

第 5 章 页面结构与配置文件

● 单元概述

前面的章节介绍了ASP.NET应用程序的基本语法和服务器控件，本章将重点介绍ASP.NET页面的组成部分，以及全局应用程序文件Global.asax和Web配置文件Web.config，使读者对Web应用程序的开发框架有更深入的了解。本章学习目标：

■掌握ASP.NET页面的组成部分：aspx文件和aspx.cs文件。

■了解全局应用程序文件Global.asax和Web配置文件Web.config在Web应用程序中的用途。

● 项目引入

经过几天的摸索之后，小王对Visual Studio 2010工具箱中一些常用的工具已经能熟练使用了。但是小王发现，每次新建一个网页，都会产生两个文件，这两个文件有什么区别，它们各自用来干什么呢？在运行时，它们又是如何组合在一起来实现功能的？这些问题常常困扰着小王，小王知道，这些问题必须弄清楚，否则就不能真正了解ASP.NET的基本运行原理。

● 项目任务书

项目任务书			
项目名称：	页面结构和配置文件	项目小组：	XX小组
估计工作量：	4学时	起止日期：	2011年**月**日
开发软件：	Visual Studio 2010		
项目任务和范围：	1.创建一个网站ch05，添加一个Web页面Default，弄懂两个文件aspx和aspx.cs文件之间的关系，理解Page指令。 2.理解网站ch05中一些默认添加的文件夹，如App_Data等。 3.为网站ch05添加一个全局应用程序类Global文件，理解其中各个事件的用途。 4.为网站ch05添加一个配置文件Config文件，理解各个节点的用途。		

5.1 aspx文件和aspx.cs文件

创建一个ASP.NET应用程序，实际上创建了两个文件：aspx文件和aspx.cs文件。其中aspx文件为用户设计界面文件，主要表现为显示界面的HTML源码，在开发过程中也可以以设计视图的形式来查看页面设计效果；aspx.cs文件为逻辑代码隐藏文件，包含事件处理方法和自定义方法等。如创建一个登录页面，设计界面Default.aspx如图5-1所示（处于“设计”状态下），控件有Label控件、TextBox控件以及Button控件。



图5-1 “设计”界面

通过单击设计界面下方的“源”可以将设计界面切换到HTML源码编辑窗口，如图5-2所示，每一个“<asp: ></asp: >”标记代表一个Web控件。



图5-2 “源”界面

还有一种是处于“拆分”状态下的界面，这个界面同时显示的设计的控件和源码，如图5-3所示，可以让设计者达到“所见即所得”的效果。



图5-3 “拆分”界面

可以通过解决方案资源管理器打开登录页面的Default.aspx.cs文件，如图5-4所示，aspx.cs文件中是以C#语言编写的逻辑处理代码，包括事件处理方法和自定义的一些方法。



图5-4 打开Default.aspx.cs文件

当用户浏览请求页面时，Web服务器自动将逻辑代码文件aspx.cs文件和布局代码文件组合在一起。组合的代码如下：

例程01 代码位置：..\ch05\Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
```

页面通过page指令联系在一起。设置page指令的属性可以设置页面间的连接信息，每个aspx文件只能包含一条@page指令，上述代码分别设置了page指令的以下几个属性：

- (1) Language属性：用来指定在页面和代码声明块中进行编译时使用的语言，包括C#、Visual Basic、Jscript.NET等.NET支持的语言。
- (2) AutoEventWireup属性：用来指示页面的事件是否自动连接代理，值为true表示自动连接，值为false表示不自动连接。
- (3) CodeFile属性：用来指定包含与页面关联的类文件的名称，如上例中与登录页面关联的类文件为Default.aspx.cs，该属性由Visual Studio 2010 Web设计器自动设置。
- (4) Inherits属性：用来指定供页面继承的代码隐藏类。

注意：CodeFile属性用来指定与页面关联的文件，而Inherits属性用来指定与页面关联的类，如在创建一个登录页面时，与此页面相关联的类文件为Default.aspx.cs文件（CodeFile属性值），此文件的内容为：

