



Professional ADO.NET with VB.NET

ADO.NET

高级编程

—— VB.NET 编程篇

(英) Paul Dickinson
Fabio Claudio Ferracchiati 等著
谭建波 李立新 译



清华大学出版社

ADO.NET 高级编程

—— VB.NET 编程篇

(英) Paul Dickinson
Fabio Claudio Ferracchiati 等著

谭建波 李立新 译

清华大学出版社
北京

内 容 简 介

ADO.NET 是 Microsoft 在 .NET 环境下使用的一种最新数据访问技术, 其功能强大, 使用方便。本书采用了大量实例和技术, 介绍了 ADO.NET 的应用方法, 主要内容涉及 ADO.NET 数据提供者的体系结构、数据读取器、数据适配器和 DataSet, 使用 XML 模式和强类型化数据集开发安全的应用程序、ADO.NET 对事务处理的支持, 以及如何定制数据提供者。

本书适合那些在 .NET Framework 中开发过 Visual Basic .NET 应用程序, 或是已经熟悉如何利用 ADO 开发数据驱动的应用程序, 而且希望掌握 ADO.NET 的中高级开发人员。

EISBN: 1-86100-806-6

Professional ADO.NET with VB.NET

Paul Dickinson, Fabio Claudio Ferracchiati et al.

Copyright© 2002 by Wrox Press Ltd.

Original English language edition published by Wrox Press Ltd.

All rights reserved.

本书中文简体字版由英国乐思出版公司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。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

北京市版权局著作权合同登记号 图字: 01-2002-6520

图书在版编目(CIP)数据

ADO.NET 高级编程——VB.NET 编程篇/(英)迪克松等著; 谭建波 李立新译. —北京: 清华大学出版社, 2003
书名原文: Professional ADO.NET with VB.NET

ISBN 7-302-06913-1

I. A… II. ①迪…②谭…③李… III. ①数据库—接口—程序设计②BASIC 语言—程序设计 IV. TP311.11

中国版本图书馆 CIP 数据核字(2003) 第 060012 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: 康博

版式设计: 康博

印 刷 者: 世界知识印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 33.75 字数: 863 千字

版 次: 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

书 号: ISBN 7-302-06913-1/TP·5116

印 数: 1~4000

定 价: 68.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前 言

ADO.NET 是 Microsoft 的最新数据访问技术，作为 .NET Framework 的重要组成部分，它已不再只是 ADO 的更新产品。ADO.NET 提供了大量 .NET 类，促进了对各种数据源中数据的有效访问，可以对数据进行复杂的处理和排序，形成了一个可在其中进行应用程序间通信的重要架构。

ADO.NET 已经成为所有数据驱动的 .NET 应用程序或是 Web 服务的核心组件。了解 ADO.NET 的强大功能和灵活性对于想有效利用 .NET 传输技术的人来说很重要。

本书主要内容

本书采用了一个实用的、面向解决方案的方法，全面讨论了 ADO.NET 类。讨论了如何在以数据为中心的应用程序开发环境中使用各种 ADO.NET 的组件。

第 1 章介绍了 .NET 数据提供者——它是一个提供与各种数据源连接的类集。随后的几章深入讨论了 ADO.NET 的具体内容：第 2 章介绍了数据读取器——介绍了数据读取器的概念，使用它们的原因，以及使用它们的时机。第 3 章介绍了 DataSet，第 4 章则介绍和讨论了数据适配器类。

第 5 章进一步讨论了 DataSet，使用它可以在与数据源断开连接的情况下处理数据。其中介绍了 XML schemas 在操作 DataSet 时的作用。第 6 章详细讲述了 XML 在 DataSet 中的使用，还讨论了诸如 XPath 和数据验证方面的问题。

第 7 章结束了对 DataSet 的讨论，其中讲述了约束、关系和视图，所有这些都影响表示和操作数据的方式。本章还介绍了 DataView，包括使用它的一些例子。

第 8 章讨论了事务处理，这一主题在商务中极为重要，其中数据源上的所有操作必须都成功或者必须都失败。本章还介绍了隔离级别，它们的影响、性能，以及一些高级的事务处理技术。

