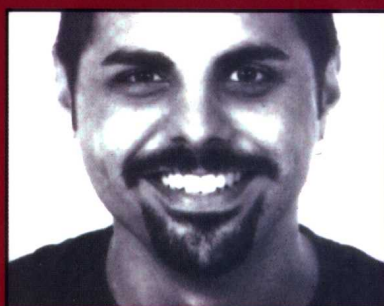
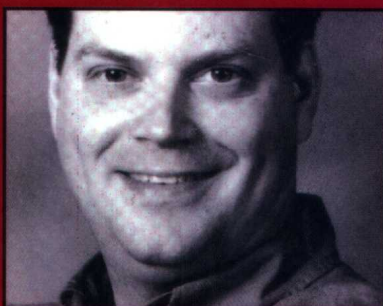
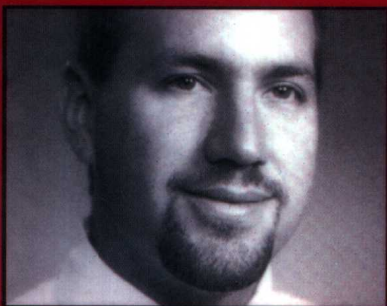
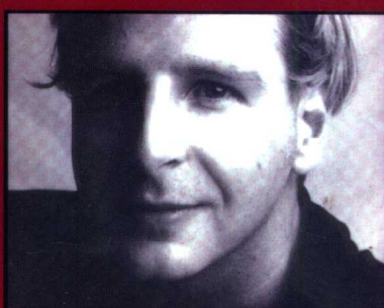
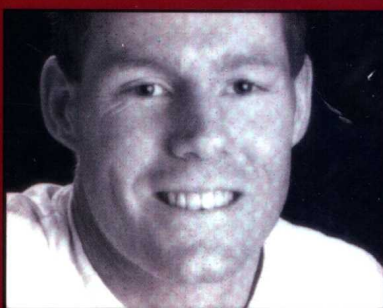




Professional ADO.NET Programming **ADO.NET** 高级编程

Paul Dickinson 等 著
张晓明 邓少鹏 译

**Professional
ADO.NET Programming**

ADO.NET
高级编程

Paul Dickinson 等 著
张晓明 邓少鹏 译



中国电力出版社

Professional ADO.NET Programming (ISBN 1-861005-27-X)

Paul Dickinson, Bipin Joshi, etc.

Authorized translation from the English language edition, entitled Professional ADO.NET Programming, published by Wrox Press, Copyright©2001

All rights reserved.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

CHINESE SIMPLIFIED language edition published by China Electric Power Press Copyright©2003

北京市版权局著作权合同登记号 图字：01-2002-0707 号

图书在版编目 (CIP) 数据

ADO.NET 高级编程 / (美) 狄金森等编著; 张晓明等 译. —北京: 中国电力出版社, 2002

ISBN 7-5083-1392-5

I .A... II.①狄... ②张... III.数据库—接口—程序设计 IV.TP311.11

中国版本图书馆 CIP 数据核字 (2002) 第 100311 号

责任编辑: 闫宏

书 名: ADO.NET高级编程

编 著: (美) Paul Dickinson 等

翻 译: 张晓明 邓少鹏

出版发行: 中国电力出版社

地址: 北京市三里河路6号 邮政编码: 100044

电话: (010) 88515918 传真: (010) 88518169

印 刷: 汇鑫印务有限公司印刷

开 本: 787×1092 1/16 印 张: 40.5 字 数: 922千字

版 次: 2003年8月北京第一版

印 次: 2003年8月第一次印刷

定 价: 69.00 元

引 言

什么是 ADO.NET

ADO.NET 是一组庞大的 .NET 类，它使我们能够以各种不同的方式检索和操作数据以及更新数据源。作为 .NET 框架的一个组成部分，它具有 .NET 框架的许多特性，例如多语言支持、垃圾回收、运行时编译、面向对象设计以及动态缓存等，并且比以前的 ADO 版本有很大的改进。

ADO.NET 是任何数据驱动的 .NET 应用程序或者 Web 服务的核心部件，理解其功能对任何希望利用 .NET 数据支持的人而言都非常重要。

本书内容

本书完整地剖析了 ADO.NET 类，这些类包含在 `System.Data`、`System.Data.Common`、`System.Data.SqlClient` 以及 `System.Data.Odbc` 命名空间中。在以数据为中心的应用程序开发过程中，我们采用实际的面向解决方案的方法有效地使用 ADO.NET 的各种组件。

