模板页面。

### 4.1.2 设计母版页

VWD 提供了专门的母版页项目用于创建母版页，并且可在可视化环境下通过拖拉方式设计母版，十分方便。

#### 1. 创建母版页

在 VWD 中，右击“解决方案资源管理器”窗口中的“实例 5”项目，并在弹出的快捷菜单中，如图 4-4 所示，单击“添加新项”命令，在打开的如图 4-5 所示的“添加新项”对话框中，选择“母版页”模板，输入需要创建的母版名称为 Default.master，注意母版文件的后缀名为 master，然后单击“添加”按钮，即可在选择的项目下创建一个母版 Default.master。



图 4-4 添加新项



图 4-5 “添加新项”对话框

在“解决方案资源管理器”窗口下双击母版文件 Default.master，查看该母版文件的设计视图，如图 4-6 所示。

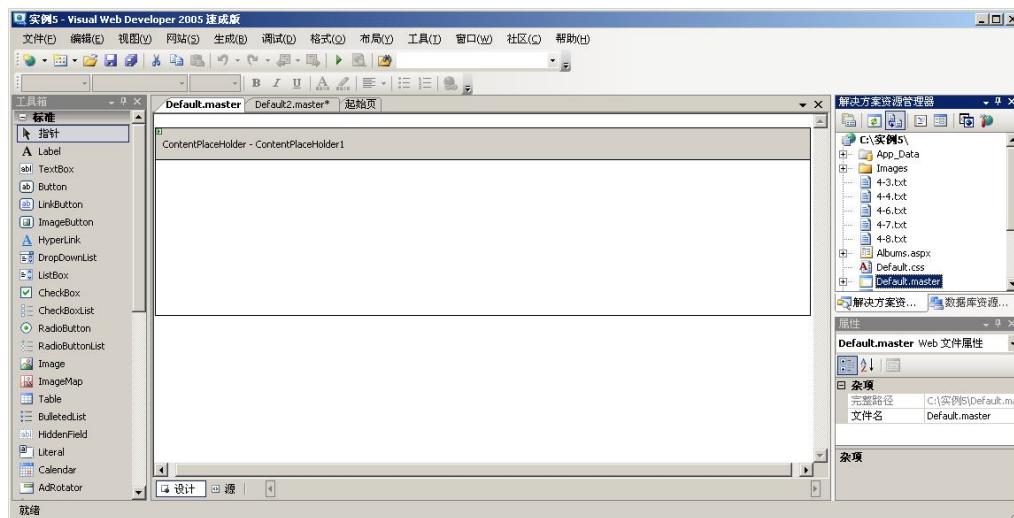


图 4-6 Default.master 设计视图

在图 4-6 中，中间的方框部分称为内容占位符（ContentPlaceHolder），该内容占位符是页面设计中的变化部分。在设计母版时，只需设计内容占位符在页面中的布局，不需设计其中的内容。而内容占位符以外的其他部分，则是母版设计的主要内容，这些部分是多个页面共享的相同部分。

查看母版文件 Default.master 的代码视图，可以得到代码 4-1 中 Default.master 页面的相关 HTML 代码。

#### 代码 4-1 Default.master 页面的相关 HTML 代码

```
1: <html xmlns="http://www.w3.org/1999/xhtml" >
2:   <head runat="server">
3:     <title>Untitled</title>
4:   </head>
5:   <body>
6:     <form id="form1" runat="server">
7:       <div>
8:
9:         <asp:contentplaceholder id="ContentPlaceholder1" runat="server">
10:           </asp:contentplaceholder>
11:
12:       </div>
13:     </form>
14:   </body>
15: </html>
```

从代码 4-1 中可以看出，创建的母版与普通的页面十分相似，事实上，母版本身就是一个页面，只不过是一个特殊的页面而已。其中的第 9 行和第 10 行是内容占位符，其他的页面要使用这个母版，就需要在该内容占位符中填充相关内容；要设计母版，就需要在以上 HTML 代码中的相关位置添加相应的代码，具体来说，主要是在第 8 行语句中，或者在第 11 行语句中添加相关代码。这些添加的代码是多个页面的共享部分，是母版设计的核心。

## 2. 设计母版页

设计母版页，可以在母版页的设计视图下，通过可视化方式拖放相关控件，输入相关内容；还可以在母版页的代码视图下，直接输入事先设计好的有关 HTML 代码，这里采用后一种方式。

为了在母版页 Default.master 中使用已经定义的样式，首先需要用代码 4-2 中的内容替换代码 4-1 中的第 2 行到第 4 行。

#### 代码 4-2 Default.master 页面的相关 HTML 代码

```
1: <head runat="server">
2:   <Link href="Default.css" type="text/css rel="stylesheet">
3:   <link href="Frame.css" rel="stylesheet" type="text/css">
4:   <title></title>
5: </head>
```

然后在代码 4-1 的第 8 行中，插入母版的设计代码，即页面的头部。这些插入的代码如代码 4-3 所示。

代码 4-3 Default.master 页面头部的相关 HTML 代码

接下来，在代码 4-1 的第 11 行中，插入母版页的设计代码，即页面的脚部。这些插入的代码如代码 4-4 所示。

代码 4-4 Default.master 页面脚部的相关 HTML 代码

```
1: <div class="footerbg" >
2:   <div class="footer">
3:     &nbsp;版权所有 © 2006 张山.
4:     <br />
5:     <br />
6:     <asp:HyperLink ID="HyperLink7" runat="server"
          NavigateUrl="~/Default.aspx">首页</asp:HyperLink>
7:     <asp:HyperLink ID="HyperLink8" runat="server"
          NavigateUrl="~/Albums.aspx" >相册</asp:HyperLink>
8:     <asp:HyperLink ID="HyperLink9" runat="server"
          NavigateUrl="~/Links.aspx"    >链接</asp:HyperLink>
9:     <asp:HyperLink ID="HyperLink10" runat="server"
          NavigateUrl="~/Resume.aspx" >简历</asp:HyperLink>
```

```

10: <asp:HyperLink ID="HyperLink11" runat="server"
   NavigateUrl("~/Register.aspx" >注册</asp:HyperLink>
11: <asp:HyperLink ID="HyperLink12" runat="server"
   NavigateUrl "~/Admin/Albums.aspx" >管理</asp:HyperLink>
12: </div>
13: </div>