```
//*****  
public partial class _Default : System.Web.UI.Page  
{  
    ... ..  
}  
//*****
```

从Default.aspx.cs中可以看出其类名为_Default(Inherits属性值)。

在Visual Studio 2010中,用户也可以根据需求不添加.cs文件,具体操作如下:在添加新Web页面时,在弹出的【添加新项】对话框中,取消选中如图5-5所示的【将代码放在单独的文件中(P)】复选框,此时新建的网页就只有文件Default2.aspx,不包含Default2.aspx.cs文件。

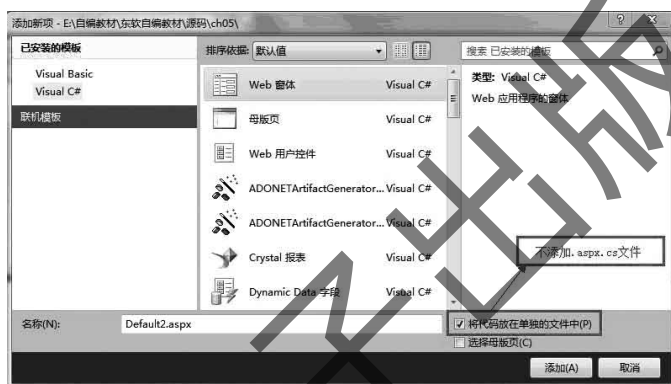


图5-5 不添加.cs文件

没有逻辑代码文件.aspx.cs的页面,其page指令只有Language属性,其代码为:

