VB.NET 和 SQL Server 2000 高级编程

——创建高效数据层

Tony Bain
Denise Gosnell 等著
康　博　译

清华大学出版社

需要全本请访问 www.ertongbook.com

(京)新登字 158 号

北京市版权局著作权合同登记号：01-2002-3176

内 容 简 介

Visual Basic.NET 和 SQL Server 2000 是 Microsoft 最重要的编程语言和数据库产品，它们为创建以数据为中心的应用程序提供了非常灵活的手段。本书从实际应用的角度出发，详细介绍了 Visual Basic.NET 和 SQL Server 2000 相结合的全部知识，具体内容包括 ADO.NET 类的相关知识、高效存储过程的编写和使用方法、应用程序的数据绑定、以 XML 格式操作数据、如何维护数据的完整性和安全性以及使用 Component Services 等，最后还给出一个案例，实际应用本书所介绍的理论知识。

本书着重介绍与数据有关的应用程序开发知识和技巧，非常适合于有一定的 VB 开发经验、非常熟悉 SQL Server(7.0 及更高版本)、有.NET Framework 基础知识，并熟悉 ADO 或 ADO.NET 的中高级开发人员。

Tony Bain, Denise Gosnell et al: Visual Basic.NET and SQL Server 2002: Building an Effective Data Layer
EISBN: 1861007051
Copyright© 2002 by Wrox Press Ltd.
Authorized translation from the English language edition published by Wrox Press Ltd.
All rights reserved. For sale in the People's Republic of China only.
Chinese simplified language edition published by Tsinghua University Press.

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。

图书在版编目(CIP)数据

VB.NET 和 SQL Server 2000 高级编程——创建高效数据层/(美)贝恩等著；康博译.

—北京：清华大学出版社，2002

书名原文：Visual Basic .NET and SQL Server 2000——Building an Effective Data Layer

ISBN 7-302-05999-3

Ⅰ. V... Ⅱ.①贝...②康... Ⅲ.①BASIC 语言—程序设计 ②关系数据库—数据库管理系统，SQL Server 2000 Ⅳ.TP31

中国版本图书馆 CIP 数据核字(2002)第 081724 号

出 版 者： 清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.com.cn

责任编辑： 郭东青

封面设计： 康博

版式设计： 康博

印 刷 者： 北京市清华园胶印厂

发 行 者： 新华书店总店北京发行所

开　　本： 787×1092 1/16　**印张：** 32　**字数：** 818 千字

版　　次： 2002 年 11 月第 1 版　2002 年 11 月第 1 次印刷

书　　号： ISBN 7-302-05999-3/TP・3579

印　　数： 0001～5000

定　　价： 59.00 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

随着 Microsoft .NET Framework 的推出，广大 Visual Basic 开发人员在编码方面就有了一个全新的平台。.NET 不仅在 Visual Basic 6 上完成了许多重要变化(为了确保与.NET Framework 的兼容性)，而且还通过新的 ADO.NET 数据存储方法增强 ADO 的功能。ADO.NET 提供一整套能够帮助我们检索和操作数据的类，也就是说，ADO.NET 并不仅仅是为了兼容.NET Framework 而进行的简单升级。在全面认识.NET 数据支持的基础上利用 ADO.NET 类对于开发数据驱动的应用程序是非常重要的。本书的主要目标是使开发人员能够很好地掌握 VB.NET 这门新语言，并传递您在编写“利用数据库实例中的数据”的应用程序时所需要的信息。在本书重点介绍与 SQL Server 2000 的连接时，注意其中给出的、有关处理其他数据源(例如，用于示例的 Access 2000 或 Oracle 8*i*)的连接的类的信息，也是非常重要的。这将允许您在 Windows Forms 和 Web Forms 应用程序中全面利用.NET Framework 的数据访问功能。本章各部分的主要内容如下：