```

通过设计母版的头部和脚部，基本完成了母版 Default.master 的设计，母版 Default.master 的设计视图如图 4-7 所示。

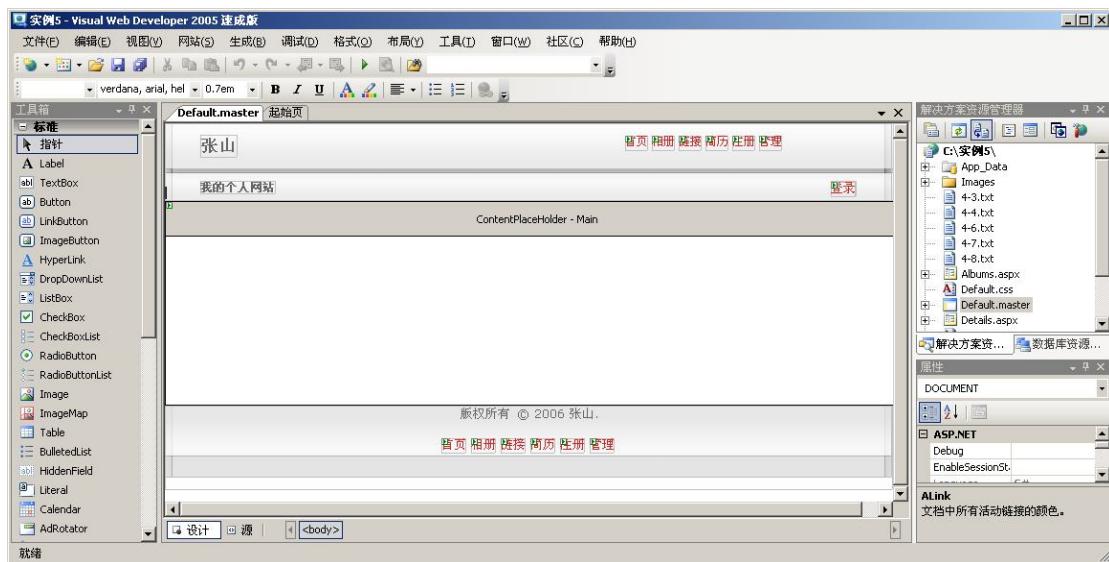


图 4-7 Default.master 完成后的设计视图

### 4.1.3 使用母版页

要使用母版页，在 VWD 中，新增加一个页面时，只需选中“选择母版页”，然后就可以在生成页面的内容占位符中填充页面设计的变化部分。

#### 1. 新建 Albums.aspx 页面

在 VWD 中，首先右击 VWD 右边“解决方案资源管理器”窗口下的“实例 5”项目，在弹出的快捷菜单中选择“添加新项”命令，然后在打开的如图 4-8 所示的“添加新项”对话框中，选择“Web 窗体”模板，在“名称”框中输入需要创建的页面名称为 Albums.aspx，并选中“选择母版页”，表明在新建 Albums.aspx 页面时，需要使用相应的母版，然后单击“添加”按钮。

在打开的如图 4-9 所示的“选择母版页”对话框中，选择相应的母版文件。对话框将列出当前可以使用的所有母版文件，这里选择 Default.master。然后单击“确定”按钮，即可新建一个使用母版的 Albums.aspx 页面。



图 4-8 新建 Albums.aspx 页面对话框



图 4-9 “选择母版页”对话框

在 VWD 中查看 Albums.aspx 的设计视图，如图 4-10 所示，由于使用了 Default.master 母版，页面的上、下两部分，即页面的头部与脚部均显示为灰色，这两部分是不可编辑的，如果需要编辑、修改这两部分内容，必须在母版 Default.master 中进行。

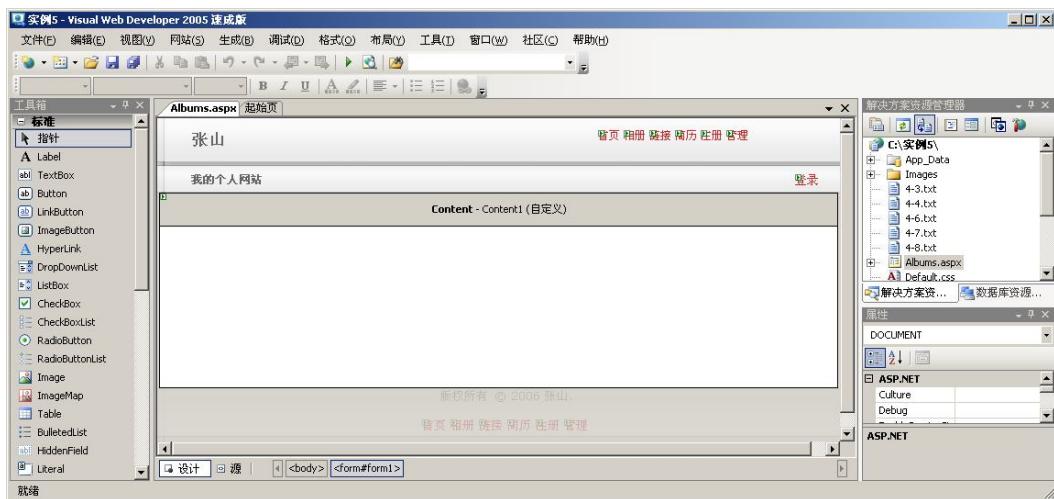


图 4-10 Albums.aspx 的设计视图

页面的中间部分，即内容占位符，是 Albums.aspx 页面中可以编辑的部分，可以通过可视化方式拖放相关控件，输入相关内容来填充所需要的内容，以便显示相册的内容。这里仍然采用在 Albums.aspx 页面的代码视图下直接输入事先设计好的有关 HTML 代码来重新设计 Albums.aspx 页面。

在 VWD 中查看 Albums.aspx 的代码视图，其 HTML 代码如代码 4-5 所示。

#### 代码 4-5 Albums.aspx 页面的 HTML 代码

```

1: <%@ Page Language="C#" MasterPageFile("~/Default.master"
2:                         AutoEventWireup="true" CodeFile="Albums.aspx.cs"
3:                         Inherits="Albums"
4:                         Title="Untitled Page" %>
5: <asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
6:                         Runat="Server">
7: </asp:Content>
```

可以发现其中的 HTML 语句十分简单，第 1 行到第 3 行的内容描述了 Albums.aspx 页面使用的母版文件是 Default.master，第 4 行到第 7 行是内容占位符，要使用母版来重新设计 Albums.aspx 页面，就是在第 6 行中插入显示相册内容的相关代码。