```
<%@ Page Language="C#" %>
```

单击左下角的【源】,打开HTML源码,发现多了一个“<script></script>”标记,用户自定义的方法以及事件都将放在此标记内。

设计如图5-6所示的页面,页面中的控件包括TextBox控件、Button控件和Label控件。

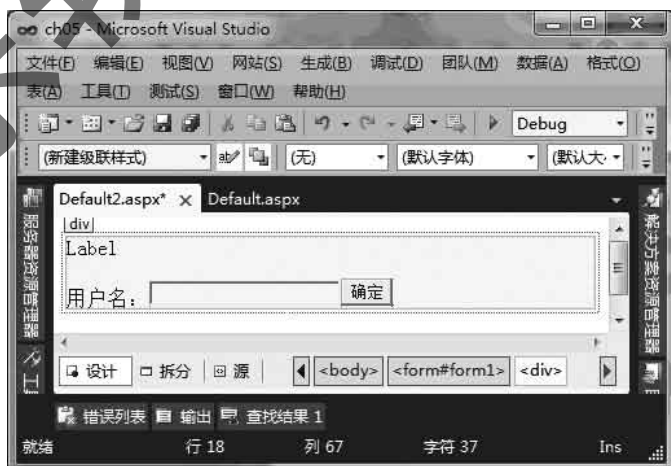


图5-6 设计页面

双击【提交】按钮，页面将打开HTML源码，如图5-7所示。

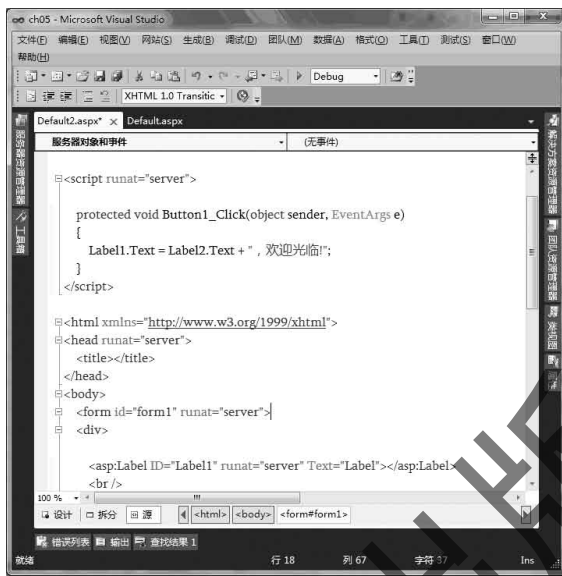


图5-7 HTML源码事件

从操作中可以发现，设计页面和逻辑代码页面放在了同一个页面，这种方式较适合于ASP程序员，缺点是因为控件代码和方法事件代码全部混在一起，容易出错且不利于查找，建议初学者不要选择这种方式。

5.2 ASP.NET默认文件夹

ASP.NET应用程序可以添加如下8个默认文件夹，每个文件夹都存放ASP.NET应用程序的不同类型的资源。

(1) Bin文件夹：包含应用程序所需的、用于控件、组件或者需要引用的任何其他代码的可部署程序集(.dll)。该目录中存在的任何.dll文件将自动地连接到应用程序。

(2) APP_Code文件夹：包含页面使用的类的源代码，用户自定义的类将自动添加到此文件夹中。

(3) APP_GlobalResources文件夹：包含编译到具有全局范围的程序集的资源(.resx文件和.resources文件)。

(4) APP_LocalResources文件夹：包含与应用程序中的特定页、用户控件或母版页相关联的资源(.resx文件和.resources文件)。

(5) APP_WebReferences文件夹：包含用于定义在应用程序中使用的Web引用的引用协定文件、架构文件和发现文档文件。

(6) APP_Browsers文件夹：描述浏览器（不管是移动设备浏览器，还是台式机浏览器）的特征和功能。ASP.NET在安装路径下的Config\Browser文件夹中安装了大量.browser文件，这些文件供所有应用程序共享。

(7) APP_Data文件夹：包含应用程序的本地数据存储。它通常以文件(诸如Microsoft Access或Microsoft SQL Server Express数据库、XML文件、文本文件以及应用程序支持的任何其他文件)形式包含数据存储。该文件夹的内容不由ASP.NET处理。该文件夹是ASP.NET提供程序存储自身数据的默认位置。

(8) 主题文件夹：包含一组定义ASP.NET页面和控件外观的文件(skin、.css、.xsl、附属文件)。

其中APP_Data文件夹在建网站的同时已经自动创建，而其他7个文件夹则需要手动添加。在解决方案资源管理器中，右击站点文件，弹出如图5-8所示的快捷菜单，选择对应的文件夹进行添加。

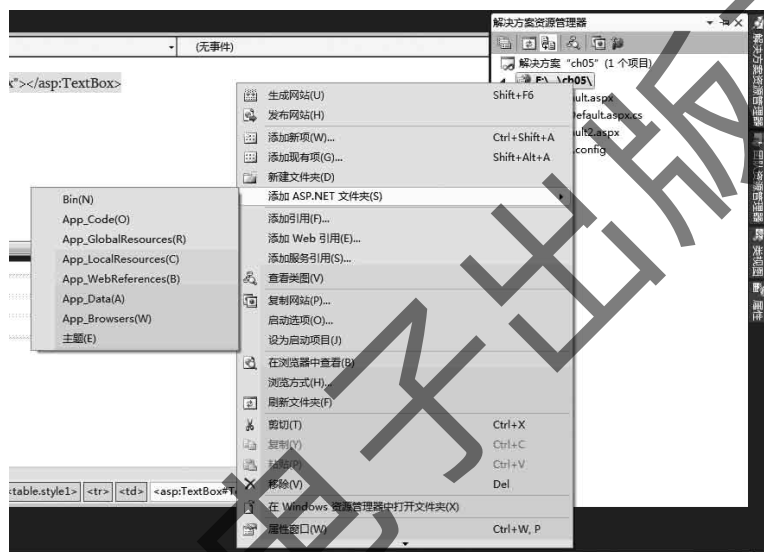


图5-8 添加ASP.NET文件夹

5.3 配置文件Web.config

Web.config文件是可选的，用来储存ASP.NET Web应用程序的配置信息(如设置ASP.NET Web应用程序的身份验证方式)。它是一个XML文件，由标记和属性组成。Web.config文件位于根目录之下，如果是第一次运行Web应用程序，将弹出【未启用调试】对话框，如图5-9所示。

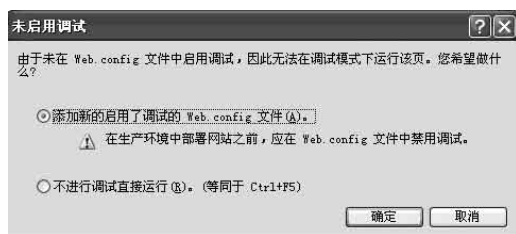


图5-9 【未启用调试】对话框

若选中【添加新的启用了调试的Web.config文件(A)】单选按钮，单击【确定】按钮，将为应用程序自动添加Web.config文件。

还可以在解决方案资源管理器中，右击站点文件，在弹出的快捷菜单中选择【添加新项】命令，在弹出的对话框中选择【Web配置文件】选项，如图5-10所示，单击【添加】按钮，将Web配置文件添加到Web应用程序中。



图5-10 添加Web配置文件

双击打开Web.config配置文件，其基本结构和内容如图5-11所示。

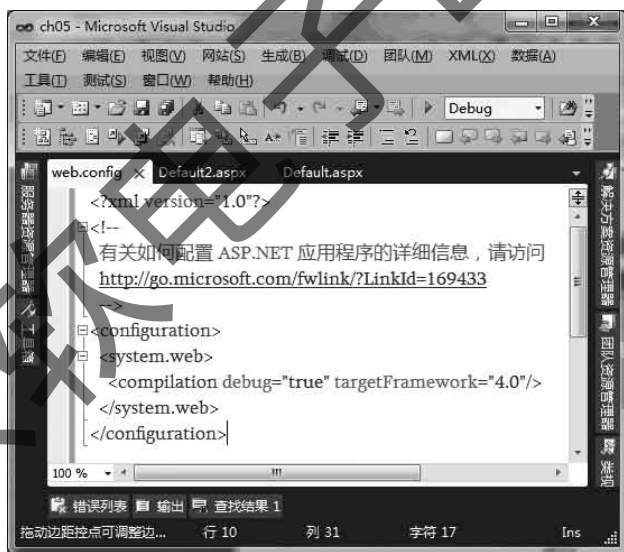


图5-11 Web.config配置文件

Web.config配置文件(默认的配置设置)以下所有的代码都应该位于节“<configuration><system.Web>”和“</system.Web></configuration>”之间，即configuration为根元素，其他的节都包含在它的内部。ASP.NET定义了一些标准节，所谓标准节就是系统已经定义好的节，可以直接使用。下面将详细介绍几个Web.config文件中常用的节。

1. <authentication>节

<authentication>节用来配置 ASP.NET 身份验证支持（为Windows、Forms、PassPort、

None四种)。该节只能在计算机、站点或应用程序的级别声明。<authentication>节必须与<authorization>节配合使用。

如下配置表示基于窗体(Forms)的身份验证配置站点,当没有登录的用户访问需要身份验证的网页时,网页将自动跳转到登录网页。

