



Beginning ASP.NET 2.0 Databases

ASP.NET 2.0 数据库入门经典 (第4版)

(美) John Kauffman 著
Thiru Thangarathinam 译
肖奕



清华大学出版社

ASP.NET 2.0

数据库入门经典

(第 4 版)

John Kauffman
(美) Thiru Thangarathinam 著
肖 奕 译

清华大学出版社

北 京

John Kauffman, Thiru Thangarathinam

Beginning ASP.NET 2.0 Databases

EISBN: 0-7645-7081-1

Copyright© 2005 by Wiley Publishing, Inc.

All Rights Reserved. Authorized translation from the English language edition published by Wiley Publishing, Inc.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2005-3306

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

ASP.NET 2.0 数据库入门经典(第4版)/(美)考夫曼(Kauffman, J.), (美)唐格勒西纳姆(Thangarathinam, T.)著;

肖奕译. —北京: 清华大学出版社, 2006.6

书名原文: Beginning ASP.NET 2.0 Databases

ISBN 7-302-12549-X

I. A… II. ①考… ②唐… ③肖… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2006)第 009822 号

出版者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社总机: 010-62770175 客户服务: 010-62776969

组稿编辑: 曹 康

文稿编辑: 于 平

封面设计: 孔祥丰

版式设计: 孔祥丰

印装者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 25.75 字数: 534 千字

版 次: 2006 年 6 月第 1 版 2006 年 6 月第 1 次印刷

书 号: ISBN 7-302-12549-X/TP·8028

印 数: 1~4000

定 价: 48.00 元

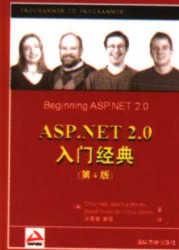
本书书评

本书是一本简单明了、通俗易懂的编程教程，对话式的叙述形式可帮助初级开发人员轻松掌握 ASP 数据库开发的基础内容；同时本书还演示了很多现实场景和实用技巧，可帮助您快速提高数据库开发技能。

—— Bradley Millington，微软公司，
Web 平台和工具小组的程序经理

译者简介

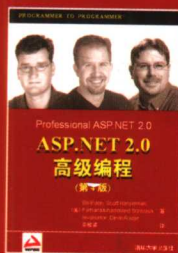
肖奕，男，计算机专业硕士，系统分析师，一直从事系统分析和程序开发工作，为企业信息化提供咨询和服务。参与翻译了四本著作。



《ASP.NET 2.0 入门经典
(第4版)》

(美)Chris Hart, John
Kauffman 等著
张楚雄 高猛 译

ASP.NET 2.0 构建在 .NET Framework 2.0 之上,它扩展了 ASP 1.0 和 ASP 1.1 的功能。本书全面介绍了 ASP.NET 2.0 的新特性和各方面的编程知识,是 ASP.NET 初学者以及任何想了解 ASP.NET 2.0 新技术的读者的理想教程。该书从初版、1.0 版、1.1 版到现在的 2.0 版,内容更加完善,技术更加实用。



《ASP.NET 2.0 高级编程
(第4版)》

(美)Bill Evjen, Scott
Hanselman 等著
李敏波 译

本书在全面介绍 ASP.NET 各种编程技能的同时重点介绍了 ASP.NET 2.0 版本中的巨大变化,详细阐述了 ASP.NET 2.0 中所包含的每个新特性。书中提供了大量的实例,可帮助读者快速掌握如何在 .NET Framework 下构建功能强大的 ASP.NET 应用程序。本书适用于 ASP.NET 初级程序员以及准备迁移到 ASP.NET 2.0 的编程老手。



《C# 入门经典(第3版)》

(美)Karli Watson, Christian
Nagel 等著
齐立波 译

这是一本曾成就无数 C# 程序员的经典名著,厚而不“重”,可帮助您轻松掌握 C# 2005 的各种编程知识,为您的职业生涯打下坚实的基础!本书自第 1 版出版以来,全球销量已达数万册,在中国也有近 3 万册的销量,已经成为广大 C# 初级程序员首选的入门教程,也是目前国内市场上最畅销的 C# 店销书之一。