将代码 4-6 中的代码插入到代码 4-5 中的第 6 行，就可以完成 Albums.aspx 页面的重新设计，并且此时的 Albums.aspx 页面使用了 Default.master 母版。下面简单说明代码 4-6 中的 HTML 代码。

#### 代码 4-6 在 Albums.aspx 中插入内容占位符的 HTML 代码

```

1: <div class="shim gradient"></div>
2: <div class="page" id="albums" >
3:   <h3>欢迎访问我的照片集</h3>
4:   <p>以下照片是我多年的外出作品。内容虽然不多，但挺精彩的，希望大家喜欢。</p>
5:   <hr />
6:
7:
8:   <asp:DataList ID="DataList1" runat="server" CssClass="view"
9:                 DataSourceID="SqlDataSource1" RepeatColumns="2"
10:                RepeatDirection="Horizontal" >
11:      <ItemStyle cssClass="item" />
12:      <ItemTemplate>
13:        <table border="0" cellpadding="0" cellspacing="0"
14:          class="album-frame">
15:          <a href='Photos.aspx?AlbumID=<%# Eval("AlbumID") %>' >
16:            &Size=M"
17:                  class="photo_198" style="border:4px solid white"
18:                  alt='Sample Photo from Album Number <%# Eval("AlbumID") %>
19:          ' /></a>
20:        </table>
21:        <h4><a href="Photos.aspx?AlbumID=<%# Eval("AlbumID") %>">
```

```
22: </ItemTemplate>
23: </asp:DataList>
24: </div>
25: <asp:SqlDataSource ID="SqlDataSource1" runat="server"
26:     ConnectionString="<%$ ConnectionStrings:Personal %>" 
27:     SelectCommand="SELECT [Albums].[AlbumID], [Albums].[Caption],
28:                     [Albums].[IsPublic], Count([Photos].[PhotoID]) AS
29:                     NumberOfPhotos
30:             FROM [Albums] LEFT JOIN [Photos] ON
31:                 [Albums].[AlbumID] = [Photos].[AlbumID]
32:             WHERE [Albums].[IsPublic] = 1
33:             GROUP BY [Albums].[AlbumID], [Albums].[Caption],
34:                     [Albums].[IsPublic]">
35: </asp:SqlDataSource>
```

以上代码基本分为 3 个部分，第 1 行到第 7 行是一般的页面装饰和文字说明；中间部分第 8 行到第 23 行是 DataList 控件的设置部分；最后是 SqlDataSource 的设置部分。需要注意的是，不要遗漏 SqlDataSource 部分。

重新运行使用母版后的 Albums.aspx 页面，实际表明是成功的。

## 2. 新建 Photos.aspx 页面

新建 Photos.aspx 页面，与前面所建立的 Albums.aspx 页面完全类似，这里只列出需要插入到内容占位符中的有关代码，如代码 4-7 所示。

代码 4-7 在 Photos.aspx 中插入内容占位符的 HTML 代码

```
1: <div class="shim solid"></div>
2: <div class="page" id="photos">
3:   <div class="buttonbar buttonbar-top">
4:     <a href="Albums.aspx">Albums </a>
5:   </div>
6:   <asp:DataList ID="DataList1" runat="Server" cssclass="view"
7:     dataSourceID="SqlDataSource1" repeatColumns="4"
repeatdirection="Horizontal"
8:     onitemdatabound="DataList1_ItemDataBound"
EnableViewState="false">
9:     <ItemTemplate>
10:       <table border="0" cellpadding="0" cellspacing="0" class="photo-frame">
11:         <tr>
12:           <td class="topx--"></td>
13:           <td class="top-x->"></td>
14:           <td class="top--x"></td>
15:         </tr>
16:         <tr>
17:           <td class="midx--"></td>
18:           <td><a href='Details.aspx?AlbumID=<%# Eval("AlbumID") %>&
Page=<%# Container.ItemIndex %>'>
19:             &
```

```

21:           Size=S" class="photo_198" style="border:4px solid white"
22:           alt='Thumbnail of Photo Number <%# Eval("PhotoID") %>' />
</a></td>
23:   <td class="mid--x"></td>
24:   </tr>
25:   <tr>
26:     <td class="botx--"></td>
27:     <td class="bot-x->"></td>
28:     <td class="bot--x"></td>
29:   </tr>
30: </table>
31: <p><%# Server.HtmlEncode(Eval("Caption").ToString()) %></p>
32: </ItemTemplate>
33: <FooterTemplate>
34: </FooterTemplate>
35: </asp:DataList>
36: <asp:panel id="Panel1" runat="server" visible="false"
CssClass="nullpanel">
37:   目前该相册中没有照片。</asp:panel>
38: <div class="buttonbar">
39:   <a href="Albums.aspx">Albums </a>
40: </div>
41: </div>
42: <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:Personal %>"
SelectCommand="SELECT * FROM [Photos] LEFT JOIN [Albums]
ON [Albums].[AlbumID] = [Photos].[AlbumID]
WHERE [Photos].[AlbumID]=@Album AND ([Albums].[IsPublic]=1)">
43:   <SelectParameters>
44:     <asp:QueryStringParameter DefaultValue="1" Name="Album"
45:       QueryStringField="Album" />
46:   </SelectParameters>
47: </asp:SqlDataSource>

```

在以上代码中，第 6 行到第 35 行设置了数据访问控件 `DataList`，第 36 行到第 37 行设置了一个面板控件，用来显示当相册中没有图片时，在页面中所显示的内容。第 42 行到第 51 行设置了数据源控件 `SqlDataSource`，其中设置了一个查询语句，该查询语句中的输入参数由页面参数来传递。

### 3. 新建 Details.aspx 页面

新建 `Details.aspx` 页面，与前面所建立的 `Albums.aspx` 以及 `Photos.aspx` 页面完全类似，同样这里只列出需要插入到内容占位符中的有关代码，如代码 4-8 所示。

**代码 4-8 在 `Details.aspx` 中插入内容占位符的 HTML 代码**

```

1: <div class="shim solid"></div>
2: <div class="page" id="details">
3: <asp:formview id="FormView1" runat="server" datasourceid="SqlDataSource1">