我们在第 1 章简要介绍了数据访问的历史，而后更具体地介绍了 ADO.NET 的概念。我们考察了它的一些特性，并将其与 ADO 2.6 版本进行比较。然后，我们转入第 2 章来介绍 .NET 数据提供者，它提供了连接各种数据源的方法。

第 3 章深入地讨论了 Visual Studio .NET 以及如何通过图形用户界面来交互地使用和简单地处理 ADO.NET。本章包含了许多演示其原理的示例。

在大体介绍了 ADO.NET 之后，我们开始更深入地阐述这一技术。第 4 章讨论 `DataReader` 的如下内容：它是什么、为什么要使用它以及如何在各种情况下使用它。在第 5 章继续介绍 ADO.NET 技术，我们讨论了 `DataSet`。第 6 章则介绍了 `DataAdapter`。

第 7 章更详细地介绍 `DataSet`，主要侧重于在不连接数据源的情况下如何处理数据；其中包括对 XSD（XML Schema Define，XML 模式定义）语言在操纵 `DataSet` 时的优势的讨论。这样，我们很自然地引入了第 8 章，介绍如何通过 `DataSet` 来使用 XML，涵盖了各种使用问题，例如数据调度以及数据过滤等。

第 9 章继续讨论 `DataSet` 的约束、关系和视图检查，所有这些都会影响到数据提供和操作。本章也介绍了 `DataView` 和一些实例。

第 10 章转入讨论事务，它在商业环境中是一个很重要的概念。在商业环境中所有的操作要么都成功要么都失败。本章考察了隔离级别及其影响、性能和一些先进的事务处理技术。

映射的概念在第 11 章中讨论：我们可以使用自己喜欢的名字来标识列首部，这样有助于理解列的含义。

第 12 章阐述了如何创建用户自己的数据服务组件，其中包括这样做的好处、创建和部

署的方法以及使用已有的组件。本章也讨论了提高数据服务组件性能的一些技巧。而后我们在第 13 章中考察了 ADO.NET 和 Web 服务，侧重于介绍数据交换、使用 XML 以及安全性。

第 14 章再次讨论了 XML，这里集中介绍了 SQL Server 2000 如何对这一跨平台的数据检索标准提供本机支持。本章使用案例来讨论，详细阐述了所有本机 XML 选项的每个环节。

第 15 章则转入阐述更加理论化的性能和安全性问题。这两个主题对于我们每分钟都需要处理大量数据访问的需求特别重要。本章涵盖了提高性能和加强安全性的许多途径。

第 16 章讨论了集成和迁移问题，主要是通过 .NET 访问 ADO 和如何处理从 ADO 迁移到 ADO.NET。

第 17 章引导读者创建自定义的 .NET 数据提供者。解释了为什么需要自己的数据提供者，介绍了其体系结构和设计方法，并给出了在实际中实现的示例。本章也介绍了使用自定义的提供者的许多方法。第 18 章与此类似，作为本书的结尾，这一章提供了在一个多层系统的中间层使用 ADO.NET 的一个完整案例，即为一个虚构的城市内部自行车投递公司开发的包裹速递系统。

本书适合于哪些读者？

本书面向熟练的开发人员，假设读者具备使用 C# 或者 Visual Basic .NET 开发和使用 .NET 框架的实际经验。我们没有介绍有关 C# 和 Visual Basic .NET 的基本知识，并且假设读者已经熟悉 Microsoft 数据访问技术。

使用本书时需要做的准备

为运行本书中的示例，读者需要具有如下环境：

- Windows 2000 或者 Windows XP
- .NET Framework SDK。本书中的代码不支持 .NET B1 版本。

本书中的示例源代码可从 Web 站点下载：http://www.wrox.com/Books/Book_Details.asp?isbn=186100527X。

约定

在本书中，我们使用了大量不同格式的文本和结构，以便于区分不同的信息。这里给出了一些格式示例，并解释它们的用途。

代码有几种字体。如果是本书中谈论的一个文字——例如在讨论 For ... Next 循环时，它是这种字体。如果是一段能够组织成一个程序并且可以运行的代码片段，则使用灰色框标记：

```
<?xml version 1.0?>
```

有时，我们也使用混合格式的代码。例如：

```
<?xml version 1.0?>
```

```
<Invoice>
  <part>
    <name>Widget</name>
    <price>$10.00</price>
  </part>
</invoice>
```