第 1 章：本章简要介绍了 ADO.NET 的新特性和功能，并概述了类的结构和能力，这些都与目前的应用程序开发密切相关。尤其是，本章介绍了 System.Data 命名空间以及其中包含的主要对象。另外，本章还界定了本书其他章节所要讨论的主要内容。

第 2 章：本章开始深入探讨 ADO.NET 并简要介绍 SqlClient 命名空间。该命名空间包含了在商务对象代码中连接数据源所需的全部类。所有这些新对象，例如 DataReader、Connection、Command、DataAdapter、DataSet 和 DataTable，本章都进行了介绍。本章还描述如何使用 Command 对象连接数据库以及 DataAdapter 如何利用检索的数据链接 Command 对象。接着，本章介绍了 DataSet 和 DataTable 对象的用法，以及它们的底层结构，也就是 DataRow 和 DataColumn 对象。

最后，本章还介绍了在使用 DataRelation 检索的数据中如何维护参照完整性，以及如何在数据库连接和数据操作过程中使用这些对象和类。

第 3 章：本章介绍一个范例应用程序，它将贯穿本书所有章节。范例 HumanResources 数据库提供典型 HR 资源的一个说明，包含关于 Employees 的详细资料、雇员的支付情况、薪水详细资料以及他们个人从事的开发课题。在创建和填充数据库时，还创建一个范例 VB.NET Windows Application(Windows 应用程序)，主要是为了说明 ADO.NET 类连接和操作数据库中数据的方法。这个应用程序提供一个接口，允许您验证数据库连接是否正在工作，而且还允许更新数据库中的记录。

第 4 章：本章着重介绍 DataReader 对象和与之相关联的一些方法，并讨论了如何通过使用一个简单的 SELECT 语句和利用一个存储过程返回数据。最后，还介绍如何利用 DataReader 对象返回多个结果集、更新数据和填充控件。

第 5 章：在本章中，我们考虑到存储过程在创建高效托管代码方面的优势。本章讨论的中心是如何编写高效的存储过程，同时还全面介绍如何按照有效的方式读取、编写和更新数据。特别是，本章介绍如何通过使用 INNER JOIN、OUTER JOIN 和 CROSS JOIN 命令在多个表中

加入数据，而且在存储操作所要求的数据时使用全局和本地表。本章还介绍条件逻辑以及如何隔离“在通过存储过程检索数据时可能出现的”问题。另外，我们还介绍如何通过编写用户定义的函数来扩充存储过程的功能。我们不能低估编写拙劣的代码所带来的性能冲击，特别是当系统的大小在不断增加时；本章的目标是确保开发人员能够尽最大所能最大化数据传输的功效和来自不同数据源的数据操作。

第 6 章：前面已经讨论过如何编写能够高效执行的存储过程，本章介绍如何在 VB.NET 代码中执行它们。本章的内容构建在先前章节所积累的知识的基础之上，并说明存储过程在应用程序(使用范例 HumanResources 数据库)中的用法。本章特别介绍一些示例，分别用于通过使用 DataReader 执行存储过程、返回多个结果集、在存储过程内外传递参数、检索和传递 XML 数据。在本章最后，您完全可以编写高效的、能够以各种方式操作数据的存储过程。

第 7 章：前面已经介绍 SqlClient 命名空间以及在 ADO.NET 中用于操作 SQL Server 2000 中的数据的关键对象，而本章主要介绍数据绑定的基本内容。在.NET 领域，VB 开发人员拥有很强的数据绑定能力(同 Visual Basic 早期版本的数据绑定能力相比)，本章将说明这一数据绑定能力。我们先介绍简单和复杂数据绑定的基本内容，然后特别介绍如何使用诸如文本框和数据网格的控件绑定数据。无论是在 Windows Forms 应用程序中将数据绑定到控件，还是在 Web Forms 应用程序中将数据绑定到控件，本章都将进行介绍。

