

开发人员专业技术丛书

ASP.NET技术内幕

(美) Stephen Walther 著
马朝晖 等译



机械工业出版社
China Machine Press

ASP的最新版本ASP.NET是Microsoft用于建立动态的数据库驱动网站的技术。内容包括: ASP.NET Web表单的使用, 高级ASP.NET页面的开发, ADO.NET的使用, ASP.NET应用程序的使用, ASP.NET应用程序的保护, ASP.NET Web服务的建立, .NET框架的利用, 自定义的ASP.NET控件的建立, ASP.NET的应用程序示例等。

本书内容丰富、图文并茂, 适合于需要创建网站的专业程序员阅读。本书是关于使用ASP.NET建立网站的完整参考书, 书中包含数百个代码示例, 读者可以使用这些示例开始建立自己的网站。

Stephen Walther: ASP.NET Unleashed.

Authorized translation from the English language edition published by Sams, an imprint of Macmillan Computer Publishing U.S.A.

Copyright © 2002 by Sams. All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2002 by China Machine Press.

本书中文简体字版由美国麦克米兰公司授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

本书版权登记号: 图字: 01-2002-0814

图书在版编目(CIP)数据

ASP.NET技术内幕/(美)沃瑟(Walther, S.)著;马朝晖等译.-北京:机械工业出版社, 2002.8

(开发人员专业技术丛书)

书名原文: ASP.NET Unleashed

ISBN 7-111-10440-4

I. A… II. ①沃…②马… III. 主页制作-程序设计 IV. TP393.092

中国版本图书馆CIP数据核字(2002)第039485号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 周宜今 张鸿斌

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年8月第1版第1次印刷

787mm×1092mm 1/16·63.75印张

印数: 0 001-4 000册

定价: 99.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

前 言

ASP的最新版本ASP.NET是Microsoft用于建立动态的数据库驱动网站的技术。

ASP是在建立可扩展的交互式网站方面最流行的语言之一。Internet上客流量最大的几个网站采用了ASP, 例如Dell Online、Barnes and Noble、1-800-Flowers和Microsoft自己的网站。

ASP.NET与以前的ASP版本差异很大。如果你曾经使用以前的ASP版本进行编程, 而且没有见识过ASP.NET的新特性, 那么准备好大吃一惊吧!

下面列出了ASP.NET的几个主要的新特性:

- ASP.NET使用CLR语言(比如Visual Basic和C#)编写的编译代码。与以前的ASP版本不同, 这个版本不使用解释性的脚本语言, 比如VBScript。
- ASP.NET页面是由服务器端控件组成的。Web服务器控件允许使用内置的对象模型对HTML元素进行表示和编程。
- ASP.NET包含一种称为Web服务的新技术。可以使用Web服务通过Internet访问方法和属性以及传输数据库数据。
- ASP.NET是Microsoft的.NET框架的一部分。在你的代码中可以使用数千个.NET类以执行各种各样的任务, 比如动态产生图像以及将数组保存进文件。
- ASP.NET包含了页面和数据缓存机制, 使用它可以很容易地明显提高网站的性能。

本书面向的读者

本书适合于那些需要创建网站的专业程序员阅读。本书是关于使用ASP.NET建立网站的完整参考书。本书包含数百个代码示例, 你可以使用这些示例开始建立自己的网站。

如果你是ASP网站建设方面的新手, 那么通过阅读本书你可以学到用ASP.NET建立网站所需的所有知识。如果你是有经验的ASP程序员, 可以通过本书了解ASP.NET的新特性。