前 言

ASP.NET 2.0 小组完成了开发 2.0 版本的艰巨工作，其意义在于以下两个方面。首先，它使得在 ASP 中使用数据比以往任何一个版本都要容易。其次，比以往任何时候，网站设计人员都可以执行更复杂的、与数据相关的任务。ASP.NET 2.0 的两个成功之处就是更快速的开发及其扩展能力，这也使得本书值得您一读。

本书读者对象

本书的读者对象与以前的版本保持一致。假如读者对使用无数据的 ASP 有基本的了解，那么可以从每一章和每一个练习获得详尽的解释。阅读本书的一般前提条件是读者应该阅读过由本书作者编写的 *Beginning ASP.NET 2.0* 一书。或者，应该熟悉 ASP.NET 1.x 版或者 ASP 3.0 版。

读者还应该对数据库有所了解。本书没有深入研究设计和管理数据库的细节。

如何使用本书

本书的讲解遵循由浅入深的原则。一般来说，我推荐您从头开始阅读本书，然后按照您自己的方式读完本书，并且认真思考每一道习题。但是，有几个例外。

第 2~4 章讲述了 3 种数据源的连接。如果您只使用 Microsoft SQL Server(或者各类 SSE 或 MSDE)或者 Oracle，那么第 3 章的内容会给您提供相关技术。您可以跳过第 2 章，略读第 4 章。

源代码

当您处理本书示例时，可以选择手动输入所有的代码，也可以使用与本书配套的源代码。本书中使用的所有源代码都可以从 <http://www.wrox.com> 或 <http://www.tupwk.com.cn/downpage> 上下载。在这个站点上，只要找到本书的书名(通过使用 Search 框或者使用书名)，并且在本书详细页面上单击 Download Code 链接即可获得本书的所有源代码。

因为很多书都有相似的书名，所以通过搜索 ISBN 可以很容易地找到本书，本书的 ISBN 是 0-7645-7081-1。

一旦下载了代码，只要用常用的压缩工具将其解压缩即可。或者，可以到 Wrox 专门的代码下载页面 <http://www.wrox.com/dynamic/books/download.aspx> 上查看本书所用的代码以及其他 Wrox 书籍的代码。

勘误表

我们尽全力保证文本或代码中没有一个错误。但是，凡事不可能完美，错误在所难免。如果您在我们的书中发现了错误，比如拼写错误或者代码错误，我们将非常欢迎您的反馈。通过发送勘误表，您也许可以节省其他读者犯错误的时间，并且可以帮助我们提供更加高质量的信息。

为了找到本书的勘误表页面，请浏览 <http://www.wrox.com> 并使用 Search 框或者书名列表找到本书的书名。然后，在本书的详细页面上单击 Book Errata 链接。在这个页面上，可以查看所有在本书中出现的错误列表。也可以从 www.wrox.com/misc-pages/booklist.shtml 上获得一个完整的书籍列表，包括至每本书的勘误表的链接。

如果您没有在 Book Errata 页面上发现“您的”错误，请浏览 www.wrox.com/contact/techsupport.shtml 并填写其中的表格将您发现的错误发送给我们。我们将检查收到的信息，如果属实，将添加消息至本书的勘误表页面中并在本书后续版本中解决这个问题。

p2p.wrox.com

如果要与作者和同行进行讨论，可以加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是基于 Web 的系统，您可以提交有关 Wrox 书籍和相关技术的消息并与其他读者和技术用户进行交流。论坛还提供订阅服务，当论坛中发表新文章时，论坛将把您选择的感兴趣的主题通过电子邮件发送给您。Wrox 作者、编辑、其他业界专家以及您的读者同伴都将会聚集在这些论坛中。

在 <http://p2p.wrox.com> 上，您将发现有些论坛不仅可以对您阅读本书给予帮助，还可以有助于您开发应用程序。如果要加入论坛，请执行如下步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接，阅读使用条款，并单击 Agree。
- (2) 完成加入所需的信息和任何您想要论坛提供的信息，然后单击 Submit。
- (3) 您将收到一封电子邮件，介绍如何验证您的账户并完成加入过程。