```

```
4:      cssclass="view" borderstyle="solid" borderwidth="0" CellPadding="0"
5:      cellspacing="0" EnableViewState="false" AllowPaging="true">
6:      <itemtemplate>
7:          <div class="buttonbar buttonbar-top" >
8:              <a href="Albums.aspx">Albums</a>
9:              <asp:Button ID="Button1" runat="server" CommandName="Page"
10:                 CommandArgument="First" Text="First" />
11:              <asp:Button ID="Button2" runat="server"
12:                 CommandName="Page" CommandArgument="Prev" Text="Prev" />
13:              <asp:Button ID="Button3" runat="server"
14:                 CommandName="Page" CommandArgument="Next" Text="Next" />
15:              <asp:Button ID="Button4" runat="server" CommandName="Page"
16:                 CommandArgument="Last" Text="Last" />
17:          </div>
18:          <p><%# Server.HtmlEncode(Eval("Caption")) .ToString() %></p>
19:          <table border="0" cellpadding="0" cellspacing="0" class="photo-frame">
20:              <tr>
21:                  <td class="topx--"></td>
22:                  <td class="top-x->"></td>
23:                  <td class="top--x"></td>
24:              </tr>
25:              <tr>
26:                  <td class="midx--"></td>
27:                  <td>&
28:                         Size=L" class="photo_198" style="border:4px solid white"
29:                         alt='Photo Number <%# Eval("PhotoID") %>' /></td>
30:                  <td class="mid--x"></td>
31:              </tr>
32:              <tr>
33:                  <td class="botx--"></td>
34:                  <td class="bot-x->"></td>
35:                  <td class="bot--x"></td>
36:              </tr>
37:          </table>
38:          <p><a href='Download.aspx?AlbumID=<%# Eval("AlbumID") %>&
39:                           Page=<%# Container.DataItemIndex %>'>Download</a></p>
40:          <div class="buttonbar" >
41:              <a href="Albums.aspx">Albums</a>
42:              &nbsp;&nbsp;&nbsp;
43:              <asp:Button ID="Button5" runat="server" CommandName="Page"
44:                 CommandArgument="First" Text="First" />
45:              <asp:Button ID="Button6" runat="server" CommandName="Page"
46:                 CommandArgument="Prev" Text="Prev" />
47:              <asp:Button ID="Button7" runat="server" CommandName="Page"
48:                 CommandArgument="Next" Text="Next" />
49:              <asp:Button ID="Button8" runat="server" CommandName="Page"
```

```

50:           CommandArgument="Last" Text="Last" />
51:     </div>
52:   </itemtemplate>
53: </asp:formview>
54: </div>
55: <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ 
56:     ConnectionStrings:Personal %>" 
57:     SelectCommand="SELECT *    FROM [Photos] LEFT JOIN [Albums] 
58:         ON [Albums].[AlbumID] = [Photos].[AlbumID]
59:         WHERE [Photos].[AlbumID] = @Album AND ([Albums].[IsPublic] = 1 )">
60:   <SelectParameters>
61:     <asp:QueryStringParameter DefaultValue="1" Name="Album" 
62:         QueryStringField="Album" />
63:   </SelectParameters>
64: </asp:SqlDataSource>

```

在以上代码中，第 3 行到第 53 行设置了数据访问控件 FormView。第 55 行到第 64 行设置了数据源控件 SqlDataSource，其中设置了一个查询语句，该查询语句中的输入参数由页面参数来传递。

## 4.2 网站导航

网站是由许许多多的页面所组成的，网站中页面之间的导航，即页面之间的相互链接，随着网站规模的越来越复杂而变得越来越不容易管理，特别是当页面结构发生变化，如增加新的页面、删除旧的页面时，网站管理员将面临巨大的挑战。

为解决网站中页面间的导航问题，ASP.NET 2.0 提供了很好的解决方案。通过 XML 格式的站点地图文件（Web.sitemap）集中定义整个网站的层次结构，而且这种层次结构与真正的页面存储物理结构无关，非常容易实现网站中页面的管理与导航。

另外 VWD 提供了可视化的导航控件，如 TreeView、SiteMapPath 以及 Menu 控件，从而不需要编写代码就可以非常方便地实现页面的导航。

### 4.2.1 创建一个网站以及站点地图文件

在实现网站的导航过程中，首先需要创建一个网站以及一个站点地图文件。

#### 1. 新建 SiteNavigation 网站

在 VWD 中，单击“文件”菜单中的“新建网站”命令，在打开的“新建网站”对话框中选择“ASP.NET 网站”项目模板，使用“文件系统”，网站的名称设定为 SiteNavigation，如图 4-11 所示。

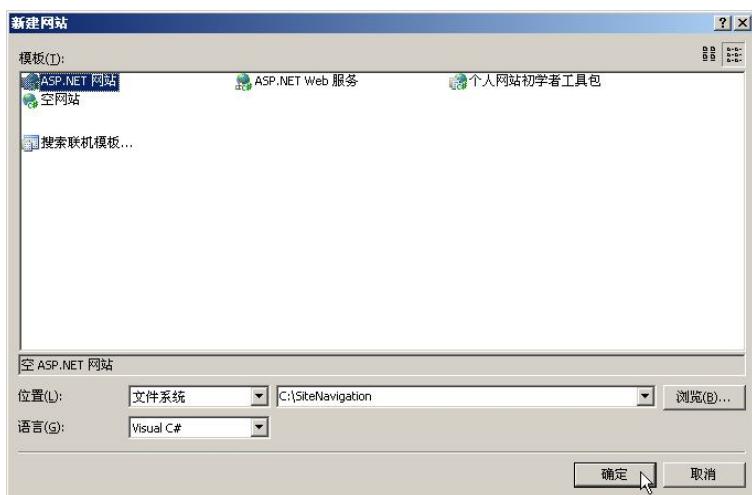


图 4-11 新建 SiteNavigation 网站

单击“确定”按钮后，VWD 会新建一个含有 App\_Data 目录以及一个 Default.aspx 页面的 SiteNavigation 网站。

右击“解决方案资源管理器”窗口中的 Default.aspx 文件，在弹出的快捷菜单中选择“重命名”命令，将 Default.aspx 文件更名为 Home.aspx 文件，然后选择该文件，在 VWD 的设计视图中输入“Home”，并将该文字设定为“Heading 1”样式。

然后右击“解决方案资源管理器”窗口中的 SiteNavigation 项目，在弹出的快捷菜单中选择“添加新项”命令，新建一个 Products.aspx 页面，选择该文件，并在 VWD 的设计视图中输入“Products”，并将该文字设定为“Heading 1”样式。