本书的第九部分包含两个完整的示例应用程序: 一个职业介绍网站和一个在线商店。

在阅读本书之前你需要知道什么

本书假设你了解HTML而且熟悉一种编程语言, 比如Visual Basic (VB) 或VBScript。

本书中讨论的几乎所有代码都是使用Visual Basic编写的。但是, 如果你熟悉VBScript, 那么你应该可以毫不费力地理解代码示例。

为了充分理解有关数据库的章节, 你应该具有一定的数据库使用经验, 比如Microsoft SQL Server、Oracle或Microsoft Access。

本书是如何组织的

尽管我鼓励你从头到尾逐章地阅读本书, 但是我想并不是每个人都有时间这么做。如果需

要的话，可以只将本书作为参考书使用，在你需要的时候直接跳到相关的章节。因此，介绍一下本书的整体组织结构对你可能有帮助：

- 第一部分：使用ASP.NET Web表单——这一部分中的各章详细说明如何使用表单和检验控件建立ASP.NET页面。这里讨论了构建交互式HTML表单所使用的所有控件。
- 第二部分：高级ASP.NET页面开发——这几章讨论如何使用用户控件创建自己的控件，如何使用可移动控件创建供手机和PDA使用的网页，如何将代码与页面的设计内容分离，以及如何在ASP.NET页面中使用第三方控件。
- 第三部分：使用ADO.NET——这一部分中的各章详细讲解ADO.NET的知识。你将学习如何执行基本的数据库任务（比如将表单数据保存进数据库），以及如何使用特殊的数据库Web控件（比如DataList和DataGrid控件）。
- 第四部分：使用ASP.NET应用程序——这几章讨论ASP.NET应用程序的特性。例如，如何使用Web.config文件配置ASP.NET应用程序，以及使用应用程序状态和会话状态。
- 第五部分：保护ASP.NET应用程序——这几章概述ASP.NET内置的安全特性。例如，如何使用基于表单的安全性对网站的页面进行口令保护，以及如何使用.NET类执行加密。
- 第六部分：建立ASP.NET Web服务——这一部分中的各章解释如何使用ASP.NET中的Web服务技术。你将学到如何创建Web服务并且将Web服务集成进你的网站。
- 第七部分：利用.NET框架——这几章讨论可以在页面中使用的许多有用的.NET类。例如，如何动态地产生图像，使用集合和正则表达式，以及从ASP.NET页面发送电子邮件。
- 第八部分：建立自定义的ASP.NET控件——这几章讨论如何使用自定义控件扩展ASP.NET框架。你将学习如何建立自己的ASP.NET控件。
- 第九部分：ASP.NET应用程序示例——本书的这一部分包含两个完整的示例应用程序：一个职业介绍网站和一个在线商店。

本书中使用的代码命名惯例

变量命名和代码惯例是个不讨人喜欢的主题。程序员喜欢制定自己的命名惯例，而且讨厌别人对他们应该如何写代码指手画脚。

无论如何，当代码必须由多个人编写时，制定公用的惯例是很有意义的。有了公用的惯例才能使你以后重读自己的代码更方便。

最流行的变量命名惯例被称为匈牙利表示法（Hungarian Notation）。它最初是由Microsoft的Charles Simonyi开发的，而且在Microsoft内部使用了多年。根据这个惯例，所有变量名的开头应该是标准的三个或四个字母的前缀，这个前缀代表变量的数据类型。例如，用于表示顾客的年龄的整数变量应该命名为intCustomerAge。

Microsoft建议程序员在使用.NET框架和ASP.NET时不使用这种惯例。这个建议的意图是他们希望你使用高级的编辑器编写代码，比如Microsoft Visual Studio。Visual Studio会自动地为你提供关于变量类型的信息。

但是，在本书中，我没有推荐你必须使用Visual Studio编写ASP.NET页面。实际上，程序员用来写代码的编辑器非常多，比如TextPad、UltraEdit和Notepad。因此，规定变量命名惯例就是必不可少的了。

而且，根据我的经验，许多公司要求自己的程序员遵循一种严格的命名惯例，这些惯例通常是匈牙利表示法的变体。我在本书中使用的命名惯例也是匈牙利表示法的一种变体。我为基本的变量数据类型、ASP.NET Web控件、ADO.NET类和事件处理子例程创建了一套前缀和命名惯例。这些惯例见下面的表格。

我的目标并不是为.NET框架中的每个类提供一个标准的前缀。那需要超过3400个标准前缀！我的目标只是为在ASP.NET中最常使用的类型提供前缀。

变量命名惯例

类 型	前 缀	示 例
Array	arr	arrShoppingList
Boolean	bln	blnIsPostBack
Byte	byt	bytPixelValue
Char	chr	chrDelimiter
DateTime	dtm	dtmStartDate
Decimal	dec	decAverageHeight
Double	dbl	dblSizeOfUniverse
Integer	int	intRowCounter
Long	lng	lngBillGatesIncome
Object	obj	objReturnValue
Short	shr	shrAverage
Single	sng	sngMaximum
String	str	strFirstName

WebControls

类 型	前 缀	示 例
AdRotator	adrt	adrtTopAd
Button	btn	btnSubmit
Calendar	cal	calMeetingDates
CheckBox	chk	chkBlue
CheckBoxList	chk1	chk1FavColors
CompareValidator	valc	valcValidAge
CustomValidator	valx	valxDBCheck
DataGrid	dgrd	dgrdTitles
DataList	dlst	dlstTitles
DropDownList	drop	dropCountries
HyperLink	lnk	lnkDetails
Image	img	imgAuntBetty
ImageButton	ibtn	ibtnSubmit
Label	lbl	lblResults

(续)

类 型	前 缀	示 例
LinkButton	lbtn	lbtnSubmit
ListBox	lst	lstCountries
Panel	pnl	pnlForm2
Placeholder	plh	plhFormContents
RadioButton	rad	radFemale
RadioButtonList	radl	radlGender
RangeValidator	valg	valgAge
RegularExpressionValidator	vale	valeEmail
Repeater	rpt	rptQueryResults
RequiredFieldValidator	valr	valrFirstName
Table	tbl	tblCountryCodes
TableCell	tblc	tblcGermany
TableRow	tblr	tblrCountry
TextBox	txt	txtFirstName
ValidationSummary	vals	valsFormErrors
XML	xmlc	xmlcTransformResults