在这种情况下，在白色背景下的代码是我们已经熟悉的代码；带有灰度的代码则是新加的代码。

建议、提示和背景信息使用下面的字体

重要的信息使用这种格式，并包含在方框中

用户支持

我们非常重视听取读者的意见，并且也希望知道读者关于本书的评论：你所喜欢的内容，不喜欢的内容，以及希望下次我们如何做得更好。你可以给我们发送自己的评论，可以填写本书后面的意见反馈卡寄给我们，也可以发送电子邮件到 feedback@wrox.com。请务必在信息中注明本书的书名。

如何下载本书的示例代码

当读者访问 Wrox 站点 <http://www.wrox.com/> 时，可以通过我们的搜索工具来查找本书，或者使用列出的书名找到本书。在 Code 列中点击“Download”，或者在本书的详细页面中点击“Download Code”。

本站点提供了可下载的 WinZip 压缩文件。当读者将该文件下载到本地硬盘上的一个目录后，需要使用 WinZip 或者 PKUnzip 等压缩工具将该文件解压。解压后，该文件通常会解压到 chapter 文件夹下。在进行解压操作时，要确保解压的目录是该文件夹。

勘误

我们尽量努力保证本书的文字和代码不出错。但是，错误是在所难免的。如果读者发现了本书的错误，例如，拼写错误或者是代码错误，我们会很感谢你的反馈。给我们反馈错误信息，会节省其他读者的理解时间，当然其他读者也会帮助我们提供很多信息。可以简单地将这些信息 E-mail 到 Support@wrox.com；您的信息将会被确认，如果正确，则我们会发布到勘错页面上，或者在本书后继的版本中使用。

为了找到我们的勘错页面，访问 <http://www.wrox.com/>，并且通过我们的“Advanced Search”或者书名来定位本书。点击本书详细页面中封面图下面的“Book Errata”链接即可。

E-Mail 支持

如果希望向本书专家直接讯问有关本书的问题，可以发送电子邮件到 support@wrox.com，

将书名和 ISBN 的后 4 为数字放入电子邮件的主题中。通常，电子邮件应包含如下的内容：

- 在“主题”行上，包含本书书名、ISBN 的后 4 位数字以及有问题的页码
- 在正文中包含你自己的名字、联系信息以及问题

不要发送笼统的邮件，我们需要详细的内容。在读者发送电子邮件信息时，我们有如下的系列支持：

- 客户支持——你的信息将会送到我们的客户支持部，他们首先查看你的信息。他们会给最常询问的问题归档，并回答有关本书或者 web 站点的各种问题
- 编辑——更深入的查询会转发给负责本书的技术编辑。他们都有很丰富的编程经验和实际产品开发经验，能够回答您所提出的技术问题
- 作者——最后，如果不凑巧，编辑也不能回答您的问题，他将会去询问本书的作者。我们会保护作者的著作名誉，但我们也会向他反映您的问题。所有 Wrox 的作者都会对他们的著作提供支持。他们会给客户和编辑发送电子邮件来回答他们的问题，这样所有的读者都会受益

Wrox 的只对与本书直接相关的问题提供支持。对于正常的支持之外的问题则由我们在 <http://p2p.wrox.com/forum> 的团体来支持。

P2P.wrox.com

与作者和同行的讨论可以加入 P2P 邮件列表。除了我们邮件支持系统之外，我们的系统提供 programmer to programmer™的邮件列表联系、论坛和新闻组。如果读者向 P2P 询问，可以得到许多 Wrox 作者和本邮件列表中的其他企业专家的回答。在 p2p.wrox.com 站点，除了在学习本书之外，在开发自己的应用程序时，也会有许多不同的列表会对你有所帮助。适合于本书的邮件列表有 ADO.NET 列表。

为了发送一个邮件列表，可以执行如下的步骤：

1. 访问 <http://p2p.wrox.com>
2. 从左侧菜单条中选择合适的目录
3. 想要加入的单击邮件列表
4. 在订阅提示下，填写您的电子邮件地址和密码
5. 回答所收到的电子邮件确认
6. 使用订阅管理器来加入更多的列表，并设置您的电子邮件偏爱。

为什么该系统能够提供最好的支持

读者可以选择加入邮件列表，也可以接收他们的每周摘要。如果没有时间或者环境接收邮件列表，你可以搜索我们的在线文档。我们会删除那些粗陋的电子邮件。我们会在 Lyris 系统中保留您自己的邮件地址。关于加入和离开列表的询问以及其他的一般询问都应该发送到 listsupport@p2p.wrox.com 中。