第 9 章研究了映射的概念。我们可以使 ADO.NET 为代码中的表和列提供不同于其在原数据源中的名称，使得应用程序更为直观。

第 10 章介绍创建我们自己的组件来访问数据服务，讨论这样做的好处，在它存在时如何部署它，以及如何使用它。另外还介绍了获得更好性能的技巧。

第 11 章介绍了 ADO.NET 和 Web 服务——特别是使用 XML 交换数据和安全性。

第 12 章又介绍了 XML，但这次是从数据库而不是 ADO.NET 的角度来讨论它。我们介绍了 SQL Server 2000 对 XML 的固有支持，可以定制服务器和开发人员在应用程序中的操作。

第 13 章更理论化地讨论性能和安全性。当然，如果要处理高级要求或是敏感数据，那么就要重点考虑它们了，因此本章介绍了提高性能和安全性的许多方法。

第 14 章讨论了集成和迁移——特别讲述了从 .NET 中访问 ADO 数据源，以及由 ADO 向

ADO.NET 进行移植。

最后，第 15 章讨论如何创建定制的 .NET 数据提供者。介绍了案例分析的整个过程——从确定所需的数据提供者，然后是体系结构和设计，到最终实现。本章还演示了使用这样一种提供者的多种方法。

本书读者对象

本书适合于那些已在 .NET Framework 中开发过 Visual Basic .NET 应用程序，或是有使用 ADO 开发数据驱动的应用程序的经验，现在又希望转向 .NET 的开发人员。它主要针对的是有相当经验的编程人员，而不是业余的 ASP.NET 开发人员或是初学者。本书并没有介绍 Visual Basic .NET 或 Microsoft 数据访问技术的基础知识。

使用本书的条件

要学习并运行本书中的示例，需要下列条件：

- Windows 2000 或 Windows XP
- .NET Framework SDK
- Visual Studio .NET(标准版或专业版，不过推荐使用后者)
- Microsoft Data Engine (MSDE)或 SQL Server 2000

示例的完整源代码可从我们的 Web 站点 www.wrox.com/books/1861008066.htm 上下载。

客户支持

我们一贯重视读者的意见，并想知道每位读者对本书的看法，包括读者喜欢和不喜欢的内容，以及读者希望我们下一次完善的地方。您可以通过发送电子邮件(地址为 feedback@wrox.com)来向我们反馈意见。请确保在反馈信息中提到本书的书名。

如何下载本书的示例代码

当您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>)时，单击主页上方的 DOWNLOAD CODE 链接，选择出现在结果页面的文本框中的书名。在开始下载之前，可以通过提供姓名和目前电子邮件地址注册。这完全是自愿行为，如果下载过程中出现问题或是代码下载包需要更新，就可方便与您联系。Wrox 可以保证不会将任何注册信息提供给第三方。详细信息可参看已链接到下载页的 Privacy Policy(保密条款)。

从我们的站点上下载的可用文件都是使用 WinZip 压缩过的文档，需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时，通常将代码解压缩到每一章所在的文件夹中。因此在解压缩前，应确保将解压缩软件设置为使用原有文件夹名。

勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误，但是错误仍然在所难免。如果您发现本书存在错误，例如拼写错误或不正确的代码段，请给我们反馈信息，我们将不胜感激。勘误表的发送可以节约其他读者学习本书的时间，而且能够帮助我们提供更高质量的信息。您的反馈信息将被检查，如果正确，将被粘贴到本书的勘误页面上，或者在本书的后续版本中使用。

要在我们的站点上找到勘误表，请访问 <http://www.wrox.com/>，并通过 Search 工具就可轻松定位到本书页面。找到本书后，只需单击 View errata 链接即可。

E-mail 支持

如果您希望直接向详细了解本书的专家咨询本书中的问题，可以发送电子邮件到 support@wrox.com，要求在邮件的主题栏中带上本书的书名和 ISBN(国际标准图书编号)的后 4 位数字。一封典型的电子邮件应包括下面的内容：