ADO.NET

类 型	前 缀	示 例
Connection	con	conNorthwind
Command	cmd	cmdReturnProducts
Parameter	parm	parmProductID
DataAdapter	dad	dadProducts
DataReader	dtr	dtrProducts
DataSet	dst	dstNorthwind
DataTable	dtbl	dtblProduct
DataRow	drow	drowRow98
DataColumn	dcol	dcolProductID
DataRelation	drel	drelMasterDetail
DataView	dvw	dvwFilteredProducts

事件处理子例程

事件处理子例程的名称包括引发此事件的控件ID，后面跟着被处理的事件的类型。例如，名为btnSubmit_Click的子例程负责处理名为btnSubmit的Button控件的Click事件。

如果引发事件的控件没有被分配ID，那么使用控件的类型替代ID。例如，名为Button_Click的子例程负责处理没有ID的Button控件的Click事件。

本书中使用的印刷惯例

本书中使用了以下印刷惯例：

- 当一个代码行实际上是上一行代码的延续时，在这行代码前使用图标。有时代码行太长，无法放在书中的一行内。如果你看到代码行前面有图标，就要记住它是上一行的一部分。
- 本书包含注意、提示和警告，以便帮助你更快地找到重要而有价值的信息。
- 有经验的ASP用户如果打算升级到ASP.NET，那么请特别注意“ASP传统提示”。

要获取本书实例代码，请访问原书出版社网站。

英文原书书名：ASP.NET Unleashed

英文原书书号：ISBN 0-672-32068-1

英文原书出版社网址：www.sampublishing.com

目 录

前言

第一部分 使用ASP.NET Web表单

第1章 建立ASP.NET页面	1
1.1 ASP.NET和.NET框架	1
1.1.1 .NET框架类库	1
1.1.2 理解名称空间	2
1.1.3 标准的ASP.NET名称空间	2
1.1.4 与.NET框架兼容的语言	3
1.2 ASP.NET控件简介	4
1.2.1 简单的ASP.NET页面	4
1.2.2 ASP.NET控件的优点	7
1.2.3 ASP.NET控件概述	12
1.3 向ASP.NET页面中添加应用逻辑	17
1.3.1 处理控件事件	17
1.3.2 处理页面事件	21
1.4 ASP.NET页面的结构	24
1.4.1 指令	24
1.4.2 代码声明块	27
1.4.3 ASP.NET控件	28
1.4.4 代码显示块	28
1.4.5 服务器端注释	29
1.4.6 服务器端包含指令	30
1.4.7 文本和HTML标记	30
1.5 小结	31
第2章 用Web服务器控件建立表单	32
2.1 建立智能表单	32
2.1.1 Label控件	32
2.1.2 TextBox控件	34
2.1.3 Button控件	40
2.1.4 RadioButton和RadioButtonList控件	47
2.1.5 CheckBox和CheckBoxList控件	61

2.1.6 DropDownList控件	73
2.1.7 ListBox控件	82
2.2 控制页面导航	91
2.2.1 将表单提交到另一个页面	91
2.2.2 使用Redirect()方法	93
2.2.3 使用HyperLink控件	94
2.3 在控件上进行格式化	95
2.3.1 基本Web控件属性	95
2.3.2 在Web控件上应用样式	99
2.4 小结	104
第3章 用检验控件执行表单的检验	106
3.1 使用客户端检验	106
3.1.1 配置客户端检验	107
3.1.2 启用和禁用客户端检验	107
3.2 必填域: RequiredFieldValidator控件	107
3.3 检验表达式: RegularExpressionValidator 控件	112
3.3.1 检验电子邮件地址	114
3.3.2 检验用户名和口令	115
3.3.3 检验电话号码	117
3.3.4 检验网址	118
3.3.5 检验条目长度	120
3.3.6 检验邮政编码	122
3.4 比较值: CompareValidator控件	123
3.4.1 比较一个控件与另一个控件的值	124
3.4.2 比较一个控件的值与固定值	125
3.4.3 执行数据类型检查	126
3.5 检查值的范围: RangeValidator控件	127
3.6 错误汇总: ValidationSummary控件	131
3.7 执行自定义的检验: CustomValidator 控件	135
3.8 禁用检验	141