第 8 章：本章介绍数据库应用程序开发过程中的其他重要概念，即事务处理(transaction)和加锁(lock)。对于如今的分布式数据库系统和高强度的用户访问，能够同时访问保存在任何系统中的数据是极其重要的。数据维护的问题会随着用户数量的增加而出现。

在本章中，我们主要考虑事务处理的原子性、在生成事务处理时存在哪些需求以及如何实现能够高效地处理多用户的系统。对于在数据库中更新数据时可能出现的问题和如何确保所有呈现在用户面前的数据的完整性，本章都将进行详细介绍。

第 9 章：本章主要介绍.NET 领域的 Component Services 的基本内容。System.EnterpriseServices 命名空间在处理组件时后台使用 COM+，且使“处理 DLL 文件”的过程比早期的 ADO 要彻底。本章讨论了服务组件的基本内容及其在分布式事务处理中的用法。接着，本章说明一个简单程序集的创建过程和如何使用 Sn 和 RegSvcs 工具注册该程序集。最后，本章创建一个客户组件，该组件能够利用注册的程序集并返回一些说明组件的用法的值。

第 10 章：本章讨论 SQL Server 2000 对 XML 的支持程度和支持的基本内容。首先详细地介绍了 FOR XML EXPLICIT 子句和返回 XML 结果集(该结果集将来适合于在自己的数据应用程序中操作)所要求的数据结构。另一种可选方法是使用 XSL 样式表格式化在 SQL Server 中检索的数据以传输数据，在介绍该方法之前，先概述上面所述子句的缺点。最后，本章介绍如何使用可用于 SQL Server 2000 和.NET Framework 的 SQLXML 功能从头开始创建一个作为 Web 服务的存储过程的示例。

第 11 章：本章介绍一个检验表的窗体，并涉及到一些与 DBA 的角色相关联的传统主题。在此，我们将讨论如何通过授予用户注册权限来限制对 SQL Server 的访问。本章还讨论 SQL Server 注册的基本内容以及 Windows Groups 和 Windows Users，且将重点放在如何在 SQL Server Enterprise Manager(SQL Server 企业管理器)中操作它们和自动使用 SQL-DMO (Distributed Management Objects，分布式管理对象)对象两个方面内容。然后，本章开始讨论 SQL Server 的

需要全本请访问 www.ertongbook.com

性能以及如何通过 Profiler 工具来监控这个性能。索引是数据库性能的一个关键体现者，进而本章也对 Index Tuning Wizard(索引调整向导)进行讨论。本章还介绍如何移动和恢复数据库、生成“能够用于创建数据库的”脚本的方法以及包含在它们中的对象。

第 12 章：在本章中，我们以一个案例分析结束全书，该案例分析的目的是说明本书前面所讨论的许多概念。本章讨论一个银行业务应用程序，该程序允许用户使用各种账户登录银行业务事务处理系统。另外，该应用程序还汇总所实施的借贷，以便提供有问题账目的平衡。这个应用程序被分解为传统的三层应用程序，其中数据层为数据库，中间层为一些代码模块(商务对象)，表示层为 Web Form 用户接口。

本书读者对象

本书适合那些有能力编写 VB 中间层代码，但又想学习如何编写以数据为中心的代码的编程人员。本书向您提供了许多快速处理新 ADO.NET 类和对象的方法。到本书的结尾处，您将全面熟悉如何从 SQL Server 检索数据并按照输出所要求的结果操作数据。您不仅可以轻松开发“以各种格式使用数据的”应用程序，而且应用程序可以保证那些数据的安全性和完整性。总之，本书为利用 Visual Basic .NET 编写高性能的、安全的数据应用程序，提供了一个蓝图。

您应当是一位中级水平的 VB.NET 开发人员，而且对整个.NET Framework 的基础知识有一个很好的了解，并有一些 ADO 或 ADO.NET(也可能是两者兼有)开发经验。另外，本书还假设您熟悉简单的 T-SQL 语句，且在 SQL Server (版本 7.0 和更高版本)中使用过存储过程。