可以阅读论坛中的信息而不加入 P2P，但是只有加入论坛才能发表文章。

一旦您加入论坛,就可以发表自己的新文章并且可以回复其他用户发表的文章。可以在任何时候阅读该 Web 站点上的信息。如果您想将一个特定论坛上的新文章通过电子邮件发送给您,可以通过论坛列表中的论坛名称并单击 Forum 图标中的 **Subscribe**。

如果需要更多有关如何使用 Wrox P2P 的信息,一定要阅读 P2P FAQ 中有关论坛软件如何工作的问题的回答,以及对很多有关 P2P 和 Wrox 书籍常见问题的回答。如果要阅读 FAQ,可以在任何 P2P 页面上单击 FAQ 链接。

目 录

第 1 章 ASP.NET 2.0 和 ADO.NET

简介 1

1.1 技术概述 1

1.1.1 .NET Framework 2

1.1.2 ASP.NET 3

1.1.3 ASP.NET 2.0 4

1.1.4 ADO.NET 6

1.1.5 ASP.NET 2.0 和数据访问 8

1.1.6 术语 9

1.2 建立本书所需的环境 11

1.2.1 安装 Web 服务器 12

1.2.2 安装 .NET Framework 2.0 版本 12

1.2.3 安装创建 Web 页的编辑器 13

1.2.4 安装数据库管理系统 14

1.3 安装 16

1.4 演示 19

1.5 本章小结 27

1.6 练习题 28

第 2 章 连接 Access 数据库 29

2.1 Microsoft Access 和 JET Database Engine 29

2.2 在 Visual Web Developer 中连接到 Microsoft Access 数据库 30

2.3 AccessDataSource 控件 33

2.3.1 选择语句中的变量 36

2.3.2 MDB 文件位置中的变量 40

2.4 处理文件访问许可 41

2.5 处理 Access 连接失败 42

2.6 在 Web 应用程序中使用 Access 的一些提示 45

2.7 使用 AccessDataSource 控件的 常见错误 46

2.8 本章小结 47

2.9 练习题 48

第 3 章 连接 SQL Server 和 SQL Server Express 49

3.1 SQL Server 和连接字符串 49

3.1.1 使用 SQL Server 数据库之前 的准备 51

3.1.2 连接字符串 51

3.2 使用 SqlDataSource 控件 52

3.3 了解 SQL Server 中的安全性 57

3.4 在 Web.config 文件中保存 连接字符串 59

3.5 数据库的结构 64

3.6 处理 SqlDataSource 的连接失败 67

3.7 常见错误 70

3.8 本章小结 70

3.9 练习题 71

第 4 章 连接其他关系型数据库 72

4.1 带提供程序的连接 72

4.1.1 连接软件的层次关系 73

4.1.2 可支持的 ADO.NET 提供程序 74

4.1.3 在 ASP.NET 2.0 中使用 提供程序 75

4.2 当使用密码时连接 Access 77

4.3 连接 Oracle 82

4.4 连接 MySQL 83

4.4.1 连接 MySQL 的理论 83

4.4.2 用于 MySQL 的 ODBC 连接的语法 83

4.5 连接其他数据库 85

4.6 常见错误 86

4.7 本章小结 87

4.8 练习题 87

第 5 章 在表中显示数据 88

5.1 在 ASP.NET 2.0 中显示数据 88

5.1.1 回顾数据绑定和数据源控件 89

5.1.2 数据绑定控件的类型 89

5.2	GridView 控件	90	6.9	练习题	145
5.2.1	GridView 控件的性能	90	第 7 章	数据的分页和排序	146
5.2.2	GridView 的呈现元素	90	7.1	排序	146
5.3	将 GridView 与数据连接	91	7.1.1	排序所需条件	148
5.3.1	从 Data Explorer 中拖放字段	92	7.1.2	排序表达式	151
5.3.2	从工具箱中拖放控件	92	7.2	分页	155
5.4	定制 GridView 的列	94	7.2.1	启用分页	155
5.4.1	在 Edit Columns 对话框中 选择列	94	7.2.2	定制分页和分页栏	157
5.4.2	列字段类型	94	7.3	分页理论和替代方法	162
5.4.3	GridView 的 AutoGenerate Columns 属性	110	7.4	排序、分页和选择三者的关系	162
5.4.4	处理 NULL 字段值	111	7.5	常见错误	164
5.5	DetailsView 控件	112	7.6	本章小结	164
5.5.1	DetailsView 的呈现元素	113	7.7	练习题	164
5.5.2	将 DetailsView 与数据连接	113	第 8 章	在列表中显示数据	166
5.6	常见错误	115	8.1	简介	166
5.7	本章小结	116	8.1.1	从 GridView 转换成其他格式	167
第 6 章	定制表的外观	118	8.1.2	列表控件的类型	167
6.1	定制外观	118	8.2	列表控件的一般概念	167