3.9 小结	142	页面	219
第4章 高级控件编程	143	6.3.4 编译完整的ASP.NET页面	222
4.1 使用视图状态	143	6.4 小结	224
4.1.1 禁用视图状态	143	第7章 用可移动控件为可移动设备服务	225
4.1.2 将值添加到视图状态	146	7.1 使用可移动设备软件模拟器	225
4.2 显示和隐藏内容	147	7.2 无线应用协议简介	226
4.2.1 使用Visible和Enabled属性	147	7.3 建立WML页面	227
4.2.2 使用Panel控件	150	7.3.1 配置IIS	227
4.2.3 模拟多页面表单	152	7.3.2 WML和XML	227
4.2.4 通过程序添加控件	155	7.3.3 创建卡片集	227
4.2.5 将控件添加到页面	157	7.3.4 用WML链接文件	229
4.2.6 Placeholder控件	158	7.4 使用ASP.NET可移动控件	229
4.2.7 动态产生表单	159	7.4.1 创建可移动表单	230
4.2.8 动态产生列表条目	161	7.4.2 动态激活可移动表单	231
4.3 使用复杂控件	164	7.4.3 显示文本	232
4.3.1 使用Calendar控件显示交互式的 日历	164	7.4.4 显示列表	235
4.3.2 使用AdRotator控件显示广告条	173	7.4.5 创建文本框	245
4.3.3 使用HTMLInputFile控件接受文件 上传	177	7.4.6 检验用户输入	247
4.4 小结	179	7.4.7 显示图像	249
第二部分 高级ASP.NET页面开发		7.4.8 进行电话呼叫	250
第5章 用用户控件创建自定义控件	181	7.4.9 用可移动控件显示广告	250
5.1 用用户控件包含标准内容	181	7.4.10 使用可移动控件显示日历	251
5.2 显露用户控件中的属性和方法	184	7.5 创建跨设备兼容的可移动页面	253
5.3 显露用户控件中的Web控件	187	7.5.1 探测设备功能	253
5.4 显露用户控件中的事件	195	7.5.2 使用DeviceSpecific选择设备	255
5.5 通过程序装载用户控件	195	7.5.3 使用表单模板集	256
5.6 小结	200	7.6 小结	258
第6章 将代码与表示分离	202	第8章 使用第三方控件	259
6.1 创建业务组件	202	8.1 使用Microsoft Internet Explorer WebControls	259
6.2 创建多层Web应用程序	211	8.1.1 使用TreeView控件	259
6.3 使用code-behind	216	8.1.2 使用ToolBar控件	271
6.3.1 code-behind是如何工作的	218	8.1.3 使用TabStrip控件	277
6.3.2 编译code-behind文件	218	8.2 使用Superexpert控件	282
6.3.3 从一个code-behind文件派生多个		8.2.1 使用Content Rotator控件	282
		8.2.2 使用SuperDataGrid控件	289
		8.2.3 使用DataForm控件	292

8.3 小结	300	第11章 使用DataList和DataGrid控件	368
第三部分 使用ADO.NET		11.1 DataList和DataGrid控件概述	
第9章 ADO.NET简介	301	11.1.1 理解事件冒泡机制	368
9.1 ADO.NET概述	301	11.1.2 使用模板	369
9.2 执行常见的数据库任务	302	11.1.3 使用DataKeys集合	370
9.2.1 打开数据库链接	303	11.2 使用DataList控件	370
9.2.2 从数据库表获取记录	305	11.2.1 在DataList中显示数据	370
9.2.3 在查询中使用参数	310	11.2.2 对DataList使用模板	372
9.2.4 向数据库添加记录	314	11.2.3 对DataList使用样式	374
9.2.5 更新数据库记录	317	11.2.4 在DataList中创建多个列	376
9.2.6 删除数据库记录	321	11.2.5 捕获DataList控件中引发的事件	378
9.3 改进数据库性能	324	11.2.6 选择DataList中的条目	381
9.3.1 使用SQL存储过程	324	11.2.7 通过DataList控件使用DataKeys 集合	383
9.3.2 获取返回值和输出参数	327	11.2.8 用DataList控件创建主/明细表单	383
9.3.3 执行复杂的存储过程	329	11.2.9 编辑DataList控件中的条目	386
9.3.4 用链接缓冲改进性能	332	11.3 使用DataGrid控件	392
9.4 高级的数据库主题	333	11.3.1 在DataGrid控件中创建列	394
9.4.1 在事务中执行数据库命令	333	11.3.2 对DataGrid使用样式	406
9.4.2 指定命令行为	336	11.3.3 对DataGrid控件中的列进行排序	408
9.4.3 获取表模式信息	338	11.3.4 对DataGrid中的记录进行分页	411
9.5 小结	339	11.3.5 选择DataGrid控件中的行	416
第10章 将数据绑定到Web控件	340	11.3.6 编辑DataGrid控件中的条目	418
10.1 数据绑定概述	340	11.3.7 使用模板编辑DataGrid控件中的 条目	421
10.2 将服务器控件绑定到数据源	345	11.4 小结	424
10.2.1 绑定到Repeater控件	345	第12章 操作DataSet	425
10.2.2 使用模板	346	12.1 理解DataSet	425
10.2.3 视图状态和Repeater控件	349	12.1.1 DataSet的元素	426
10.2.4 绑定到DropDownList控件	350	12.1.2 向DataSet添加DataTable	426
10.2.5 绑定到RadioButtonList控件	351	12.1.3 将控件绑定到DataSet	429
10.2.6 绑定到CheckBoxList控件	353	12.2 理解DataTable	430
10.2.7 绑定到ListBox控件	354	12.2.1 通过程序创建DataTable	430
10.2.8 绑定到其他控件	355	12.2.2 设置DataTable属性	431
10.3 创建主/明细表单	361	12.2.3 在DataTable中设置列属性	433
10.3.1 创建单页面的主/明细表单	361	12.2.4 在DataTable中计算列值	441
10.3.2 创建多页面的主/明细表单	364	12.2.5 在DataTable之间定义关系	442
10.4 小结	367		