关于作者

Paul Dickinson

Paul 在其幼年时期就表现出了很高的天赋。他在 Atari ST 和 Kick Off 2 时就发现了自己的计算能力。人们从很远的地方跑来观赏他的数学能力。为了更好地发展自己，他进入 UMIST 研究计算理论。Paul 并不热衷于理论，因此毕业后就从事了几年的软件开发，编写控制物质分光器的软件。他最近忙于开发实验室的信息管理系统。现在，Paul 很少上机了，大部分时间都奔波于公司事务。他感谢自己的妻子 Sue 支持他全天候地工作。读者可以访问他的主页与他联系：www.pauldickinson.com

Fabio Claudio Ferracchiati

Fabio Claudio Ferracchiati 是一位软件开发者和技术作家。在其早期的十年职业生涯中，他利用传统语言和老的 Microsoft 工具，如 Visual Basic 和 Visual C++ 工作。五年后，他决定转向 Internet 及其相关技术。1998 年，他开始兼职为 Italian 和国际性杂志编写技术书籍。实际上，他是在罗马，使用 Microsoft 技术为 CPI Progetti (<http://www.cpiprogetti.it>) 开发 Internet/Intranet 解决方案。Fabio 要感谢 Wrox 员工和他的 CPI 同事们。他还特别感谢 Angelo 的帮助。

他还要感谢他的爱人 Danila。她对他的工作给予了很大的支持，在他编写本书期间，她一直陪伴在他的身边。另外，她在生活饮食上很好地照顾了 Fabio。每天都为他准备可口的饭菜，这是不容易的（他要对妻子说“饭菜可口”）。他觉得自己再也找不到 Danila 这样好的姑娘。谢谢！

Kevin Hoffman

Kevin Hoffman 是一个软件技术专家，对于 .NET 框架的所有技术和内容，他都了如指掌。起初他是在 Commodore VIC-20 上编写 BASIC 程序，这台机器还是他祖父从垃圾堆里捡出来修好给他用的，从此他便开始了他的技术生涯。作为一名工程师，他主持过国家的 UNIX 大型机软件开发，并且也是一家最著名的 Internet 电子商务 Web 站点的软件工程师。最近他开始编写并讲授有关编程技术方面的书籍。

Kevin Hoffman 能完成这部著作，首先要感谢他的母亲 Marie，她给了他很大的关怀和支持。另外，我也要感谢我的弟弟 Kurt 和弟妹 Nina，他们也给予了我很多的关心和鼓励。

Bipin Joshi

Bipin Joshi 是 Mumbai 的一名软件开发人员，擅长 Microsoft 技术。目前，他在 Mastek 公司从事 .NET 技术工作，Masktek 公司是印度的一家应用软件公司，是商用软件的开发中心。他的个人 Web 站点 www.bipinjoshi.com 提供了有关 .NET 各种主题的论文、手册和源代码。

他也为其他一些著名站点撰写文章。在工作之余，他潜心研究神学。读者可以访问他的站点与他联系：www.bipinjoshi.com

本书要感谢他的父母和兄弟，没有他们的大力支持和鼓励，本书也是无法完成的。

Donny Mack

Donny Mack 是一位 MCP、MCSD 和 Microsoft .NET MVP。他是华盛顿州人，也是 DotNetJunkies.com 站点的创始人之一，该站点是专门研究 ASP.NET 和其他 Web 相关的 .NET 技术的一个教育机构。DotNetJunkies.com 是一个免费的在线 Web 站点，是 .NET 开发者可以讨论 .NET 的地方。在与 Doug Seven 合作创立 DotNetJunkies.com 之前，Mack 在 Microsoft 公司担任 Visual InterDev/ASP 技术支持，专门为全球的开发者提供疑难解答和技术更新的咨询。Mack 跟踪前沿技术，他经常在闲暇时间里为 DotNetJunkies.com 编写代码和开发内容。他与 Doug Seven 以及 ASP.NET 开发小组一起研究技术发展问题。

John McTainsh

1982 年，John McTainsh 就在中学里的 Apple II 和 ZX81 上编程。主要从事 Motorola 芯片组的汇编程序开发和微控制器的编程工作。他在 1998 年完成了工程学位，而后转到亚洲的沿海石油和天然气开采公司工作。由于商业驱动，他很快又回到了编程领域中，使用 C 和 C++ 编写用于地下水探测器的控制系统程序。而后他编写了许多控制系统和各种其他应用程序和驱动程序，主要针对 PC 硬件。现在 John McTainsh 定居加拿大，并作为一个小组负责人主持开发用于计算机辅助分发系统的公共安全软件，可以应用到警车、救火车和救护车上。