6.1.1	BackColor 和 BackImageUrl	119	8.3	数据绑定列表控件	169
6.1.2	Font 和 ForeColor	120	8.4	DropDownList 控件	172
6.1.3	Height 和 Width	120	8.5	处理列表中的选择	174
6.1.4	CellSpacing 和 CellPadding	121	8.5.1	SelectedIndex 和 SelectedValue	174
6.1.5	Borders 和 GridLines	121	8.5.2	自动回送	176
6.1.6	HorizontalAlign	123	8.6	常见错误	178
6.1.7	ShowHeader 和 ShowFooter	123	8.7	本章小结	179
6.1.8	ToolTip	123	第 9 章	筛选和主从数据	180
6.2	定制独特的样式	125	9.1	使用 QueryString 筛选 GridView 记录	181
6.2.1	GridView 和 DetailsView 样式	125	9.2	使用 TextBox 筛选 GridView 记录	184
6.2.2	空表	128	9.3	使用 SQL 的 LIKE 操作符	185
6.2.3	DetailsView 特有样式	129	9.4	选择的理论和 ControlParameters	186
6.2.4	列样式和字段样式	131	9.5	使用列表控件筛选 GridView 记录	188
6.3	使用层叠样式表	133	9.5.1	由具有硬编码项的 DropDownList 作为主控件	189
6.4	样式的优先级	137	9.5.2	由具有数据绑定项的列表框 作为主控件	190
6.5	主题和皮肤	141			
6.6	在 Visual Studio 中使用 Auto Format	142			
6.7	常见错误	143			
6.8	本章小结	144			

9.5.3 由具有初始隐藏在 GridView 内的默认设置的列表框作为主控件	192	12.3 从 GridView 开始执行 DetailsView 的 INSERT	258
9.6 使用 GridView 和 DetailsView 在同一页面上显示详细信息	196	12.3.1 在同一个页面上执行插入的 GridView 和 DetailsView	258
9.7 使用 GridView 和 DetailsView 在不同页面上显示详细信息	200	12.3.2 GridView 和 DetailsView 分别位于不同页面	258
9.8 层叠 DropDownLists	205	12.4 使用模板字段进行插入	259
9.9 常见错误	206	12.5 RadioButtonLists 和 DropDownLists 中的数据条目	261
9.10 本章小结	207	12.6 复选框中的数据条目	265
9.11 练习题	208	12.7 使用 FormView 进行插入	265
第 10 章 在模板化控件中显示数据	209	12.8 常见错误	269
10.1 GridView 和 DetailsView 模板字段	215	12.9 本章小结	270
10.2 DataList 控件	218	第 13 章 作为数据源的业务对象	272
10.3 Repeater 控件	223	13.1 简介	272
10.4 DetailsView 控件	224	13.1.1 一个 N 层应用程序的层次	273
10.5 模板化控件中的导航	224	13.1.2 N 层体系结构的优势	273
10.6 FormView 控件	225	13.2 使用 VS DataSet 建立数据访问逻辑层	274
10.7 模板化控件的比较和选择	228	13.2.1 使用 VS DataSet 设计器创建数据访问逻辑层	274
10.8 常见错误	229	13.2.2 与 DataSet 进行数据绑定	277
10.9 本章小结	230	13.3 实现主从功能	279
10.10 练习题	231	13.4 实现并数据绑定于自定义对象	283
第 11 章 数据的更新与删除	232	13.5 在自定义对象中实现排序	290
11.1 数据修改概述	232	13.6 常见错误	295
11.2 命令字段	234	13.7 本章小结	295
11.3 简单更新	236	第 14 章 XML 和其他分层数据	297
11.4 DataKeyNames 和更新	239	14.1 什么是分层数据	297
11.5 DetailsView 中的更新	241	14.1.1 分层数据的类型	298
11.6 参数集合	243	14.1.2 ASP.NET 2.0 的分层数据控件	298
11.7 处理更新中的 NULL 值	244	14.2 XmlDataSource 控件	299
11.8 删除整条记录	246	14.3 用来显示分层数据的 TreeView 控件	299
11.9 常见错误	248	14.4 显示 XML 数据	301
11.10 本章小结	248	14.5 使用 XmlDataSource 的 XPath	302
11.11 练习题	249	14.6 格式化 TreeView	304
第 12 章 插入新记录	250	14.7 将 XSL 转换用于 XML	307
12.1 使用 DetailsView 进行基本插入	253		
12.2 在数据源控件中启用插入	255		