12.2.6 获取DataTable模式信息	444
12.2.7 在DataTable中更新记录	446
12.3 理解DataView	449
12.3.1 在DataView中对行进行过滤	449
12.3.2 在DataView中对行进行排序	452
12.3.3 在DataView中搜索行	453
12.4 通过ASP.NET页面使用DataSet	454
12.4.1 缓存DataSet	455
12.4.2 显示缓存的数据	455
12.4.3 过滤缓存的数据	457
12.4.4 在缓存的DataSet中搜索行	459
12.5 小结	462
第13章 操作XML	463
13.1 XML类概述	463
13.2 通过DataSet使用XML	464
13.2.1 将XML文档读入DataSet	464
13.2.2 通过ReadXml使用模式	466
13.2.3 从DataSet写XML文档	470
13.2.4 通过DataSet使用XMLData Documents	472
13.3 使用样式单转换XML	476
13.4 使用ASP.NET Xml控件	476
13.5 使用XslTransform类	479
13.6 使用强类型的DataSet	480
13.7 小结	484
第14章 使用ADO.NET创建搜索页面	485
14.1 使用SQL Server全文搜索	485
14.1.1 配置Full-Text Search Service	485
14.1.2 对数据库数据执行任意文本查询	487
14.1.3 对数据库数据执行布尔查询	491
14.1.4 上传二进制文档并且进行索引	497
14.2 使用Microsoft Indexing Service	503
14.2.1 配置Microsoft Indexing Service	503
14.2.2 将SQL Server配置为使用Microsoft Indexing Service	504
14.2.3 获取文档属性	505
14.2.4 对文件系统数据执行任意文本	

查询	506
14.2.5 对文件系统数据执行布尔查询	509
14.2.6 对文档属性执行查询	510
14.3 小结	511

第四部分 使用ASP.NET应用程序

第15章 创建ASP.NET应用程序	513
15.1 ASP.NET应用程序概述	513
15.2 使用应用程序状态	514
15.2.1 理解应用程序状态和同步	515
15.2.2 使用Global.asax文件	517
15.2.3 理解Context和使用Global.asax 文件	518
15.2.4 处理应用程序Start和Init事件	519
15.2.5 处理Application_BeginRequest 事件	521
15.3 使用Web.Config文件	524
15.3.1 研究配置部分	525
15.3.2 修改配置设置	526
15.3.3 设置配置位置	527
15.3.4 锁定配置设置	528
15.3.5 添加自定义的配置信息	528
15.4 使用HTTP处理器和模块	529
15.4.1 使用HTTP处理器	530
15.4.2 使用HTTP模块	533
15.4.3 创建WhosOn应用程序	535
15.5 小结	540
第16章 跟踪用户会话	541
16.1 使用浏览器cookie	541
16.1.1 cookie如何工作	542
16.1.2 创建和读取会话cookie	542
16.1.3 创建和读取持久性cookie	544
16.1.4 设置cookie属性	545
16.1.5 cookie的限制	546
16.2 使用会话状态	547
16.2.1 向会话状态中添加条目	547
16.2.2 从会话状态中删除条目	548