学习本书所需要的准备工作

为了很好地学习本书内容，您需要 Visual Studio.NET 的 Enterprise 或 Professional 版本的完整产品许可，而且有权使用 SQL Server 2000 (Standard、Development 或 Enterprise 版本)的一个实例。

用户支持

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。

如何下载本书的范例代码

当您访问 Wrox 公司站点(地址为 http://www.wrox.com/)时，通过 Search 工具或书名列表，可以方便地定位需要的书目。然后，单击 Code 列中的 Download 超链接，或者单击本书的详细信息页面中的 Download Code 超链接，就可以下载相应的范例代码。

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。把附件保存到本地磁盘上的文件夹中后，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。在解压缩的过程中，应确保解压缩程序被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，请反馈信息给我们，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 工具或者书名列表轻松定位本书页面。然后，单击 Book Errata 超链接即可，该链接位于本书的详细页面中的封面图解下面。

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。一个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。对于一些被频繁提到的问题将被归档，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详细技术问题。
- 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/ forum 中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，特将编程人员加入到 P2P 站点的邮件列表中，而且我们惟一的系统将 programmer to programmer™(由程序员为程序员而作)的编程理念与邮件列表、论坛、新闻组以及所有其他服务内容(一对一的邮件支持系统除外)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所检查到。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(对于本书为 Databases)。

(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。

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 ····· 1
1.1　通用数据访问和 ADO 历史 ····· 1
1.2　ADO.NET 代表数据访问的新时代 ····· 2
1.2.1　DataReader ····· 3
1.2.2　DataSet ····· 3
1.2.3　使用 DataSet 代替 DataReader 的时机 ····· 5
1.2.4　托管提供程序 ····· 6
1.2.5　用于数据访问的命名空间 ····· 8
1.3　小结 ····· 15

第 2 章　SqlClient 命名空间 ····· 16
2.1　SqlConnection 对象 ····· 16
2.1.1　属性 ····· 17
2.1.2　方法 ····· 22
2.1.3　入池连接 ····· 22
2.2　SqlCommand 对象 ····· 25
2.2.1　属性 ····· 25
2.2.2　方法 ····· 27
2.3　SqlDataReader 对象 ····· 31
2.3.1　属性 ····· 32
2.3.2　方法 ····· 32
2.4　SqlDataAdapter 对象 ····· 33
2.4.1　属性 ····· 35
2.4.2　方法 ····· 36
2.4.3　更新数据 ····· 38
2.5　DataSet 对象 ····· 42
2.5.1　属性 ····· 42
2.5.2　方法 ····· 43
2.6　DataTable 对象 ····· 46
2.6.1　属性 ····· 47
2.6.2　方法 ····· 48
2.7　DataRow 对象 ····· 52
2.7.1　属性 ····· 53
2.7.2　方法 ····· 54

2.7.3 RowVersion ····· 55
2.8 DataColumn 对象 ····· 58
2.9 DataRelation ····· 64
2.10 小结 ····· 72

第 3 章 HR 应用程序范例 ····· 73
3.1 HumanResources 数据库 ····· 73
3.2 HumanResources 项目 ····· 83
3.2.1 创建 EmployeeDetails 窗体 ····· 83
3.2.2 EmployeeDetails 代码 ····· 86
3.2.3 调试 ····· 103
3.2.4 测试 HumanResources 应用程序 ····· 106
3.3 小结 ····· 107

第 4 章 用 ADO.NET 处理数据流 ····· 108
4.1 概述 ····· 108
4.2 使用 DataReader 检索数据 ····· 109
4.2.1 使用 SELECT 语句检索数据 ····· 110
4.2.2 GET 方法 ····· 113
4.2.3 检索多个结果集 ····· 116
4.2.4 通过执行存储过程检索数据 ····· 118
4.3 使用 DataReader 更新数据 ····· 120
4.4 使用 DataReader 填充控件 ····· 121
4.5 小结 ····· 125