John McTainsh 要感谢他的父母、妻子和女儿，他们给了他很大的支持。谢谢 Claudia、Rebecca、Dorothy 以及 Albert。

Matthew Milner

Matthew Milner 是明尼阿波利斯市的 BORN 技术总监，他设计和构建了用于各种工业客户的 Microsoft 解决方案。Matt 主要从事 Windows DNA 结构研究，并且准备转向研究 .NET 和所有新特性。在空闲时，Matt 喜欢游历明尼苏达州的许多自然资源。

Jan Narkiewicz

Jan D. Narkiewicz 是 Software Pronto 机构的技术总监 (jann@softwarepronto.com)。Jan 开始他的 Microsoft 开发人员生涯要归功于篮球明星迈克尔·乔丹。早在 20 世纪 90 年代初期，Jan 就注意到了不论篮球比赛上发生什么，迈克尔·乔丹所在的队总能获胜。类似地，不论在技术上有什么变化，Microsoft 总能获胜（这个概念是十年前的，现在可能发生了变化）。很明显这就是潮流。几年后 Jan 在一个拥有 1700 万客户的电子邮件系统工作，这里给了他充分发展自己的机会。他掌握了许多新技术和新概念，例如 COM/DCOM、COM+、C++、VB、C#、ADO、SQL Server、Oracle、DB2、ASP.NET、ADO.NET、Java、Linux 以及 XML 等。Jan 在其空闲时间里，他是 U.C 伯克利分校的 Windows 论坛的学术主席，讲授 U.C Santa Cruz 分校的老师，他还编写了《ASP Today》一书。另外他还踢踢足球。

Doug Seven

Doug Seven 是 DotNetJunkies.com 的另一个创始人，该站点是一个免费的 Web 站点，开发者可以在这里学习更多关于 .NET 框架——尤其是 ASP.NET 的知识。Seven 是作为 Nordstrom、Microsoft 和 GiftCertificates.com 的技术人员来到 DotNetJunkies.com 的，而且他还是西雅图咖啡公司的培训专家。在 Microsoft，Seven 是一个开发支持小组的技术骨干，他讲授 Visual Basic 和 ADO 中的类。在 Seven 离开 Microsoft，到 DotNetJunkies.com 之前，他是 GiftCertificates.com 的 Web 开发人员。

Seven 是 .NET 框架的许多资源书籍的作者，这些书籍包括：《Programming Data-Driven Web Applications with ASP.NET》和《ASP.NET: Tips, Tutorials and Code》等。他还为很多杂志和 Web 站点撰稿。

在他闲暇时候，他喜欢自娱自乐，喜欢编写程序回答别人的技术问题。他为人随和，喜欢喝好酒、坐在池塘边或者是晒太阳。

Doug Seven 能够完成这本书与他的家人和朋友的支持是分不开的。他要特别感谢 Dawniel Giebel 对我的理解、支持和生活上的照顾；感谢 Donny Mack 和他一起工作到深夜；感谢 Mark Anders、Rob Howard、Scott Guthrie 和 Susan Warren 很热心地回答他提出的问题；感谢 Dene Madam D. 'Holdsworth、Jason Pace 和 Tres Henry 这些 DotNetJunkies.com 上的忠实网友——某天他们的站点也将成为和 GotDotNet.com 一样的酷站；感谢 Wrox 出版社的 Alex Lowe、Trevor Scott、David Barnes、Claire Brittle、Craig McQueen 以及 Jeffery Hasan；感谢 Ed Hickey 的专业建议：“每天都坚持编写代码……即使是一行”；感谢 Jon Serious 让他能够浏览 SeriousDotNet.com；感谢 Lance Hayes 美妙的乡村音乐；感谢 Corinne “Jimi” Gurkey 和 Willy G. 永远都是那么酷；感谢 Dave Sceppa 当他在 Microsoft 时在 ADO 上给他的帮助。这样，他们才永远充满活力，才能够编写出这部书。