```
<authentication mode="Forms" >
  <forms loginUrl="logon.aspx" name=".FormsAuthCookie"/>
</authentication>
```

其中元素loginUrl表示登录网页的名称, name表示Cookie名称。

2. <authorization>节

<authorization>节用来控制对 URL 资源的客户端访问(如允许匿名用户访问)。此节可以在任何级别(计算机、站点、应用程序、子目录或页)上声明。必须与<authentication>节配合使用。

以下配置可以禁止匿名用户的访问:

```
<authorization> <deny users="?" />
</authorization>
```

可以使用User.Identity.Name 来获取已经通过验证的当前的用户名;使用 Web.Security.FormsAuthentication.RedirectFromLoginPage方法将已验证的用户重定向到用户刚才请求的页面。

3. <sessionState>节

<sessionState>节用来为当前应用程序配置会话状态设置(如设置是否启用会话状态,会话状态保存位置)。

示例:

```
<sessionState mode="InProc" cookieless="true" timeout="20"/>
</sessionState>
```

mode="InProc"表示:在本地储存会话状态(你也可以选择储存在远程服务器或SAL服务器中或不启用会话状态)。

cookieless="true"表示:如果用户浏览器不支持Cookie时启用会话状态(默认为False)。

timeout="20"表示:会话可以处于空闲状态的分钟数。

4. <customErrors>节

<customErrors>节用来为 ASP.NET应用程序提供有关自定义错误信息的信息。它不适用于XML Web Services 中发生的错误。

示例:当发生错误时,将网页跳转到自定义的错误页面。

```
<customErrors defaultRedirect="ErrorPage.aspx" mode="RemoteOnly">
</customErrors>
```

其中元素defaultRedirect表示自定义的处理错误的网页名称, mode元素表示对不在本地Web服务器上运行的用户显示自定义(友好的)信息。

5. <httpRuntime>节

<httpRuntime>节用来配置 ASP.NET HTTP 运行库设置。该节可以在计算机、站点、应用程序和子目录级别声明。

如下配置用来控制用户上传文件最大为4M，最长时间为60秒，最多请求数为100。

```
<httpRuntime maxRequestLength="4096" executionTimeout="60"
appRequestQueueLimit="100"/>
```

在标准节中，有一个appSettings节允许自定义简单的设置，它有一个add标记，这个标记可以使用代码进行访问，如在Web.config文件中配置了如下内容：

```
<appSettings>
    <add key="ConnectionString" value="Data
Source=.;Database=BBSDB;Integrated Security=SSPI"/>
</appSettings>
```

在上面的节中配置了数据库连接字符串，如果应用程序的数据库连接有所改变，只需改变此配置即可，具有很大的灵活性。

若要访问Web.config文件的内容，可以通过使用ConfigurationSettings.AppSettings 静态字符串集合来访问，如下代码用来获取上面例子中建立的连接字符串。