- 在主题栏中必须有本书的书名、ISBN 的后 4 位数字和问题所在的页码。
- 正文部分应包括读者的名字、联系信息和问题。

我们将不返回无用邮件，因为我们仅仅需要有用的详细资料，以便可节约您和我们的时间。当您发送一个电子邮件信息时，它将经过下面一系列支持：

- 客户支持。首先，您的信息将被递送到我们的客户支持人员手中，并由他们阅读。他们备有常见问题的文件，并将立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑支持。接着，一些有深度的问题将被送到对本书负责的技术编辑手中，他们在程序设计语言或者特定的产品上有着丰富的经验，能够回答相关主题的技术问题。
- 作者支持。最后，如果编辑不能回答您的问题(这种情况很少发生)，他们将请求本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书提供技术支持。作为回应，他们将发送电子邮件给用户和编辑，进而使所有的读者受益。

Wrox 公司的支持过程仅仅对那些与我们出版的书目内容直接相关的问题提供支持，对于超出常规书目支持的问题，您可以从 <http://p2p.wrox.com/>论坛的公共列表中获得支持信息。

p2p.wrox.com 站点

为了便于作者和其他人讨论，请加入到 P2P 站点的邮件列表中，除了一对一的邮件支持系统外，我们独特的系统将 programmer to programmer™(由程序员为程序员而著)的编程理念与邮件列表、论坛、新闻组等其他服务相联系。如果您向 P2P 发送一个问题，应该相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家检查到。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 p2p.wrox.com 站点中找到许多对自己有所帮助的邮件列表。

按照下面的步骤可以预订一个邮件列表：

- (1) 登录 <http://p2p.wrox.com/> 站点。
- (2) 从左边的菜单栏选择一个适当的类别。
- (3) 单击您希望加入的邮件列表。

- (4) 按照说明订阅并填写自己的邮件地址和密码。
- (5) 回复您收到的确认邮件。
- (6) 使用预订管理程序加入更多的邮件列表并设置自己的邮件首选项。

本系统提供最佳支持的原因

您可以选择加入到整个邮件列表，也可以只接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。当存在加入和离开列表，以及任何有关列表的其他常见问题时，请发送邮件到 listsupport@p2p.wrox.com。

目 录

第 1 章 ADO.NET 概论	1
1.1 ADO.NET 体系结构	1
1.2 .NET 数据提供者	2
1.3 DataSet 类	6
1.4 建立到数据源的连接	8
1.4.1 具体的连接类	9
1.4.2 数据提供者中的连接入池	12
1.5 命令的使用	13
1.6 ADO.NET 和 ADO 2.6	20
1.6.1 断开连接的数据访问	20
1.6.2 只读只前向的访问	20
1.6.3 提供者特有的类	21
1.6.4 在 .NET 中使用 ADO 2.x	22
1.7 小结	23
第 2 章 使用数据读取器	24
2.1 数据读取器的基础知识	24
2.1.1 IDataReader 接口	24
2.1.2 IDataRecord 接口	25
2.2 数据读取器操作	26
2.2.1 导航多个结果集	26
2.2.2 采用类型安全的方式访问数据	27
2.2.3 获取结果集的模式	30
2.3 综合运用	32
2.4 常见的异常	36
2.4.1 IndexOutOfBoundsException	36
2.4.2 InvalidOperationException	36
2.5 与数据读取器性能有关的问题	36
2.5.1 列序号与列名称	37
2.5.2 类型安全和非类型安全的访问	38
2.6 小结	39
第 3 章 数据集	40
3.1 DataSet 对象模型	40