第 5 章 编写优秀的存储过程 ····· 126
5.1 存储过程和性能 ····· 126
5.2 安全性 ····· 127
5.3 执行存储过程 ····· 128
5.3.1 动态 SQL ····· 129
5.3.2 SET NOCOUNT ····· 130
5.4 存储过程中的参数 ····· 131
5.5 SELECT *语句 ····· 132
5.6 延期的名称解析方案 ····· 133
5.7 连接 ····· 134
5.7.1 内部连接 ····· 135
5.7.2 外部连接 ····· 137
5.7.3 交叉连接 ····· 140
5.8 临时表 ····· 141
5.9 游标 ····· 144

5.10　内联决策 …… 147
5.11　流程控制 …… 151
5.11.1　IF…ELSE …… 151
5.11.2　WHILE…BREAK…CONTINUE …… 153
5.11.3　RETURN …… 154
5.12　传递结果集 …… 155
5.13　导航树 …… 162
5.14　嵌套过程 …… 165
5.15　用户定义函数 …… 166
5.15.1　标量 UDF …… 167
5.15.2　内联 UDF …… 168
5.15.3　表格式 UDF …… 168
5.16　全局变量 …… 169
5.16.1　@@ERROR …… 170
5.16.2　@@ROWCOUNT …… 170
5.16.3　@@SERVERNAME …… 171
5.16.4　@@IDENTITY …… 171
5.17　处理错误 …… 172
5.17.1　RAISERROR …… 174
5.17.2　Ad Hoc 错误消息 …… 176
5.17.3　用户定义错误消息 …… 176
5.17.4　处理引发的错误 …… 177
5.18　小结 …… 178

第 6 章　在 VB.NET 中使用存储过程 …… 179
6.1　在 VS.NET 中创建存储过程 …… 179
6.2　DataReader …… 182
6.3　多个结果集 …… 184
6.4　传递参数 …… 188
6.4.1　SqlParameter 对象 …… 189
6.4.2　SqlDataReader 和参数 …… 190
6.5　返回参数 …… 197
6.6　检索 XML …… 198
6.7　以 XML 传递数据 …… 200
6.8　SQLXML …… 203
6.9　小结 …… 212

第 7 章　数据绑定 …… 213
7.1　概述 …… 213
7.2　用于 Windows Forms 的数据绑定 …… 214

完整全本请访问 www.ertongbook.com

7.2.1 简单数据绑定 ····· 214
7.2.2 复杂数据绑定 ····· 220
7.3 用于 Web Forms 的数据绑定 ····· 229
7.3.1 绑定到 DataGrid ····· 229
7.3.2 使用 DataGrid 更新数据 ····· 232
7.4 小结 ····· 240

第 8 章 事务和封锁 ····· **241**
8.1 ACID 测试 ····· 241
8.1.1 原子性(A) ····· 241
8.1.2 一致性(C) ····· 242
8.1.3 隔离(I) ····· 242
8.1.4 持久性(D) ····· 242
8.2 隐式事务和显式事务 ····· 242
8.3 使用事务 ····· 242
8.3.1 使用 SqlTransaction 还是 T-SQL ····· 243
8.3.2 BEGIN TRANSACTION 命令 ····· 243
8.3.3 ROLLBACK TRANSACTION 命令 ····· 245
8.3.4 COMMIT TRANSACTION 命令 ····· 247
8.3.5 SAVE TRANSACTION 命令 ····· 248
8.3.6 事务范围 ····· 252
8.4 事务内的命令 ····· 253
8.5 事务和锁定 ····· 258
8.5.1 锁定的作用 ····· 258
8.5.2 设置锁定 ····· 258
8.5.3 锁定粒度 ····· 260
8.5.4 隔离级别 ····· 261
8.5.5 锁定提示 ····· 273
8.5.6 显示锁定信息 ····· 275
8.6 最好的事务惯例 ····· 276
8.6.1 仅在必需时采用事务 ····· 276
8.6.2 精简事务 ····· 277
8.6.3 使用最低级别的隔离 ····· 277
8.6.4 建立可以确保成功操作的事务 ····· 277
8.6.5 避免死锁 ····· 280
8.7 小结 ····· 283