重复上述步骤，分别再新建 Hardware.aspx 页面、Software.aspx 页面、Services.aspx 页面、Training.aspx 页面、Consulting.aspx 页面和 Support.aspx 页面，并分别在其页面中输入与文件名相同的名字，如在 Hardware.aspx 页面中输入“Hardware”，并将该文字设定为“Heading 1”，在 Software.aspx 页面中输入“Software”，并将该文字设定为“Heading 1”样式等，这里不再重复。

到此为止，在 SiteNavigation 网站中新建了 8 个页面，其网站目录结构如图 4-12 所示。

## 2. 建立站点地图文件

右击“解决方案资源管理器”窗口中的 SiteNavigation 项目，在弹出的快捷菜单中选择“添加新项”命令，在项目模板中选择第 6 行第 1 列的

“站点地图”模板，如图 4-13 所示，此时新建的文件名称为 Web.sitemap。这里需要说明的是，不应更改这个站点地图文件的名称，因为在 VWD 中，许多导航控件的数据源就是读取这个默认的 Web.sitemap 文件的。

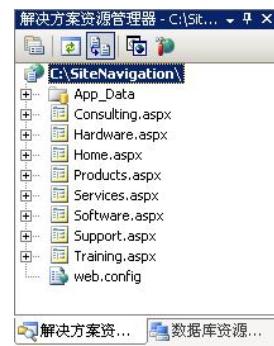


图 4-12 SiteNavigation 网站的目录结构



图 4-13 新建站点地图文件

在图 4-13 的对话框中，单击“添加”按钮，就会新建一个空白的站点地图文件 Web.sitemap。在“解决方案资源管理器”窗口中双击“Web.sitemap”文件，除保留第一行内容之外，删除其余的内容，并将代码 4-9 中的内容复制到 Web.sitemap 文件中去，然后保存 Web.sitemap 文件。

#### 代码 4-9 站点地图文件

```

1: <siteMap>
2:   <siteMapNode title="Home" description="Home" url("~/home.aspx")>
3:     <siteMapNode title="Products" description="Our products"
url "~/Products.aspx">
4:       <siteMapNode title="Hardware" description="Hardware we offer"
url "~/Hardware.aspx" />
5:       <siteMapNode title="Software" description="Software for sale"
url "~/Software.aspx" />
6:     </siteMapNode>
7:     <siteMapNode title="Services" description="Services we offer"
url "~/Services.aspx">
8:       <siteMapNode title="Training" description="Training"
url "~/Training.aspx" />
9:       <siteMapNode title="Consulting" description="Consulting"
url "~/Consulting.aspx" />
10:      <siteMapNode title="Support" description="Support"
url "~/Support.aspx" />
11:    </siteMapNode>
12:  </siteMapNode>
13:</siteMap>

```

站点地图文件是一个 XML 格式的文件，通过该文件实现站点结构的集中管理。在代码 4-9 所示的站点地图文件中，将 SiteNavigation 网站中的 8 个页面设定为 3 个层次。第 1 个层次是 Home.aspx 页面，第 2 个层次是 Products.aspx 页面和 Services.aspx 页面。其中通过 Products.aspx 页面链接到第 3 个层次页面，即 Hardware.aspx 页面和 Software.aspx 页面；通

过 Services.aspx 页面链接到第 3 个层次页面，即 Training.aspx 页面、Consulting.aspx 页面以及 Support.aspx 页面。

从这里可以看出，SiteNavigation 网站的站点地图的层次结构与 SiteNavigation 网站的目录结构是没有关联的，便于对网站的集中管理。

#### 4.2.2 使用 TreeView 控件实现导航

在 VWD 中，打开 Home.aspx 页面，在设计视图下，将控件工具箱“数据”控件组中的 SiteMapDataSource 控件拖放到 Home.aspx 页面的适当位置，该数据源控件在使用时不需要进行任何设置，它将自动读取站点地图文件 Web.sitemap 中的内容。

然后将控件工具箱 Navigation 控件组中的 TreeView 控件拖放到 Home.aspx 页面，单击 TreeView 控件右上方的智能任务菜单，并在出现的任务菜单中选择数据源，如图 4-14 所示。

选择 SiteMapDataSource1 为数据源后，TreeView 控件的界面会马上发生变化，自动读取 Web.sitemap 中的内容并显示 SiteNavigation 网站的层次结构，如图 4-15 所示。



图 4-14 选择数据源



图 4-15 选择数据源后

图 4-16 是 Home.aspx 页面的运行界面，在 TreeView 控件中显示了 SiteNavigation 网站中 8 个页面的层次结构，非常清楚，一目了然。单击 Home 左边的折叠、展开按钮，可以将 Home 所包含的页面折叠隐藏或展开显示，非常方便。



图 4-16 Home.aspx 页面的运行界面

单击 TreeView 控件中的每个页面链接，可以转移到相关的页面，实现页面的导航。而这一功能的实现并不需要编写相关的代码。

这里只在 Home.aspx 页面中添加了导航的控件 TreeView，如果要实现 8 个页面之间的相

互链接，还需要在其他的 7 个页面中分别添加导航控件 TreeView，这里不再重复。

### 4.2.3 使用 SiteMapPath 控件显示导航路径

VWD 还提供了 SiteMapPath 控件，用来显示导航的路径，即显示当前的页面以及该页面所处的层次路径。

在 VWD 中，打开 Products.aspx 页面，在设计视图下，将控件工具箱“数据”控件组中的 SiteMapPath 控件拖放到 Products.aspx 页面中文字“Products”的下方，在使用 SiteMapPath 控件时也不需要进行任何设置，它将自动读取站点地图文件 Web.sitemap 中的内容，显示页面的路径，如图 4-17 所示。

重复上述步骤，在 Hardware.aspx 页面中也添加 SiteMapPath 控件，其显示的内容如图 4-18 所示，最后的路径名称表示当前页面的名称，前面的路径名称用链接地址来表示，同样表示相关页面的名称，并分别表明不同的层次，如 Home 属于第一层次，Products 属于第二层次，单击相关的链接路径，可以转移到相关的页面。