14.8	处理 TreeView 控件中的事件	310
14.9	除 TreeView 之外的其他 控件的分层数据	313
14.9.1	XML 和 GridView	313
14.9.2	用于 DataList 的 XML 源	315
14.9.3	使用 XPath 在 DataList 控件中显示数据	317
14.9.4	使用嵌套的 XML 和 DataList	319
14.10	使用 XmlDataSource 控件的参见错误	321
14.11	本章小结	322
第 15 章	数据缓存	324
15.1	缓存及其优点	324
15.2	何时使用缓存	325
15.2.1	理解状态	326
15.2.2	理解状态的失效	326
15.3	ASP.NET 2.0 中的缓存选项	326
15.3.1	显示 ASP.NET 中的数据	327
15.3.2	基于时间的缓存	329
15.3.3	具有参数的缓存	332
15.3.4	实现筛选功能	334
15.3.5	SQL Cache Invalidation	337
15.3.6	局部页面缓存	342
15.4	常见错误	345
15.5	本章小结	346

第 16 章	处理数据控件的事件	347
16.1	在事件触发时执行控件	348
16.2	事件类型	348
16.3	编写事件处理程序的常用技术	348
16.3.1	触发事件时值的传递	353
16.3.2	事件处理程序的位置	354
16.4	命令和自定义按钮事件	359
16.4.1	使用由命令按钮(命令字段) 触发的事件	360
16.4.2	使用由按钮通过自定义 操作触发的事件	360
16.5	列表选择和页面事件	362
16.6	数据控件绑定事件	365
16.7	一般错误事件	368
16.8	常见错误	371
16.9	本章小结	372
16.10	练习题	373

附录 A	SQL 语句的简单实用的介绍	374
A.1	语法综述	380
A.2	获取数据列	381
A.3	限制记录集	382
A.4	记录的次序	383
A.5	Distinct 和 Top	383
A.6	更改数据	384
A.7	连接(Join)	385
A.8	SQL 注入	385

附录 B	练习答案	386
-------------	-------------	------------

第 1 章 ASP.NET 2.0 和 ADO.NET

简介

考虑到 ASP.NET 2.0 和 .NET Framework 2.0 给初级程序人员带来的快捷和强大功能，我非常高兴能够编写这一版本。我相信您将会经常惊诧于自己花费很少的精力却能得到高质量的结果。我想，就像我编写第一个示例时一样，您也会经常发出这样的感叹，“喔，事情就应该是这样”。幸运的是，您的经理和客户们都将对您创建的访问快速且内容精确的 Web 站点留下深刻的印象。

本章将会介绍如何在 ASP.NET 2.0 Web 页上使用数据，如何使用本书提供的软件和数据来构建一台能够使用 ASP.NET 2.0 的机器，另外还演示了您将学会使用的一些强大功能。有些材料将会回顾一些预备知识(ASP.NET 的基本知识并熟悉基本的数据库操作)。以下是章节内容的安排：

- 概述 .NET Framework、ASP.NET 和 ADO.NET
- 介绍 ASP.NET 2.0 以及 ASP.NET 2.0 Web 页如何使用数据
- 指导本书所需的安装任务
- 演示几个应用了数据的 ASP.NET 2.0 页面