第 9 章 组件服务 ····· **284**
9.1 已服务组件 ····· 284

9.2 建立已服务组件 …… 285
9.2.1 创建已服务组件 …… 285
9.2.2 创建客户端组件 …… 295
9.3 对象池和其他 COM+特性 …… 299
9.4 小结 …… 301

第 10 章 SQL Server 和 XML …… 302
10.1 FOR XML EXPLICIT …… 302
10.1.1 结果集结构 …… 304
10.1.2 排序 …… 306
10.1.3 属性和元素值 …… 309
10.1.4 填充结果集 …… 312
10.2 使用样式进行转换 …… 323
10.3 作为 Web 服务的存储过程 …… 336
10.3.1 配置存储过程 Web 服务 …… 337
10.3.2 引用 Web 服务 …… 344
10.3.3 使用 Web 服务 …… 345
10.4 小结 …… 347

第 11 章 开发人员的有效管理 …… 349
11.1 安全管理 …… 349
11.1.1 SQL Server 登录 …… 349
11.1.2 数据库权限 …… 363
11.2 监控 SQL Server …… 373
11.2.1 SQL Server Profiler …… 373
11.2.2 配置长期运行的查询 …… 376
11.2.3 后台记录 …… 381
11.2.4 Index Tuning Wizard …… 384
11.3 移动数据库 …… 387
11.3.1 Copy Database Wizard …… 388
11.3.2 分割/重组 …… 389
11.3.3 复原到新服务器 …… 390
11.4 生成脚本 …… 391
11.5 小结 …… 394

第 12 章 案例分析：IBankAdventure …… 395
12.1 概述 …… 395
12.2 IBankAdventure 数据库 …… 396
12.2.1 创建数据库 …… 396

12.2.2 数据库表 ……396
12.2.3 链接我们的数据 ……400
12.2.4 存储过程 ……402
12.2.5 数据库的安全策略 ……408
12.3 IBankAdventure 商务对象 ……409
12.3.1 DataAccess 商务对象 ……409
12.3.2 Bank 商务对象 ……413
12.3.3 BillPay 商务对象 ……420
12.3.4 Security 商务对象 ……424
12.4 IBankAdventure 用户接口 ……426
12.4.1 保护我们的 Web 站点 ……426
12.4.2 创建 IBank 项目 ……427
12.5 需要改进的地方 ……465
12.6 小结 ……465

附录 A ADO.NET 对象模型 ……466
A.1 Command 类 ……466
A.2 Connection 类 ……467
A.3 DataAdapter 类 ……469
A.4 DataColumn 类 ……471
A.5 DataColumnMapping 类 ……472
A.6 DataReader 类 ……473
A.7 DataRelation 类 ……476
A.8 DataRow 类 ……477
A.9 DataSet 类 ……479
A.10 DataTable 类 ……482
A.11 DataTableMapping 类 ……484
A.12 DataView 类 ……485
A.13 DataViewManager 类 ……487
A.14 DataViewSetting 类 ……488
A.15 OleDbError 类 ……489
A.16 OleDbException 类 ……489
A.17 Parameter 类 ……490
A.18 SqlError 类 ……492
A.19 SqlException 类 ……493
A.20 Transaction 类 ……494

第1章 ADO.NET

纵观数据访问方法—— ADO 和 ADO.NET 的过去和现在，我们即将进入 Visual Basic .NET 数据库应用开发的时代。本书将详细讲述设计 VB.NET 数据库应用程序的方法，并提供了大量的源代码示例演示如何实现我们所介绍的概念。不过，首先简要介绍 ADO.NET 在访问数据(我们的应用程序中所需的数据)方面所起的作用，是很重要的。本章的主要内容如下：