**Products**

[Home](#) > Products

图 4-17 Products.aspx 页面

**Hardware**

[Home](#) > [Products](#) > Hardware

图 4-18 Hardware.aspx 页面

图 4-19 是 Hardware.aspx 页面的运行界面。



图 4-19 Hardware.aspx 页面的运行界面

这里需要说明的是，SiteMapPath 控件显示了当前的页面以及到达该当前页面的层次路径，但该路径并不表示用户浏览页面的历史路径。

### 4.2.4 使用 Menu 控件实现导航菜单

在 VWD 中，控件工具箱 Navigation 控件组中还提供了 Menu 控件，可以实现菜单形式的页面导航。

在 VWD 中打开 Products.aspx 页面，在设计视图下，将控件工具箱“导航”控件组中的

Menu 控件拖放到 Products.aspx 页面中 SiteMapPath 控件的下方，然后单击 Menu 控件右上方的任务菜单，如图 4-20 所示，在其中选择数据源，选择“新建数据源”后，打开如图 4-21 所示的选择数据源类型对话框。

在图 4-21 中，选择“站点地图”类型的数据源，即 Menu 控件的数据源来自于站点地图文件 Web.sitemap，然后单击“确定”按钮即可。



图 4-20 选择数据源



图 4-21 选择数据源类型

图 4-22 是 Products.aspx 页面的运行界面。



图 4-22 Products.aspx 页面的运行界面

在图 4-22 中，将鼠标移动到 Menu 控件的相关位置，将会出现 Home 下的子菜单 Product 以及子菜单 Services，还有 Services 下的子菜单 Training、Consulting 以及 Support。单击菜单中的任意一个链接，就可以实现页面之间的转移。

#### 4.2.5 在母版页中实现站点导航

在实现 SiteNavigation 网站页面导航的过程中，有 8 个页面，根据需要在每一个页面添加相关的导航控件，这对于一个页面数量不大的网站来说是无关紧要的，但对于一个有成百上千页面的大中型网站来说，工作量是巨大的，而且一旦要修改导航控件的界面和位置，网站管理

员将面临巨大的挑战。

为解决上述问题, ASP.NET 2.0 提供了很好的母版解决方案。通过定义一个或多个母版页 (Master pages), 类似于模板的概念, 将共同拥有的页面外观集中在一个或几个母版中, 便于页面的制作、修改和管理。

### 1. 新建母版页

右击“解决方案资源管理器”窗口中的 SiteNavigation 项目, 在弹出的快捷菜单中选择“添加新项”命令, 在项目模板中选择第 1 行第 2 列的“母版页”模板, 如图 4-23 所示, 此时新建的母版页文件名称为 MasterPage.master。



图 4-23 新建母版页

在图 4-23, 单击“添加”按钮, 即可新建一个名称为 MasterPage.master 的母版页, 该母版页在设计视图下显示的是一个默认的内容占位符控件 (ContentPlaceHolder), 如图 4-24 所示。

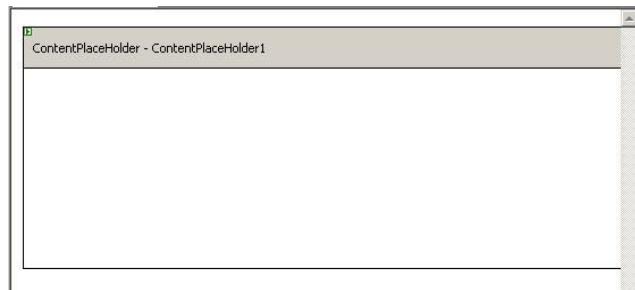


图 4-24 母版页的设计视图

### 2. 在母版页中添加导航控件

在母版页 MasterPage.master 的设计视图中, 单击内容占位符控件, 在按下键盘的向左箭头键后, 再按下空格键, 此时就会在内容占位符控件的上部插入一个空白行。

将控件工具箱“数据”控件组中的 SiteMapDataSource 控件拖放到 MasterPage.master 页面的内容占位符控件的上方位置, 然后单击 SiteMapDataSource 控件, 在按下键盘的向右箭头键

后，再按下空格键，此时会在 SiteMapDataSource 控件的下方插入一个空白行。

单击 VWD 菜单“布局”中的“插入表”命令，用于在当前的光标位置，即在 SiteMapDataSource 控件的下方插入一个表格，打开如图 4-25 所示的“插入表”对话框，这里设定表格的行数为 1，列数为 2，宽度为 100%，然后单击“确定”按钮，即可插入一个 1 行 2 列的表格。

将控件工具箱 Navigation 控件组中的 TreeView 控件拖放到表格的左边一列，并在 TreeView 控件的任务菜单中将数据源选择为 SiteMapDataSource1；然后将控件工具箱 Navigation 控件组中的 SiteMapPath 控件拖放到表格的右边一列。单击表格右边一列中的空白处，并按下 Shift+Enter 组合键，在表格右边的一列中新建一个空白行，再用鼠标将内容占位符控件拖放到表格的右边单元中，在 SiteMapPath 控件的下方，图 4-26 给出了母版页 MasterPage.master 的设计视图。



图 4-25 “插入表”对话框



图 4-26 母版页的设计视图

### 3. 创建内容页面

在新建了母版页 MasterPage.master 后，要创建内容页面就相对简单多了。

右击“解决方案资源管理器”窗口中的 Home.aspx 页面，在弹出的快捷菜单中选择“删除”命令，将原有的 Home.aspx 页面删除。

重复上述步骤，将原有的其他 7 个页面分别删除。

再次右击“解决方案资源管理器”窗口中的 SiteNavigation 项目，在弹出的快捷菜单中选择“添加新项”命令，在打开的“添加新项”对话框中选择“Web 窗体”项目模板，如图 4-27 所示。

在图 4-27 中，设定新建的页面名称为 Home.aspx，并在语言栏的右边位置选择“选择母版页”，表明该页面将会使用前面已经建立的母版页，此时将会打开如图 4-28 所示的“选择母版页”对话框。

在图 4-28 的对话框中，左边列出了 SiteNavigation 网站的目录结构，右边列出了现有的母版页列表，这里只显示了一个母版页文件 MasterPage.master，选定该母版页文件，然后单击“确定”按钮，即可完成使用母版页的 Home.aspx 页面的新建。