目 录

引 言

关于作者

第 1 章 数据访问和.NET	1
1.1 .NET 框架	1
1.2 为什么不使用其他数据访问技术.....	4
1.3 ADO.NET 体系结构.....	7
1.4 ADO.NET 和 ADO 2.6	18
1.5 使用 ADO.NET.....	25
1.6 小结	31
第 2 章 .NET 的数据提供者	32
2.1 SQL Server 数据提供者	33
2.2 OLE DB 数据提供者	33
2.3 用户接口	34
2.4 建立连接	39
2.5 使用命令	46
2.6 小结	54
第 3 章 Visual Studio .NET 和 ADO.NET	55
3.1 Connection 类.....	55
3.2 Command 数据组件.....	64
3.3 Data Adapter 组件.....	74
3.4 DataSet 数据组件.....	85
3.5 DataView 数据组件	96
3.6 DataGrid 组件	98
3.7 小结	107
第 4 章 DataReader 的使用	108
4.1 本章需要注意的一些地方	109
4.2 DataReader 基础	109
4.3 ADO.NET DataReader	111
4.4 DataReader 操作	113
4.5 综合应用	126
4.6 通常可能产生的异常	130

4.7	DataReader 性能问题	131
4.8	小结	134
第 5 章	DataSet	135
5.1	DataTable	136
5.2	填充 DataSet	152
5.3	Tables 集合	156
5.4	Relations 集合	161
5.5	合并 DataSet	167
5.6	缓存 DataSet 以提高性能	170
5.7	小结	173
第 6 章	DataAdapter 的使用	174
6.1	DataAdapter 和 DataSet	175
6.2	更新数据库	188
6.3	小结	197
第 7 章	类型化 DataSet 和 DataSet 模式	198
7.1	XSD 概述	198
7.2	DataSet 模式	210
7.3	类型化的 DataSet	217
7.4	注释类型化的 DataSet	225
7.5	小结	230
第 8 章	XML 和 DataSet	232
8.1	XmlDocument (W3C DOM)	233
8.2	XPath	235
8.3	DataSet 模式	240
8.4	DataSet 和 XML 数据	249
8.5	DataSet 调度	255
8.6	数据过滤	259
8.7	DataSet 和 XmlDataDocument	265
8.8	使用 XSL 和 XSLT 转化	270
8.9	小结	273
第 9 章	约束、关联和视图	275
9.1	约束	275
9.2	DataRelation	290
9.3	DataView	298
9.4	综合	308
9.5	小结	318

第 10 章 事务	319
10.1 什么是事务	319
10.2 ADO.NET 的事务支持	321
10.3 编写事务的数据库的应用程序	323
10.4 高级技术	332
10.5 小结	338
第 11 章 映射	339
11.1 使用 SQL 的 AS 关键字	339
11.2 ADO.NET 的映射机制	340
11.3 使用映射的 Web 服务	348
11.4 小结	358
第 12 章 制作数据服务组件	359
12.1 安装 ODBC.NET	360
12.2 什么是数据服务组件以及为什么使用它	360
12.3 创建一个数据服务组件	362
12.4 部署数据服务组件	373
12.5 使用数据服务组件	376
12.6 性能和优化技巧	388
12.7 卸载组件	399
12.8 小结	399
第 13 章 ADO.NET 和 Web 服务	401
13.1 安装代码示例	402
13.2 Web 服务——新的 DCOM	403
13.3 通用标准	403
13.4 创建基本的 Web 服务	405
13.5 创建基本的消费者	409
13.6 在 Web 服务中交换数据	427
13.7 Web 服务的安全性	445
13.8 小结	454
第 14 章 SQL Server 的内置 XML 支持	455
14.1 FOR XML	456
14.2 OPENXML	478
14.3 小结	483
第 15 章 性能和安全	485
15.1 优化数据访问	485
15.2 连接池	503

15.3	消息排列	505
15.4	安全问题	508
15.5	小结	517
第 16 章	集成和迁移	518
16.1	InterOp	518
16.2	迁移	524
16.3	小结	551
第 17 章	创建自定义.NET 数据提供者	552
17.1	数据提供者库	553
17.2	应用程序需求	553
17.3	结构和设计	554
17.4	实现数据提供者配件	556
17.5	自定义数据提供者的使用	591
17.6	小结	597
第 18 章	实例学习——自行车递送公司	598
18.1	需求	599
18.2	设计	601
18.3	实现	607
18.4	如何部署系统	624
18.5	如何编译系统	626
18.6	小结	627

数据访问和.NET

本章将对 ADO.NET 作简要的概述。在此不介绍具体的代码，而是简述其概念，这是快速了解 ADO.NET 的有效方法。希望本章能够使读者对 ADO.NET 的基本工作原理有一个初步的了解，对其显著特性有大致地了解。在阅读本章之后，希望读者能够熟悉 ADO.NET 的优点，而后进一步开始学习本书！