16.2.3 启动用户会话	548	18.2 跟踪和监视应用程序	597
16.2.4 结束用户会话	548	18.2.1 跟踪页面的执行	597
16.2.5 处理会话事件	549	18.2.2 监视ASP.NET进程	603
16.2.6 在进程内存存储会话状态	551	18.2.3 获取进程信息	605
16.2.7 在Windows服务中存储会话状态	551	18.3 记录事件	607
16.2.8 在数据库表中存储会话状态	552	18.4 使用调试器	615
16.2.9 禁止会话状态	553	18.4.1 连接调试器	615
16.3 使用无cookie会话	554	18.4.2 建立断点	615
16.3.1 启用无cookie会话	554	18.4.3 建立观察	616
16.3.2 无cookie会话的限制	555	18.4.4 逐步执行ASP.NET页面	616
16.4 小结	555	18.5 小结	617
第17章 缓存ASP.NET应用程序	556	第五部分 保护ASP.NET应用程序	
17.1 使用页面输出缓存	556	第19章 使用基于表单的身份验证	619
17.1.1 按参数改变缓存内容	558	19.1 使用表单身份验证	619
17.1.2 按头改变缓存内容	560	19.1.1 启用表单身份验证	620
17.1.3 按自定义的字符串改变缓存内容	562	19.1.2 配置表单身份验证	623
17.1.4 设置缓存位置	564	19.1.3 配置表单授权	624
17.1.5 使用HttpCachePolicy类	564	19.1.4 获取用户信息	625
17.2 使用页面分段缓存	565	19.1.5 创建注销页面	627
17.2.1 按参数改变页面分段缓存	567	19.1.6 用Web.Config文件进行用户身份验证	628
17.2.2 页面分段缓存的限制	569	19.1.7 用XML文件进行用户身份验证	632
17.3 使用页面数据缓存	570	19.1.8 用数据库表进行用户身份验证	637
17.3.1 在缓存中添加条目	570	19.1.9 实现基于角色的身份验证	643
17.3.2 添加缓存文件依赖性	572	19.1.10 创建自定义的身份验证票据	645
17.3.3 添加缓存触发器依赖性	574	19.1.11 表单身份验证和Web阵	648
17.3.4 添加缓存键依赖性	576	19.2 使用Passport身份验证	649
17.3.5 建立绝对的过期策略	577	19.2.1 启用Passport身份验证	649
17.3.6 建立变化的过期策略	578	19.2.2 允许用户登录和注销	650
17.3.7 设置缓存条目优先级	579	19.2.3 获取用户信息	653
17.3.8 创建缓存回调方法	580	19.3 小结	656
17.4 小结	584	第20章 基于Windows的身份验证	657
第18章 应用程序跟踪和错误处理	585	20.1 使用基于Windows的身份验证	657
18.1 响应错误	585	20.1.1 配置Internet Information Server安全	657
18.1.1 查看错误信息	588	20.1.2 配置Microsoft Windows安全	658
18.1.2 页面级错误处理	590		
18.1.3 应用程序级错误处理	594		

20.1.3 配置Windows身份验证	659
20.1.4 配置Windows授权	659
20.1.5 配置自定义角色	661
20.1.6 获取用户信息	663
20.2 模拟用户账号	663
20.3 设置安全策略	665
20.4 小结	666
第21章 加密网上发送的数据	667
21.1 使用Secure Sockets Layer	667
21.1.1 加密	667
21.1.2 身份验证	668
21.1.3 数据完整性	669
21.1.4 SSL有多安全	669
21.2 将服务器配置为使用SSL	669
21.2.1 产生证书请求文件	669
21.2.2 申请服务器证书	670
21.2.3 安装服务器证书	671
21.3 在ASP.NET页面中使用SSL	671
21.4 使用.NET加密类	672
21.4.1 使用散列算法	673
21.4.2 使用对称加密算法	675
21.4.3 使用不对称加密	679
21.5 小结	685

第六部分 建立ASP.NET Web服务

第22章 创建XML Web服务	687
22.1 XML Web服务概述	687
22.1.1 XML Web服务促进通信	688
22.1.2 XML Web服务允许聚集	689
22.2 创建简单的XML Web服务	689
22.2.1 设置WebMethod属性	690
22.2.2 设置WebService属性	690
22.2.3 预编译XML Web服务	691
22.3 从浏览器测试XML Web服务	692
22.3.1 用HTTP-Get调用XML Web服务	692
22.3.2 用HTTP-Post调用XML Web服务	693
22.3.3 用SOAP调用XML Web服务	693

22.4 通过代理类访问XML Web服务	694
22.4.1 产生XML Web服务代理类	694
22.4.2 使用XML Web服务代理类	695
22.4.3 使用Web服务描述语言工具	696
22.4.4 设置代理类属性	697
22.5 在XML Web服务中传送复杂数据	698
22.5.1 XML Web服务和数组	698
22.5.2 XML Web服务和类	700
22.5.3 Web服务和DataSet	703
22.5.4 Web服务和二进制文件	704
22.6 XML Web服务和网站的交互	706
22.6.1 XML Web服务和应用程序状态	706
22.6.2 XML Web服务和会话状态	708
22.7 小结	711
第23章 高级的XML Web服务	712
23.1 使用WebService行为	712
23.1.1 WebService行为的限制	712
23.1.2 用WebService行为创建简单的 页面	713
23.1.3 使用WebService行为回调函数	716
23.1.4 缓存WebService行为中的错误	717
23.1.5 使用WebService行为执行部分更新	718
23.1.6 使用WebService行为获取数据库 数据	721
23.2 保护XML Web服务	724
23.2.1 安全的XML Web服务	725
23.2.2 创建数据库表	725
23.2.3 创建Login()方法	726
23.2.4 获取自定义的SOAP头	729
23.2.5 验证会话键	729
23.2.6 缓存会话键	730
23.2.7 建立安全的XML Web服务	731
23.2.8 访问安全的Web服务	734
23.3 使用HTML模式匹配	735
23.3.1 创建WSDL文档	736
23.3.2 指定正则表达式模式	738
23.3.3 创建简单的HTML模式匹配服务	738