在本章的中间部分，您将会发现一个软件安装的步骤列表，该软件用来完成本书的所有练习，包含 .NET Framework 2.0(包括 ASP.NET)、一个用来创建页面的编辑器、数据库软件以及一些样本数据库。

在最后一节中，您将会看到以下 3 个演示：

试 一 试	主 题
1	用 GridView 表从 XML 中获取数据，并对数据进行分页和排序
2	用 DataList 从 Access 中获取数据
3	用 Tree View 从 XML 中获取数据

1.1 技术概述

大约有将近一百万名开发人员使用了 .NET Framework 的第一个版本来开发网站。所以在 2003 年的夏天，当从 Microsoft 传出将有新版本发布的传闻时，许多人的耳朵都竖了起来，这个新的版本将使得创建 ASP.NET 页面所需的代码行数减少 70%。像这样大规模的提高生产效率在程序设计中是不多见的。当 ASP.NET 2.0 的代码样例在 2003 年秋的 Microsoft 专业开发人员大会(Microsoft Professional

Developer's Conference)上演示时,结果比预计的还要好。以前使用 ASP.NET 的第一个版本程序员需要花费几个小时才能做出的页面,现在使用 ASP.NET 2.0 只需要几分钟就完成了。简单来说,在 .NET Framework 2.0 最终版本发布之后,任何继续使用第一个版本创建 ASP.NET 页面的程序人员都将花费大量的额外时间才能达到相同的结果。

也许就像其他任何领域一样,ASP.NET 2.0 所提供的优势是能够方便地将数据集成进页面。程序员不再需要知道详细的连接、命令以及数据阅读器和数据适配器对象,就能执行普通数据任务了。对于初学者掌握能力,ASP.NET 2.0 使得基本数据的使用简单易学,并且有能力进行更复杂的数据应用。

1.1.1 .NET Framework

Microsoft 开发出了作为基本原理的 .NET 和一系列技术,用于在互联网中让计算机共同工作。总的目标就是让信息和进程在大范围的系统和设备之间顺畅地交流。.NET 不是一种语言,也不是一个特别的产品。更确切地说,它是一套标准和规范,并已经应用于自 2002 年以来 Microsoft 发布的所有产品中。

.NET 包含了一种使用开放标准的 XML 格式交换信息的标准化格式。可扩展标记语言(Extensible Markup Language, XML)不需要请求者具备任何有关数据存储如何保存信息的专门知识——数据都以自描述的 XML 格式取出。同样地,目前几乎所有的数据存储都可以用 XML 来提供信息,这对于所有 .NET 数据客户都具有吸引力。

.NET 支持软件的 Web Services 标准,可请求在使用了开放平台标准的简单对象访问协议(Simple Object Access Protocol, SOAP)和 XML 的远程软件上运行代码。.NET 网站可以从另外一个网站上找到该网站所提供的服务,并使用这些服务。这样可以使得网站从其他的网站上获得 HTML、计算后的结果或者数据集。

作为 .NET 开端的一部分,Microsoft 发布了一套运行时编程工具和应用编程接口(API),称为 .NET Framework,让开发团队能够创建 .NET 应用程序和 XML Web Services。.NET Framework 由公共语言运行库(Common Language Runtime, CLR)和一套统一的类库组成。

CLR 为运行的应用程序提供了一个完全管理的执行环境,其中包括几个服务,例如程序集装载和卸载、进程和内存的管理、安全实施以及即时编译等。CLR 名称的意思就是指能够用多种语言编写应用程序,并且将源代码编译成 CLR 能够读懂并运行的中间语言,而无需考虑原来所使用的语言。这种“语言独立性”就是 CLR 的关键特性(也是 ASP.NET 的特性),它允许开发人员使用自己喜欢的语言工作,比如 C#、VB 或者 Cobol,都能够获得 .NET Framework 的常用特性。

.NET Framework 还包括了一套类库,这套类库提供了每一个应用程序所需的常用功能。可以使用 .NET Framework 支持的任何语言来访问这些类库。由这些类库提供的服务(以及相应的命名空间)如下:

- 基本类型(System)
- 输入/输出(System.IO)
- 数据访问(System.Data)
- 安全(System.Security)
- 数据结构(System.Collections)
- 配置(System.Configuration)
- 连网(System.Net)
- 反射(System.Reflection)
- 全球化(System.Globalization)
- 绘画和制图(System.Drawing)
- 跟踪和诊断(System.Diagnostics)
- 窗口(客户端)应用程序模型(System.Windows.Forms)
- Web 应用程序模型(System.Web)

注意, .NET Framework 包含了两个应用程序编程模型, 一个用于客户端应用程序(System.Windows.Forms), 另一个用于基于 Web 的应用程序(System.Web)。本书着重讲解后一个模型。 .NET Framework 中的 System.Web 命名空间是 .NET Framework 的一部分, 它提供了 ASP.NET 功能。换句话说, ASP.NET 就是构建应用程序的所有 .NET Framework 的一部分。

1.1.2 ASP.NET

ASP.NET 是一种用于创建基于 Web 的应用程序的编程模型。从本质上来说, 运行时和 .NET Framework 类库集可以用于创建动态 Web 页。它需要在 Web 服务器的环境中运行, 例如 Microsoft Internet Information Server(Microsoft 互联网信息服务器, IIS), 并且根据服务浏览器请求指示在服务器上执行程序。与直接由 Web 服务器提供的静态 HTML 不同的是, ASP.NET 页面实际上是在服务器上执行以后再产生结果的。页面的最后生成也许是由许多不同的指令和/或数据源构造的。

ASP.NET 页面以.aspx 扩展名存储。页面由程序员将文本、标记(例如 HTML)以及 ASP.NET 特定服务器标记和脚本组合在一起, 然后存储在 Web 服务器上。可以将存储后的 ASP.NET 页面看成是一套描述如何创建一个 HTML 页面的指令。当该页面被请求浏览时, 服务器端程序将会用纯标记来创建一个客户端浏览器可以读懂并能呈现(render)的页面。因为呈现后的输出是纯标记, 所以任何浏览器都能够读懂; 所有的动态过程都发生在 Web 服务器端。ASP.NET 特定服务器标记非常强大, 例如, 它可以对用户的动作作出反应, 连接至数据存储以及自动创建非常复杂的 HTML 结构。

正像前面提到的那样, ASP.NET 只是 .NET Framework 的一部分, 所以 ASP.NET 页面可以利用这个框架提供的所有服务, 包括连网、数据访问、安全以及更多其他

服务。因为 ASP.NET 可以使用所有这些服务，所以相比以前，能够创建更加丰富的 Web 应用程序。只需花少量的时间来构建所有应用程序所需的构建块，而将大多数时间用在应用程序独有的特殊逻辑上。

ASP.NET 还在 Web 编程中引入了一些独特的新技术，可以在典型的动态服务器页面(Active Server Pages, ASP)上极大地改善开发模式：

- **语言独立性**——因为 ASP.NET 是 .NET Framework 的一部分，所以可以使用您自己选择的语言来构建 ASP.NET 应用程序，例如 C#、VB 或 J#。而典型的 ASP 则仅限于 JScript 或者 VBScript 页面。
- **编译而不是解释**——与典型的 ASP 在每一次页面请求时都解释编程结构不同，ASP.NET 在服务器端动态地将页面编译成可以运行得非常快的本机编程指令。可以很明显地看到典型的 ASP 页面的性能与相同 ASP.NET 页面的性能之间相差的数量级别。
- **事件驱动编程模式**——在典型的 ASP 中，页面总是以自顶向下的线性方式执行，并且 HTML 标记常常与程序指令混合在一起。任何一个有一定 ASP 经验的人都知道这样会使得页面难以阅读，甚至更加难以维护。ASP.NET 引入了事件驱动模型，这个模型允许您将代码与标记内容分离，将代码并入处理专门任务的有意义的单元中，例如响应客户端的按钮单击动作。这个类似 VB 的事件模型极大地提高了页面的可读性和可维护性。
- **服务器控件**——典型的 ASP 需要动态地将 HTML 片断代码接合在一起呈现，这样做的结果就是在应用程序中一遍又一遍地编写相同的代码(您需要多少次才能从数据库查询中构建一张表格)。ASP.NET 带给 Web 编程的一个最大的好处就是能够将公共的呈现和行为封装成服务器控件(server control)，可以在应用程序中很方便地重复使用。就像 HTML 标记一样，服务器控件以声明的形式创建，但是表现为一个位于服务器端的可编程对象，它可以与代码进行交互并输出定制的动态 HTML 呈现。ASP.NET 包含了大约 80 多个服务器控件，这些控件封装了从标准表元素到复杂控件(如网格和菜单)的所有内容。
- **控件设计时间的改善(当使用 Visual Web Developer 时)**——开发人员通过使用设计时间界面可以减少花费在开发复杂页面上的时间，这些界面包括敏捷任务面板、标签级导航栏和可以设置控件属性的向导。