```
SqlConnection con = new
SqlConnection(ConfigurationSettings.AppSettings["ConnectionString"]);
```

5.4 Global.asax文件

Global.asax 文件有时候也叫做 ASP.NET全局应用程序文件，提供了一种在一个中心位置响应应用程序级或模块级事件的方法。可以使用Global.asax文件实现应用程序的安全性以及其他一些任务。Global.asax 位于应用程序根目录下，Global.asax 文件继承自HttpApplication 类，它维护一个HttpApplication 对象池，并在需要时将对象池中的对象分配给应用程序。Global.asax 文件包含以下事件。

(1) Application_Init: 在应用程序被实例化或第一次被调用时，该事件被触发。对于所有的HttpApplication 对象实例，它都会被调用。

(2) Application_Disposed: 在应用程序被销毁之前触发。这是清除以前所用资源的理想位置。

(3) Application_Error: 当应用程序中遇到一个未处理的异常时，该事件被触发。

(4) Application_Start: 在HttpApplication 类的第一个实例被创建时，该事件被触发。它允许你创建可以由所有HttpApplication 实例访问的对象。

(5) Application_End: 在HttpApplication 类的最后一个实例被销毁时，该事件被触发。在一个应用程序的生命周期内它只被触发一次。

(6) Application_BeginRequest: 在接收到一个应用程序请求时触发。对于一个请求来说，它是第一个被触发的事件，请求一般是用户输入的一个页面请求(URL)。

(7) Application_EndRequest: 针对应用程序请求的最后一个事件。

(8) Session_Start: 在一个新用户访问应用程序 Web 站点时，该事件被触发。

(9) Session_End: 在一个用户的会话超时、结束或它们离开应用程序 Web 站点时，该事

件被触发。

添加Global.asax文件的方法如下：

在解决方案资源管理器中，右击站点文件，在弹出的快捷菜单中选择【添加新项】命令，在弹出的对话框中选择【全局应用程序类】选项，如图5-12所示，单击【添加】按钮，将Global.asax文件添加到Web应用程序中。



图5-12 添加Global.asax文件

此时在Global.asax文件中，自动地添加了一些用来处理应用程序的事件，如图5-13所示。

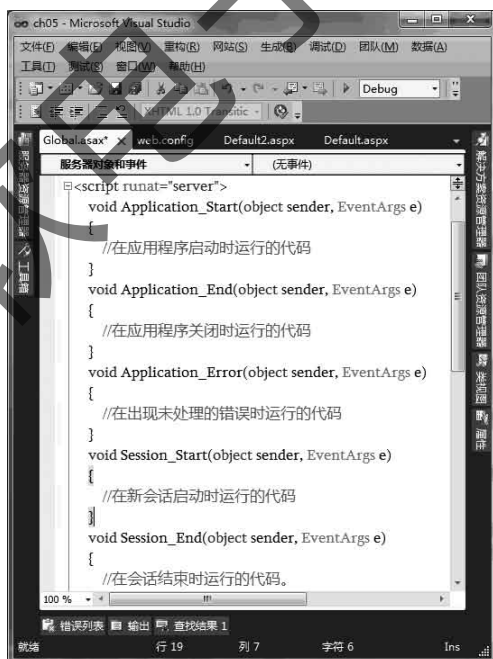


图5-13 在Global.asax文件中添加的事件

小 结

本章主要介绍了ASP.NET页面的结构，Web配置文件Web.config和全局应用程序文件Global.asax。首先详细介绍了代码隐藏技术，代码分离技术，然后介绍了ASP.NET的配置，重点介绍了ASP.NET的几个常用的配置设置。

本章重点和难点：

(1) 掌握ASP.NET页面的组成部分：aspx文件和aspx.cs文件，了解代码隐藏技术和代码分离技术。

(2) 配置文件Web.config的创建，以及常用的配置设置。

(3) Global.asax文件的创建，以及包含的常用事件。

项目训练

1. 选择题

(1) 在Page指令中用来指定供页面继承的代码隐藏类的属性是()。

A. Language

B. CodeFile

C. Inherits

D. AutoEventWireup

(2) 应用程序所需要的组件(.dll文件)一般存储在ASP.NET的()文件夹中。

A. Bin

B. APP_Code

C. APP_Browsers

D. APP_Data

(3) 下面哪一个配置可以用来为 ASP.NET应用程序提供有关自定义错误信息的信息()。

A. <sessionState>节

B. <authentication>节

C. <customErrors>节

D. <httpRuntime>节

2. 填空题

(1) 创建一个Web应用程序，实际上创建了两个文件：_____和_____。

(2) Global.asax 位于应用程序_____下，Global.asax 文件继承自_____类，它维护一个HttpApplication 对象池，并在需要时将对象池中的对象分配给应用程序。

(3) Web.config配置文件中所有的节都包含在_____内部。

3. 问答题

(1) 简述配置文件Web.config的常用配置。

(2) 简述Global.asax文件包含的常用事件。

4. 上机操作题

添加一个配置文件Web.config，并添加连接数据库的字符串。