3.2	DataTable	41
3.2.1	DataColumn	42
3.2.2	DataRow	43
3.2.3	约束	43
3.2.4	PrimaryKey	44
3.2.5	动态构建一个 DataTable	44
3.2.6	DataTable 事件	45
3.2.7	DataTable 事件的示例	46
3.3	填充 DataSet	56
3.3.1	构造一个数据适配器	56
3.3.2	调用 Fill 方法	57
3.3.3	检索表集合的元数据	58
3.3.4	用多个 DataTable 填充表集合	60
3.4	Relations 集合	60
3.5	合并 DataSet	67
3.5.1	合并两个 DataSet	68
3.5.2	合并两个 DataSet 并保留原始值	68
3.5.3	合并两个有着不同模式的 DataSet	68
3.6	缓存 DataSet 获取更好的性能	69
3.7	小结	72
第 4 章	数据适配器的使用	74
4.1	数据适配器	74
4.1.1	利用更复杂的查询	74
4.1.2	利用少量记录填充 DataSet 对象	76
4.1.3	利用存储过程填充 DataSet	79
4.2	数据库的更新	80
4.2.1	行状态和字段值形式	80
4.2.2	修改 DataSet 中的记录	81
4.2.3	RowState 和更新操作	83
4.3	数据适配器事件	88
4.4	并发管理	90
4.4.1	并发控制的类型	90
4.4.2	ADO 中的并发	92
4.4.3	手工控制并发问题	92
4.4.4	利用数据适配器控制并发	95
4.4.5	解决并发问题	97
4.5	小结	98

第 5 章 类型化 DataSet 和 DataSet 模式	99
5.1 XSD 概述	99
5.1.1 简单类型	100
5.1.2 用户定义的类型	103
5.1.3 复杂类型	104
5.1.4 元素组	105
5.1.5 属性组	107
5.1.6 XSD 注释	107
5.1.7 XmlSchema 类	108
5.2 DataSet 模式	110
5.2.1 模式转换	111
5.2.2 约束	113
5.3 强类型化的 DataSet	118
5.3.1 在 Visual Studio .NET 中构建一个类型化的 DataSet	118
5.3.2 手工构建类型化 DataSet	123
5.3.3 强类型化 DataSet 和关系数据	123
5.3.4 填充已具有模式的 DataSet 对象	125
5.3.5 类型化 DataSet 的性能	125
5.4 注释类型化 DataSet	125
5.4.1 codegen 命名空间	126
5.4.2 msdata 命名空间	127
5.4.3 注释过的类型化 DataSet 示例	128
5.5 小结	131
第 6 章 XML 和 DataSet	132
6.1 XmlDocument (W3C DOM)	132
6.2 XPath	134
6.2.1 轴	136
6.2.2 节点测试	137
6.2.3 谓词	137
6.2.4 使用 XPath 的示例	137
6.3 DataSet 模式	139
6.3.1 模式推断	139
6.3.2 提供的模式	144
6.3.3 利用模式进行文档验证	146
6.4 DataSet 和 XML 数据	149
6.4.1 加载 XML	149
6.4.2 编写 XML	152

6.4.3 失真和 DataSet 模式	153
6.5 DataSet 和 XmlDataDocument	155
6.5.1 通过 XSD 实现的 XML 文档的关系投影	155
6.5.2 类型化 DataSet 的关系投影	159
6.6 小结	160
第 7 章 约束、关系和视图	161
7.1 约束	161
7.1.1 惟一性约束	162
7.1.2 外键约束	164
7.1.3 引用完整性	165
7.1.4 定制约束	167
7.2 DataRelation	175
7.3 DataView	181
7.3.1 排序	182
7.3.2 过滤	183
7.3.3 根据行的状态过滤	186
7.3.4 在 DataView 中编辑数据	188
7.3.5 DataViewManager	189
7.3.6 数据绑定	190
7.4 综合运用	192
7.5 小结	201
第 8 章 事务处理	202
8.1 什么是事务处理	202
8.1.1 ACID 属性	202
8.1.2 数据库事务处理	203
8.1.3 事务处理词汇	204
8.2 ADO.NET 事务处理支持	204
8.3 编写事务处理型数据库应用程序	207
8.3.1 实现事务处理	207
8.3.2 测试使用隔离级别的效果	212
8.3.3 何时使用事务处理	216
8.3.4 事务处理和性能	217
8.3.5 事务处理的默认行为	217
8.3.6 事务处理和用户确认	218
8.3.7 同时进行 ADO.NET 和 DBMS 事务处理	218
8.4 高级技巧	218
8.4.1 保存点	219