- ADO 简史
- ADO.NET 简介
- DataSets 和 DataReaders 简介
- 通过托管提供程序(Managed Provider)从数据存储检索数据
- System.Data 命名空间概要
- System.Data.SqlClient 和 System.Data.OleDb 命名空间概要

下来开始概述 ADO。

1.1 通用数据访问和 ADO 历史

在详细讲述 ADO.NET 之前，我们将花点时间简要介绍通用数据访问和 Actirex 数据对象(ADO)的历史。ADO.NET 是由 ADO 发展而来的。ADO 替换了数据访问对象(DAO)和远程数据对象(RDO)数据访问方法，并提供功能更强大的检索和管理数据的方法。DAO 和 RDO 设计用于客户服务器应用程序并且效果不错。

不过，由于具有多层应用程序和多种数据存储的分布式计算变得越来越普遍，就需要更为成熟的数据访问方法通过多个平台与多种数据源进行通信。OLE DB 旨在解决这个问题，而 ADO 是通过应用程序访问 OLE DB 的方法。OLE DB 是 Microsoft 的进行通用数据访问的访问方法，而不管数据是否驻留在关系数据库、文件系统、电子邮件服务器等中。允许采用有意义的方法快速从关系或非关系数据库收集数据，OLE DB 就可以将开放数据库连接(ODBC)提高到另一个水平。ODBC 是功能很强大的数据库 API，您几乎可以使用它访问任何关系数据库。ODBC 也提供对非关系数据库的支持，但是由于关系结构都必须创建在非关系结构的顶部，所以通常都很慢。另一方面，ADO 也可以迅速高效地处理文本文件、关系数据库和多种关系和非关系数据源。除了可以访问 OLE DB 数据存储外，ADO 也提供对 ODBC 的支持，由于系统中广泛地使用 ODBC。

ADO 对象模型包含下列对象：

- Connection
- Command
- Parameter

- Recordset
- Field
- Property
- Error

ADO 对象模型包含下列集合：

- Fields
- Properties
- Parameters
- Errors

Recordset 是 ADO 对象模型的核心，它允许采用与单一表类似的方式检索并存储数据。Field 和 Property 对象与存储 Recordset 信息有关，Fields 和 Properties 集合也是如此。例如，Recordset 包含 Field 对象，该对象由“用于填充它的”数据元素组成。由于在指定的 Recordset 中可能有一个或多个字段，Fields 集合可以存储多条记录。Property 对象包含描述 Recordset 的信息，它有一个可以存储多个值的 Properties 集合。Error 对象和 Errors 集合也是如此。您可以使用 Connection 对象建立与数据源的连接，使用 Command 对象发送数据源命令。还可以使用 Parameter 对象和 Parameters 集合将参数传递给在基本数据存储上执行的命令(如存储过程参数)。

ADO 非常可靠，它已成为 Visual Studio 应用程序访问数据的默认标准方法。虽然 ADO 比过去的其他数据访问方法要优越，但是它仍然存在一些不足：

- ADO Recordset 对象看起来像单一表。
- Recordset 不支持表间关系。当需要从多个数据源检索数据时，Recordset 要求您将它们汇编到单一结果集表中，方法是指定 recordset 的来源为单一 SQL 语句。该语句将多个表联结到一起形成单一结果。
- Recordset 主要在连接模式下运行(虽然它修改后也可以用于断开连接模式)。这就会导致多个数据库锁定和活动连接，从而影响应用程序的性能和可伸缩性。
- COM 编组通过线路(wire)传输断开连接的 Recordset。只使用受到支持的 COM 数据类型，由于必须进行类型转换，所以这样就会消耗系统资源。
- 穿过防火墙进行通信比较困难，这是由于多数防火墙都配置用于防止系统级请求，如 COM 编组。
- 通过防火墙进行通信时，XML 是非常有价值的，但是 Recordset 本身并不支持 XML。ADO 可以使用有限的 XML 支持的意思就是，例如，不能将任意 XML 文档载入 Recordset。