1.1.3 ASP.NET 2.0

ASP.NET 的第一个版本(1.0 和 1.1)在 2001 年至 2003 年间迅速风靡了 Microsoft 的开发阵营。程序人员很快便感觉到他们通过使用强大和灵活的 .NET Framework 可以大大减少编码时间，而且 CIO 们也看到当程序员花在解决客户代码的疑难问题上的时间减少时，他们就可以将更多的资源投入到更高级的 IT 结构的改善上。

ASP.NET 确实是一个里程碑式的版本，它简化了开发人员的工作。

但是，就在第一个版本发布之前，Microsoft 的 ASP.NET 小组就已经在为开发 ASP.NET 2.0 而工作了。他们以下面雄心勃勃的目标来开始他们的工作：

- 使创建一个典型的 Web 应用程序所需代码行数减少 70%。
- 提供一套可扩展的应用程序服务，用来为通用应用程序任务提供构建块，例如成员、角色、个人化以及导航等。
- 创建一系列基于任务的服务器控件，这些控件可以调节上述服务，交付完全、可定制的用户接口(UI)，以最小的代码量来展示这些服务。
- 当与 .NET Framework 协同工作提供页面服务时，改善 IIS 的性能。
- 提供管理功能，以便加强 ASP.NET 服务器的部署、管理和运行。
- 改善宿主公司所用的工具，以便可以支持多站点并能够将开发人员的项目迁移至公共部署环境。
- 让 ASP.NET 的几乎所有特性都能够方便地扩展或者用定制的高级任务的执行替换。

在这里，我们有必要再来细细地回想一下第一个目标，也就是使编写一个动态 Web 应用程序所需的代码量减少 70%。这有可能吗？Microsoft 的 ASP.NET 小组已经仔细考虑过各种以定制代码执行的通用任务，并且专门制定了将这些任务封装进构建块(特别是服务器控件)的方式，这样就能够自动地完成这些任务了。例如，大多数 Web 应用程序都需要安全或者导航或者个性化服务来为用户提供定制的体验。在 ASP.NET 2.0 中，这些任务是通过一系列可配置的应用程序服务，以及与这些应用程序服务进行对话的服务器控件来实现的，这样可以极大地减少实施这些通用服务所需的应用程序的代码量。但是，在所有这些通用任务当中，有一个任务是绝对独立于其他应用程序的，这就是数据访问服务。数据是驱动所有动态 Web 应用程序的公用线程，所以毫不奇怪，ASP.NET 小组为了减少代码量和在 ASP.NET 2.0 的应用程序中执行数据访问所需的概念，而制订了一些大胆的目标：

- 在 ASP.NET 中可以通过声明的(无代码)方式来定义一个数据源。
- 可以通过声明的(无代码)方式用 UI 控件显示数据，无需在页面的执行生存期中的特定时间进行显式的数据绑定。
- 可以通过声明的(无代码)方式执行通用数据任务，例如排序、分页、过滤、更新、插入以及删除数据。
- 可以使用多种 UI 控件来显示数据，包括灵活的网格/详细控件，该控件既可以显示又可以操作数据。
- 为创建定制的数据源，启用可扩展的模型来支持新的数据类型。

ASP.NET 2.0 具有一些程序员可以使用的特定的服务器控件，用于在页面上添加数据交互。这些专门的数据控件分为两组：数据源控件和数据绑定控件。数据源控件创建与数据库的链接。数据绑定控件则从数据源控件获取信息，并在页面上创