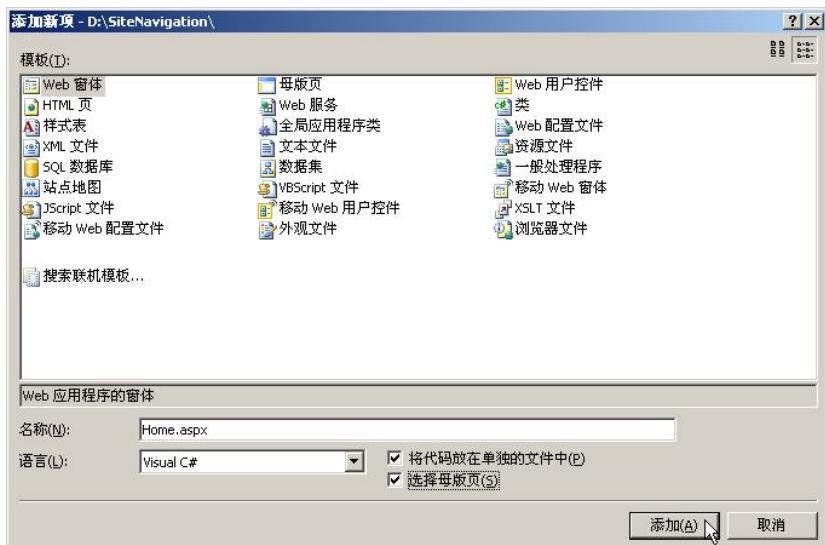


图 4-27 新建内容页面



图 4-28 选择母版页

在 VWD 中打开 Home.aspx 页面的设计视图，界面如图 4-29 所示。



图 4-29 设计 Home.aspx 页面

在 Home.aspx 页面中，只有内容占位符控件部分是可以编辑的，而母版页部分是灰色的。

不可编辑部分，母版部分的编辑、修改只能在母版页中实现。单击内容占位符控件中的空白处，输入“Home”，并将该文字的格式设定为“Heading 1”。

重复这一步骤，实现其他的7个页面。

图4-30是Products.aspx页面的运行界面。



图4-30 Products.aspx页面的运行界面

### 4.3 实例6——在母版页中设计页面导航

在实例5中创建了一个Default.master母版，主要实现的是页面导航以及网页的版权说明等功能，利用普通的HyperLink控件，可实现页面地址的链接，如果网站的页面层次结构发生变化，则需要更改母版中的多个链接地址，至少需要在页面的头部与脚部重复同样的工作。

VWD提供了专门的页面导航控件，如SiteMapPath控件和Menu控件等，它们位于VWD左边工具箱中的Navigation控件组中，要使用这些控件，首先需要建立整个网站的站点地图文件(web.sitemap)，并通过SiteMapDataSource数据源控件来绑定这个站点地图文件，然后使用导航控件Menu即可实现页面的自动导航，通过使用SiteMapPath控件，还可以显示页面路径。

通过使用页面导航控件，如果网站页面数量增加或者网站页面结构发生变化，只需要修改web.sitemap文件即可，便于页面的管理。添加页面导航等功能后的母版Default.master的设计视图如图4-31所示。

实例6是在实例5的基础上重新设计母版Default.master，构造页面导航。在进行实例6的设计过程中，应首先建立实例6的网站。

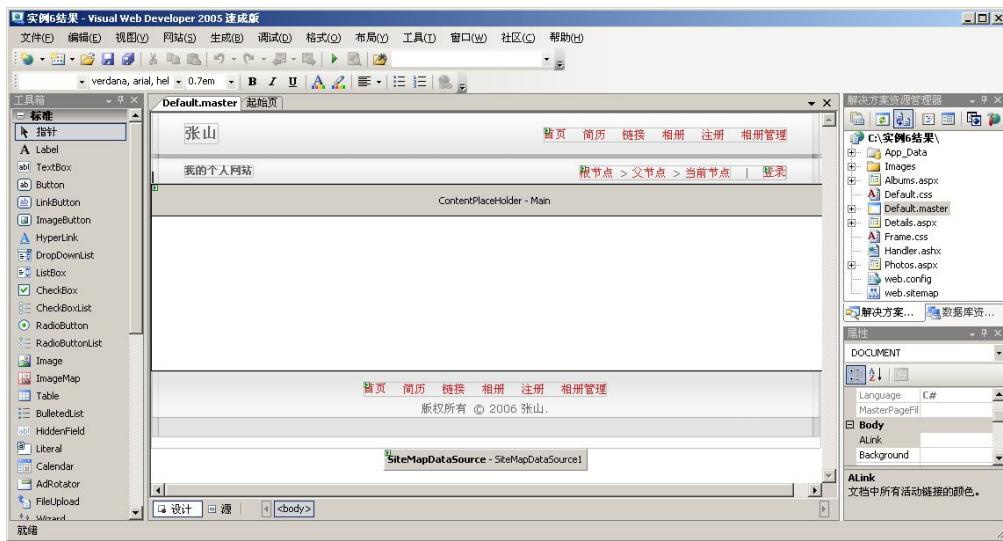


图 4-31 Default.master 的设计视图

### 4.3.1 建立 web.sitemap

在前面的第 1 章中，曾经分析过 PWS 整个网站的层次结构，也就是 11 个页面之间的相互关系，建立站点地图文件 web.sitemap 的目的，就是通过 web.sitemap 文件描述 PWS 整个网站的层次结构。

#### 1. 新建站点地图文件

在 VWD 中，右击 VWD 右边“解决方案资源管理器”窗口中的“实例 6”项目，然后在弹出的快捷菜单中选择“添加新项”命令，在打开的如图 4-32 所示的“添加新项”对话框中，选择“站点地图”模板，此时的文件名默认设置为 web.sitemap，不能修改这个文件的名称，然后单击“添加”按钮，即可在选择的项目下创建一个站点地图文件。