既然我们对 ADO 的特性和局限性已有了基本了解，接下来就讨论一下 ADO.NET 并看看它与 ADO 相比有什么改进。

1.2 ADO.NET 代表数据访问的新时代

ADO.NET 是一种新的数据访问策略，它不只是 ADO 的改进版本。在许多方面它都采用了新的思维方式，它涉及到我们通常使用“与数据源断开连接的”数据的领域。这种思想就是，在高度互联的世界中，应用程序中的数据可以有多个来源(例如，印地安那州的电子邮件服务器、

佛罗里达的数据库或伦敦的文件服务器)以及多种格式。也就是说，您希望快速取回数据，这样就可以在本地使用这些数据，而不需要保持到数据存储的连接。稍后就可以将任何修改传递到基数据存储(方法也是建立快速连接来进行修改操作)。断开连接的数据的本质与在使用时大多数 Recordset 都会保持到基本数据存储的连接的 ADO 有点不同。ADO.NET 提供对 XML 的内在支持，可以使数据传输通过防火墙的过程更容易。由于旧版本的原因需要使用 ADO 时，您仍然可以访问 ADO 库。

ADO.NET 提供了两种访问数据的基本方法：通过 DataReader 和 DataSet。 DataReader 是只向前读的数据流，仅仅用于对数据进行快速有效的访问。DataSet 就是人们听到 ADO.NET 这个术语时通常想到的，这是由于 DataSet 利用几个新特性替换了 Recordset 概念。接下来，我们将逐一讨论这些访问数据的方法，首先讨论 DataReader。

1.2.1　DataReader

DataReader 在连接模式下运行，检索只向前的、只读的、且位于数据存储中的数据副本。访问 DataReader 时保持到数据库的连接处于打开状态，结束访问时必须关闭连接。它设计用于快速数据访问，在这个过程中，您连接数据库并检索信息(例如，然后通过 Web 浏览器或 Windows 窗体显示)，然后关闭数据库连接。DataReader 不支持回滚和前滚或修改操作。它与 ADO 只向前的 Recordset 最接近。

下面描述 DataSet 的差别，然后讨论在什么情况下使用哪个更好。

1.2.2　DataSet

DataSet 是 ADO.NET 的核心，是数据在内存中的副本。与 ADO Recordset 不同的是，它有多个表，每个表都来自不同的数据存储。可以通过关系将 DataSet 中的表彼此相连。简而言之，DataSet 只是内存数据库，它没有与填充它的基本数据存储相连。

让我们分析一下 DataSet 的工作原理图，如图 1-1 所示。

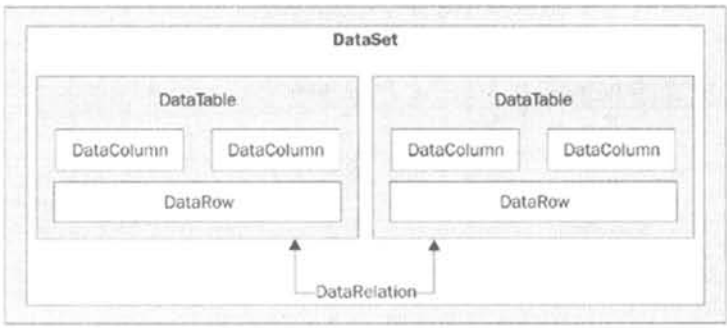


图　1-1

如图 1-1 所示，DataSet 可以包含一个或多个 DataTable，利用数据存储的信息对它们进行填充。DataSet 中的每个 DataTable 都有一个或多个 DataColumn，它们定义保存哪些数据列。它也有一个或多个 DataRow，它们是每个存储的记录的实际值。使用 DataRelation 描述 DataSet 中的表之间的关系(如图 1-1 所示)，您可以在 DataSet 中创建数据的 DataView，它可以创建单

需要全本请访问 www.ertongbook.com