ADO.NET 是由 Microsoft 发布的一种数据访问技术。不同于以往的技术，这种技术是构建在称为 .NET 框架的新平台上的。这个平台将革新工程开发的各个领域，ADO.NET 只是其中的一部分。因此，下面将简化介绍一下 .NET 的主要特性。

1.1 .NET 框架

可以毫不夸张地说，Microsoft 发布的这套新的开发平台及其运行时环境和 .NET 框架将在 Microsoft 程序设计的各个方面引发一场轰动性的革命，该平台的优点体现在开发应用程序和代码的各个方面。.NET 框架本身是一个非常庞大的主题，在此无法详细阐述它的每个小细节，但是在利用 ADO.NET 进行开发之前，了解 NET 体系的基本原理是非常重要的，因此本章将简述其基本原理。

有关 .NET 编程的更详细信息，请参阅《Professional .NET Framework》，ISBN 1-681005-56-3。

1.1.1 通用语言运行时

.NET 框架是建立在 CLR (Common Language Runtime, 通用语言运行时) 基础之上的。CLR 是在运行时管理 .NET 代码的执行环境，从某种意义上讲，它与 JVM (Java Virtual Machine, Java 虚拟机)、Visual Basic 6 运行时 (msvbvm60.dll) 相似。类似地，.NET 框架也需要安装在某台机器上。但是，所不同的是 CLR 专门设计用于支持多种不同语言编写的应用程序。事实上，许多语言是针对 JVM 开发的 (目前已超过 .NET)，但是多种语言支持功能并不是 JVM 最初的设计动机。对 CLR 而言，这确实是最重要的功能。

为了获得跨语言的支持，在将 .NET 程序部署到被称为 IL (Intermediate Language, 中间语言) 的低级语言上之前，应该事先编译 .NET 程序。Microsoft 所实现的这种语言，被称为 MSIL (Microsoft Intermediate Language, Microsoft 中间语言)。而后这种 IL 代码在运行过程

中即时地被编译为本机代码。也就是说，无论源代码是什么语言，.NET 可执行程序 and DLL 总是部署在 IL 上，因此最初用 C# 编写的组件与用 VB.NET 编写的组件已没有什么区别。这有助于实现跨语言的互操作性（例如，用某种语言编写的类可以继承其他 .NET 语言编写的类）。但是，它也允许应用程序不经过改动直接部署到其他支持的平台上（目前的系统平台有 Windows 9x/ME、Windows NT4、Windows 2000 或者 Windows XP），例如 JIT 编译器能够针对部署的计算机的处理器/OS 进行优化。

Microsoft 为如下几种 .NET 语言提供了编译器：

- ❑ C#——一种新设计的支持 NET 框架的 C 语言。本书中的绝大部分代码都是 C# 代码
- ❑ Visual Basic .NET——支持 .NET 的一种 Visual Basic 语言版本。（具有完整的面向对象特性、结构化的异常处理机制以及 VB 开发人员所希望的其他多种特性）
- ❑ JScript.Net——Microsoft 实现的 JavaScript 脚本语言，支持 .NET
- ❑ 受管 C++——具有“管理功能扩展”的 C++ 语言，支持那些使用该语言的已有特性不能实现的 .NET 功能特性。与其他 3 种语言不同，C++ 编译器不使用 .NET 框架 SDK，而是运行在 Visual Studio .NET 环境中
- ❑ J#——实质上是支持 .NET 框架的 Visual J++ 语言（包含 Microsoft 对 Java 的扩展，例如 COM 支持）。本书编写的时候，J# 发布了 β1 版本，该版本可以从 <http://msdn.microsoft.com/visualj/jsharp/beta.asp> 下载

除了这些 Microsoft 语言外，第三方公司可以支持更多类型的语言（如 COBOL、Perl 以及 Eiffel 等）。有关这三种语言的更多信息，请访问如下站点：COBOL 的 <http://www.adtools.com/dotnet/index.html>、Perl 的 <http://aspn.activestate.com/ASPN/Downloads/PerlASPX/More> 以及 Eiffel 的 http://msdn.Microsoft.com/library/techart/pdc_eiffel.html。