8.4.2 嵌套的事务处理	221
8.4.3 与 DataSet 和数据适配器一起使用事务处理	221
8.5 小结	224
第 9 章 映射	225
9.1 利用 SQL 的“AS”关键字	225
9.2 ADO.NET 映射机制	226
9.2.1 在检索数据时利用映射	226
9.2.2 利用映射的名称插入记录	230
9.3 利用映射的 Web 服务	234
9.3.1 创建供应商数据库	236
9.3.2 创建供应商 Web 服务	237
9.3.3 创建 Pet Lovers 应用程序	238
9.4 小结	244
第 10 章 构建数据服务组件	245
10.1 数据服务组件的概念以及使用它的原因	245
10.1.1 什么是数据服务组件	246
10.1.2 数据服务组件的优点	247
10.2 创建一个数据服务组件	247
10.3 部署数据服务组件	257
10.4 使用数据服务组件	259
10.4.1 在 ASP.NET Web 窗体中使用 DAL 组件	259
10.4.2 在 Web 服务中使用 DAL 组件	267
10.5 性能和优化技巧	269
10.6 小结	275
第 11 章 ADO.NET 和 Web 服务	277
11.1 安装示例代码	278
11.2 Web 服务——新的 DCOM	279
11.3 通用标准	280
11.4 构建一项基本的 Web 服务	281
11.5 构建一个基本的使用者	285
11.5.1 构建一个 HTTP 使用者	286
11.5.2 用 Visual Studio .NET 构建一个 SOAP 使用者	290
11.5.3 什么是代理客户机	299
11.5.4 wsdl.exe 实用程序	301
11.6 Web 服务中的数据交换	303
11.6.1 处理 DataSet	304

11.6.2	使用 XML 和 Web 服务	314
11.7	Web 服务安全	321
11.7.1	使用 Windows 身份验证方法	322
11.7.2	使用基于 SOAP 的身份验证	324
11.8	小结	329
第 12 章	SQL Server 的本地 XML 支持	330
12.1	FOR XML	331
12.1.1	FOR XML 的可选参数	333
12.1.2	FOR XML RAW	334
12.1.3	FOR XML AUTO	336
12.1.4	FOR XML EXPLICIT	339
12.2	OPENXML	348
12.3	SQLXML 的托管类	352
12.3.1	SQLXML 还是 ADO.NET	352
12.3.2	SQLXML 托管类的体系结构	352
12.3.3	SQLXML 对象模型	354
12.4	小结	370
第 13 章	性能和安全	371
13.1	优化数据访问	371
13.1.1	选择数据读取器还是数据集	371
13.1.2	往返操作	373
13.1.3	存储过程	373
13.1.4	配置 DataAdapter 命令	375
13.1.5	大量数据的处理	379
13.1.6	编组问题	388
13.1.7	DataSet 串行化	389
13.1.8	通过 HTTP 传递 XML	391
13.2	连接入池	391
13.2.1	SqlConnection	391
13.2.2	OleDbConnection	392
13.3	消息队列	392
13.3.1	是否排队	393
13.3.2	发送消息	393
13.3.3	接收消息	394
13.4	安全问题	396
13.4.1	代码访问安全	396
13.4.2	CAS 应用	398

13.4.3	SSL	402
13.4.4	加密	403
13.5	小结	405
第 14 章	集成和迁移	406
14.1	互操作	406
14.1.1	RCW 和 COM 的互操作	406
14.1.2	从.NET 访问 ADO	406
14.1.3	平台调用服务(PInvoke)	410
14.2	迁移	410
14.2.1	ADO 数据类型	411
14.2.2	迁移连接	412
14.2.3	迁移 Recordset	413
14.2.4	迁移命令和存储过程	418
14.2.5	XML 保存中的变化	421
14.2.6	处理异常和错误	424
14.2.7	流	427
14.3	小结	429
第 15 章	创建自定义的.NET 数据提供者	430
15.1	数据提供者库	430
15.2	应用程序要求	431
15.2.1	零售商店	431
15.2.2	电子商务网站	431
15.2.3	电话销售	431
15.3	体系结构和设计	431
15.4	实现数据提供者程序集	433
15.4.1	OQProvider 命名空间	434
15.4.2	OrderObject	435
15.4.3	OrderItem	437
15.4.4	MSMQ 知识回顾	439
15.4.5	OQConnection	442
15.4.6	OQCommand	446
15.4.7	OQDataReader	455
15.4.8	OQDataAdapter	461
15.4.9	OQException	466
15.5	使用自定义数据提供者	467
15.5.1	零售商店接口	467
15.5.2	电子商务 Web 站点接口	469