23.3.4 在HTML模式匹配中使用输入参数	741	第26章 发送电子邮件和访问网络	808
23.3.5 建立一个Six Degrees Web服务	743	26.1 从ASP.NET页面发送电子邮件	808
23.4 小结	747	26.1.1 Microsoft SMTP Service	808
第七部分 利用.NET框架		26.1.2 发送简单的电子邮件	810
第24章 操作集合和字符串	749	26.1.3 使用MailMessage类	812
24.1 使用集合	749	26.1.4 向电子邮件消息添加附件	813
24.1.1 使用ArrayList集合	749	26.1.5 发送HTML电子邮件	814
24.1.2 使用HashTable集合	756	26.2 使用Message Queuing	817
24.1.3 使用SortedList集合	758	26.2.1 配置Microsoft Message Queuing	818
24.2 使用字符串	761	26.2.2 使用消息队列	818
24.2.1 对字符串进行格式化	761	26.2.3 向队列发送消息	820
24.2.2 使用String方法和属性	768	26.2.4 从队列获取消息	823
24.2.3 使用StringBuilder类	772	26.2.5 显示消息体	825
24.3 使用正则表达式	775	26.2.6 发送和接收复杂的消息	827
24.3.1 使用正则表达式类	776	26.3 使用HTTP协议访问其他网站	830
24.3.2 研究正则表达式的元素	779	26.3.1 使用WebClient类	831
24.3.3 匹配单一字符	779	26.3.2 使用HttpWebRequest类	832
24.3.4 匹配特殊字符	780	26.3.3 解析域名	834
24.3.5 匹配字符的替代序列	781	26.4 小结	835
24.3.6 使用数量词进行匹配	781	第27章 用GDI+动态创建图形	836
24.3.7 正则表达式的超范围问题	782	27.1 建立简单的图像	836
24.3.8 捕获和反向引用	783	27.2 使用GDI+绘制对象	838
24.3.9 使用替换模式	784	27.2.1 创建位图	838
24.3.10 设置正则表达式选项	785	27.2.2 使用Graphics对象	839
24.4 小结	786	27.2.3 设置图形质量	840
第25章 操作文件系统	787	27.2.4 使用颜色	840
25.1 使用文件和目录	787	27.2.5 使用画刷	843
25.1.1 创建和读取文本文件	787	27.2.6 使用画笔	848
25.1.2 创建和读取二进制文件	790	27.2.7 绘制矩形	855
25.1.3 显示目录的内容	792	27.2.8 绘制直线	855
25.1.4 获取路径信息	793	27.2.9 绘制椭圆	855
25.2 使用串行化	794	27.2.10 绘制曲线	855
25.2.1 使用二进制串行化	795	27.3 绘制文本字符串	857
25.2.2 使用XML串行化	801	27.4 建立GDI+应用程序	859
25.3 小结	807	27.4.1 创建ASP.NET绘图应用程序	859
		27.4.2 创建简单的图表应用程序	863
		27.5 小结	867

第八部分 建立自定义的ASP.NET控件

第28章 开发自定义的控件	869
28.1 创建简单的控件	869
28.2 使用HtmlTextWriter	871
28.3 向控件添加属性和方法	872
28.3.1 使用属性访问函数	874
28.3.2 使用控件方法	876
28.4 解析内部内容	877
28.5 向控件添加子控件	878
28.6 自定义控件和事件	884
28.7 参与postback	887
28.8 创建组合控件	895
28.9 派生现有的控件	900
28.10 访问当前上下文	901
28.11 调试控件	902
28.12 小结	904
第29章 高级控件开发	905
29.1 向控件添加模板	905
29.1.1 创建一个模板的多个实例	908
29.1.2 向一个控件添加多个模板	910
29.2 创建DataBound控件	913
29.2.1 实现不同的数据源	913
29.2.2 实现简单的DataBound控件	914
29.2.3 使用DataBound控件和模板	916
29.2.4 使用DataBound控件和状态	918
29.2.5 将自定义控件绑定到DataSet	922
29.3 使用自定义控件和Web服务	928
29.3.1 创建特色产品Web服务	928

29.3.2 创建特色产品自定义控件	929
29.3.3 显示特色产品控件	931
29.4 小结	932

第九部分 ASP.NET应用程序示例

第30章 创建职业介绍网站	933
30.1 安装职业介绍网站	934
30.2 创建主页	934
30.3 对用户进行身份验证	936
30.4 创建虚拟URL	942
30.5 列出和更新职位	943
30.6 创建职业介绍XML Web服务	944
30.7 小结	951
第31章 创建网上商店	952
31.1 ASP.NET Unleashed示例商店概述	952
31.2 安装ASP.NET Unleashed示例商店	954
31.3 建立导航系统	954
31.4 缓存产品数据	956
31.5 建立购物车	958
31.6 动态装载产品模板	964
31.6.1 使用LoadControl方法	965
31.6.2 code-behind产品模板	967
31.7 小结	968