1. 垃圾收集器

CLR 提供的极为重要的服务之一是垃圾收集器。在 C 和 C++ 中，如果一个对象被实例化，它所使用的内存在重新利用之前需要先被释放。如果不释放内存，则将导致“内存泄漏”——也就是那些不能由系统再使用的空闲内存块。随着泄漏的内存空间的增加，应用程序的运行效率将会大大降低。但是由于错误不明显而且也仅仅是拖延了时间，因此这些错误很难跟踪定位。通过使用垃圾收集器，CLR 解决了此类问题。在某些特定时期（当堆中没有空间时），垃圾收集器将检查所有的对象引用，并且释放那些已超出运行范围和应用程序不再访问的对象所占有的内存空间。这使得程序员从手工释放这些类的操作中解脱出来，而且解决了内存泄漏问题。这里有两点需要注意：第一，因为无法精确预测垃圾收集器何时开始工作（虽然能够强制进行空间收集操作），所以在完成对象操作之后，对象仍可能驻留在内存中一段时间；第二，CLR 不会清除非受管资源——程序员需要亲自去做这些事。最为有效的清除方法是使用一种称为 Dispose 的方法，该方法将释放所有外部资源，可以在完成对象操作时调用该方法。

2. 通用语言基础设施

虽然 .NET 框架现在只是在 Windows 平台上可用，Microsoft 已经向 ECMA（European Computer Manufacturer's Association，欧洲计算机厂商协会）提交了 .NET 的一个子集，即

CLI (Common Language Infrastructure, 通用语言基础设施), 以促使 .Net 成为一个开放的标准。为 FreeBSD 和 Linux 操作系统平台设计的 CLI 正在开发中。类似地, C# 和 IL (后来称为 CIL) 的规范也已经提交给 ECMA, 那些非 Microsoft 实现的 CLI 也都包含在其中。

1.1.2 配件

.NET 代码可以部署为配件。配件由一些编译了的 IL 代码组成, 它必须包含一个主文件 (动态配件情况除外, 它可以整个存储在内存中)。它可以是一个可执行文件、一个 DLL 或一个编译了的 ASP.NET Web 应用程序或 Web 服务。除了主文件外, 配件还要包含资源文件 (如图像或图标等) 和其他代码模块。最为重要的是配件包含元数据 (metadata)。元数据由两部分组成: 类型元数据 (type metadata) 及其在配件中定义的方法, 其中类型元数据包含所有导出的类型。与 IL 代码类似, .NET 的配件包含一个清单 (manifest) 部分。该部分包含配件元数据或有关配件自身的信息, 例如版本和创建号码。

这种元数据使得配件具有完整的自描述特性: 配件本身包含所有安装和运行应用程序所需的信息, 因而不需要任何类型库和注册表项。安装程序只需将配件拷贝到目标机器上。另外, 由于配件包含版本信息, 同一组件的多种版本可以同时安装到同一台机器上。这将避免出现“DLL 冲突”——当某应用程序开始安装某些已存在的组件的新版本时, 将会破坏那些使用老版本的程序的运行。

1.1.3 通用类型系统

CLR 跨语言特征的基础是 CTS (Common Type System, 通用类型系统), 为了使不同语言定义的类能够相互通信, 需要一种通用方法来表示数据——组通用数据类型。IL 中所有预定义类型都定义在 CTS 中。这意味着 .NET 代码中的全部数据最终以相同的数据类型存储, 因为所有 .NET 代码都编译成 IL。

CTS 有两种基本数据类型: 值类型 (value Type) 和引用类型 (reference Type)。值类型 (包括大部分内置类型以及结构和枚举类型) 直接包含数据。例如, 整数类型变量直接把整数存储于程序堆栈中。引用类型 (包括 String 和 Object, 以及数组和大部分用户定义类型, 如类和接口) 只在堆栈数据中存储其引用指针——数据本身存储于存储器不同区域中, 我们称这些区域为堆。当向方法传递参数时, 这些类型的差异会很明显。所有方法的参数其默认的传送方式是传值, 而不是传引用指针。然而在引用类型中, 引用不过是存储数据的堆的位置。结果, 引用类型参数的行为与我们所期望的一样——在方法体中改变变量值也会影响原始变量。如果需要把 ADO.NET 对象传入方法, 要牢记这一点。

通用语言规范

对于 CTS, 值得注意的一点是: 并非所有特性都可以用所有语言来描述。例如, C# 中包含带符号字节数据类型, 而该类型在 Visual Basic .NET 中却找不到。这就可能会引起语言互操作性方面的问题, 所以 CLS (Common Language Specification, 通用语言规范) 定义了 CTS 的一个子集, 并且所有编译器必须支持该子集。这极有可能暴露那些不属于 CLS 的特性 (例如, 某个带有 sbyte 类型公共属性的 C# 类)。但是, 切记这样的特性并不保证其他语言能够访问到它——在此例中, 不能从 VB.NET 访问该类。