15.5.3 电话销售接口.....	471
15.6 小结.....	473
附录 A Visual Studio .NET 和 ADO.NET.....	474
A.1 连接类.....	474
A.2 命令数据组件.....	480
A.3 数据适配器组件.....	486
A.4 DataSet 数据组件.....	497
A.4.1 类型化数据集.....	498
A.4.2 生成并填充类型化数据集对象.....	503
A.5 DataView 数据组件.....	507
A.5.1 使用 DataView 查看选择的行.....	507
A.5.2 使用 DataView 排序数据行.....	509
A.6 DataGrid 组件.....	510
A.6.1 DataGrid Web 组件.....	510
A.6.2 DataGrid 窗口组件.....	518
A.7 小结.....	519

第1章 ADO.NET 概论

在如今这个 Internet 盛行、信息极为丰富的世界中，几乎所有应用程序都需要访问这样或那样格式的数据。在某些情况下会将数据存储在接受诸如 Microsoft SQL Server 或 Oracle 等产品控制的数据库中。而在另一些情况下，则会将数据存储的电子表格或 XML 文件，或是大量有着不同专用数据格式的文件中。不过，不管数据存储的具体情况如何，都必须采用某种技术允许应用程序连接到数据，并执行命令对它进行处理。在 .NET Framework 中，这样的技术就是 ADO.NET。

从实际应用来看，ADO.NET 是一个类库，在 .NET Framework 的 System.Data 命名空间中定义这些类——这些类表示到数据库的连接、数据库中的表，和表中所包含的数据类型。更广义地讲，ADO.NET 就是 Microsoft 最新的数据访问策略——仍然是实现通用数据访问的最佳做法。

本章将首先简要介绍 ADO.NET，然后再集中讨论一些关键功能。因此，根据设计，我们将快速讲解这些知识，并辅以代码片断进行说明——这是掌握各种概念的最佳方法。最后，读者应该可以很好地掌握 ADO.NET 的基本知识，及其令人兴奋不已的功能。

1.1 ADO.NET 体系结构

ADO.NET 对象模型由两个基本组件构成：.NET 数据提供者和数据集。尽管这两个组件必须配合使用，但是它们却表示两种截然不同的与数据进行交互的方式。从本质上讲，数据提供者(这里有几个)专门用于特定类型的数据源，开发人员可以使用这种有效的途径直接处理数据源中的数据。另一方面，数据集则可以看作是一种数据存储，它并不是专门用于处理某种特定数据源的数据，不过它却有强大的功能用于操纵其所包含的数据。

从这些定义可知，这是两种数据访问策略。第一种策略使用数据提供者连接到数据源，先对数据源稍作处理(包括简单的读数据操作)，然后再断开连接。第二种策略依然使用数据提供者连接到数据源，不过它会将数据复制到数据集中，再断开连接。接下来，就可以在空闲的时候使用这些数据集中的数据——如果某个阶段需要修改数据源，则可以重新连接到数据源并修改数据。

由于 .NET 数据提供者是这两种访问策略的关键，所以需要它们轻型而高效，在应用程序和数据源之间创建一个最小的层。实际应用中，.NET 数据提供者通常由下面 4 个主要对象构成：

- 连接(connection)对象，此对象连接到数据源。
- 命令(command)对象，此对象执行数据库命令(如 SQL 命令或存储过程)可以操作数据源。
- 数据读取器(data reader)对象，此对象在数据源到应用程序之间提供只前向的只读数据流。
- 数据适配器(data adapter)对象，此对象是数据集和数据源之间的桥梁。数据适配器对象使用命令对象对数据源执行 SELECT、INSERT、UPDATE 和 DELETE 命令，将任何已