第十部分 附 录

附录A 从ASP迁移到ASP.NET	969
附录B HTML控件参考	977
附录C Web控件参考	986

第一部分 使用ASP.NET Web表单

第1章 建立ASP.NET页面

本章主要内容：

- ASP.NET和.NET框架
- ASP.NET控件简介
- 向ASP.NET页面中添加应用逻辑
- ASP.NET页面的结构

在本章中，你将学习如何建立ASP.NET Web表单页面。不要让“表单”这个词误导了你。Web表单页面能够做的事远不只是显示标准的HTML表单，可以用它创建具有交互式、动态或数据库驱动的内容的页面。

Web表单页面是由两个构件组成的。首先，使用ASP.NET控件组装用户界面的动态部分。例如，ASP.NET控件使你能够显示“智能”HTML表单以及提供数据库数据的交互式网格。本章的第一部分概述所有ASP.NET控件。

Web表单页面的第二个构件是应用逻辑，这包括点击表单按钮时执行的代码，或者获取在控件中显示的数据库数据的代码。在本章的第二部分中，你将学习如何在Web表单页面中添加应用逻辑以便处理控件和页面事件。

最后，在本章的最后一部分中，向你介绍ASP.NET Web表单页面的结构。你将了解Web表单页面的各个部分以及适用于每个部分的内容类型。

在学习本章之后，你就可以开始使用ASP.NET控件并且用应用逻辑处理控件和页面事件来建立交互式动态页面，最终创建动态Web站点。

1.1 ASP.NET和.NET框架

ASP.NET是Microsoft的.NET框架的一部分，它包含大量编程类，可以满足任何想得到的编程需要。在下面两节中，你将学习ASP.NET如何与.NET框架配合，以及可以在ASP.NET页面中使用的语言。

1.1.1 .NET框架类库

假设你是Microsoft的决策人，再假设你必须支持多种编程语言，比如Visual Basic、JScript和C++。这些编程语言的功能多有重复。例如，每种语言都有用于访问文件系统、操作数据库和操作字符串的方法。

而且，这些语言包含类似的编程结构。例如，每种语言都可以表达循环和条件。尽管用

Visual Basic写的条件语法不同于C++写的条件语法，但是编程原理是一样的。

最后，大多数编程语言具有类似的变量数据类型。例如，在大多数语言中，都有办法表达字符串和整数。整数的最大和最小尺寸取决于语言，但是基本的数据类型是一样的。

为多种语言维护这些功能要做大量工作。为什么要在基本的编程元素上重复投资呢？如果能够只创建这些功能一次并且在所有语言中使用，那么不是简单得多吗？

.NET框架类库就能达到这个目的。它包含大量类，可以满足任何想得到的编程需要。例如，.NET框架包含用于处理数据库访问、操作文件系统、操作文本和产生图形的类。另外，它还包含更多特殊化的类，用于执行处理正则表达式和处理网络协议等任务。

另外，.NET框架包含代表所有基本变量数据类型的类，比如字符串、整数、字节、字符和数组。

对于本书来说最重要的是，.NET框架类库包含用于建立ASP.NET页面的类。总之，你要知道在建立ASP.NET页面时可以访问任何.NET框架类。

1.1.2 理解名称空间

你可以想象得到，.NET框架非常大。它包含数千个类（超过3400个类）。好在这些类并不是堆在一起的。.NET框架的类由名称空间组成层次结构。

ASP传统提示 在以前的Active Server Pages版本中，只能访问五个标准类（Response、Request、Session、Application和Server对象）。而在ASP.NET中可以访问超过3400个类。

名称空间（namespace）是类的逻辑分组。例如，与操作文件系统相关的所有类集中在System.IO名称空间中。

名称空间组织成一个层次结构（逻辑树）。这个树的根是System名称空间。这个名称空间包含用于基本数据类型（比如字符串和数组）的所有类。它还包含用于操作随机数、日期和时间的类。

类的完整的名称空间可以惟一地标识.NET框架中的任何类。例如，以下代码惟一地代表一个文件系统文件的类（File类）：

```
System.IO.File
```

System.IO代表名称空间，而File代表具体的类。

注意 查看.NET框架的参考文档，可以了解.NET框架类库中标准类的所有名称空间。

1.1.3 标准的ASP.NET名称空间

ASP.NET页面可以默认地使用某些名称空间中包含的类（对于其他名称空间，则必须显式地导入）。这些默认的名称空间包含ASP.NET应用程序中最常使用的类，如下所示：

System——包含所有基本数据类型和其他有用的类，比如与产生随机数和操作日期和时间相关的类。

System.Collections——包含用于操作标准集合类型（比如散列表和数组列表）的类。

System.Collections.Specialized——包含代表特殊化集合（比如链表和字符串集合）的类。

System.Configuration——包含用于操作配置文件（Web.config文件）的类。

System.Text——包含用于编码、解码和操作字符串内容的类。