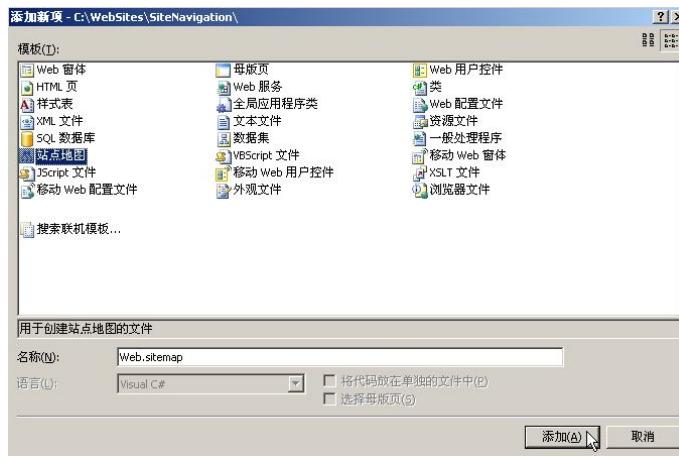


图 4-32 “新加新项”对话框

单击站点地图文件 web.sitemap，查看其中的内容，发现该文件是一个 XML 文件，其中定义了一个简单的框架来描述站点的层次结构。

## 2. 设置站点地图文件

打开站点地图文件 web.sitemap，将其中的语句清空，全部删除掉，然后将代码 4-10 中的 XML 文件粘贴到其中，保存 web.sitemap 即可。

代码 4-10 站点地图文件的代码

```
1: <?xml version="1.0" encoding="gb2312" ?>
2: <siteMap>
3:   <siteMapNode title="首页" url="Default.aspx">
4:     <siteMapNode title="简历" url="Resume.aspx" />
5:     <siteMapNode title="链接" url="Links.aspx" />
6:     <siteMapNode title="相册" url="Albums.aspx" >
7:       <siteMapNode title="照片" url="Photos.aspx" >
8:         <siteMapNode title="详细" url="Details.aspx" />
9:       </siteMapNode>
10:      </siteMapNode>
11:      <siteMapNode title="注册" url="Register.aspx" />
12:      <siteMapNode title="相册管理" url="Admin/Albums.aspx" >
13:        <siteMapNode title="照片" url="Admin/Photos.aspx" >
14:          <siteMapNode title="详细" url="Admin/Details.aspx" />
15:        </siteMapNode>
16:      </siteMapNode>
17:    </siteMapNode>
18:  </siteMap>
```

下面来详细说明这个 web.sitemap 文件。以上的 web.sitemap 描述了 PWS 的页面层次结构，整个网站分为 4 个层次。第 1 个层次是主页，即 Default.aspx 页面；通过 Default.aspx 页面所链接的页面地址是第 2 个层次，它们是 Resume.aspx、Links.aspx、Albums.aspx、Register.aspx 以及管理页面 Admin/Albums.aspx；在第 2 个层次中的 Resume.aspx 页面，可以链接到第 3 个层次中的页面 Photos.aspx，再通过这个 Photos.aspx 页面链接到第 4 个层次的 Details.aspx 页面。同样，在第 2 个层次中的管理页面 Admin/Albums.aspx 页面，可以链接到第 3 个层次中的页面 Admin/Photos.aspx，再通过这个 Admin/Photos.aspx 页面链接到第 4 个层次的 Admin/Details.aspx 页面。

### 4.3.2 使用 SiteMapDataSource

使用 SiteMapDataSource 控件是比较简单的，首先单击 VWD 左边控件工具箱中“数据”控件组下面的 SiteMapDataSource 控件，然后将其拖放到母版 Default.master 页面的下方即可。

SiteMapDataSource 控件不需要像其他数据源控件那样设置其他的参数，这是因为 SiteMapDataSource 控件内部已经绑定将站点地图文件 web.sitemap 作为它的数据源，所以站点地图文件 web.sitemap 的名称不能随意改变，否则 SiteMapDataSource 控件将找不到它的数据源。

### 4.3.3 使用 Menu

使用 Menu 控件也比较简单，单击控件工具箱中 Navigation 控件组下面的 Menu 控件，并将其拖放到母版 Default.master 页面上方的相关位置，然后在 VWD 右下方的属性框中设置相应的属性。

首先设置 Menu 控件的数据源，DataSourceID 当然应该为 SiteMapDataSource1；CssClass 设置为 menua 的样式；然后设置布局方式采用水平的显示方式，Orientation 设置为 Horizontal；要显示的菜单层次数目 StaticDisplayLevels 为 2，以便显示 Home 以及页面上的所有页面地址，图 4-33 是 Menu 控件的属性窗口。

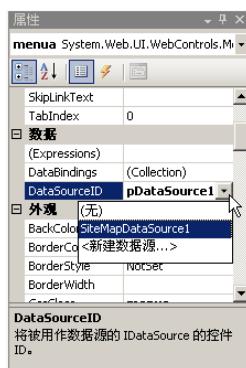


图 4-33 Menu 控件的属性窗口

Menu 控件设置的详细代码如代码 4-11 所示。

#### 代码 4-11 站点 Menu 控件设置的代码

```
1: <asp:menu id="menua" runat="server" datasourceid="SiteMapDataSource1"
2:   cssclass="menua" orientation="Horizontal" maximumdynamicdisplaylevels="0"
3:   skiplinktext="" staticdisplaylevels="2" />
```

对于母版脚部的导航，同样再次建立一个 Menu 控件，设置与上面相同的属性，只是 Menu 控件的名称为 menub，CssClass 的样式为 menub。

### 4.3.4 使用 SiteMapPath

使用 SiteMapPath 控件也相当简单，同样单击 VWD 左边控件工具箱中 Navigation 控件组下面的 SiteMapPath 控件，并将其拖放到母版 Default.master 页面上方的相应位置即可。

SiteMapPath 控件也不需要像其他数据源控件那样设置其他的参数，同样是因为 SiteMapPath 控件内部已经绑定将站点地图文件 web.sitemap 作为它的数据源。

在 VWD 右下方的属性框中可以设置 SiteMapPath 控件的相应属性，设置 SiteMapPath 控件的详细代码如代码 4-12 所示。

#### 代码 4-12 设置站点 SiteMapPath 控件的代码

```
1: <asp:SiteMapPath id="SiteMapPath1" runat="server" PathSeparator=" " 
2:   RenderCurrentNodeAsLink="true" />
```

这里将 SiteMapPath 控件的路径分隔符设置为 “>”，即用符号 “>” 来分开各页面的标题； RenderCurrentNodeAsLink 设置为 true，表明将当前页面的路径地址设置为一个链接地址。

## 4.4 小结

下面对本章的内容作一个小结：

- **母版页：**介绍了在 PWS 网站中，如何使用母版页简化页面的制作与管理。
- **页面导航：**介绍了站点地图 web.sitemap、站点地图数据源 SiteMapDataSource 控件的使用，以及导航控件 TreeView、Menu 和 SiteMapPath 的使用。

## 练习题

1. 母版页中包含另外一个母版页，称之为嵌套母版页，请将 PWS 网站的页面设计改为使用嵌套母版页。
2. 站点地图文件是否表示真实的物理文件存储结构？请使用导航控件 TreeView、Menu 和 SiteMapPath 分别表示站点地图文件。