

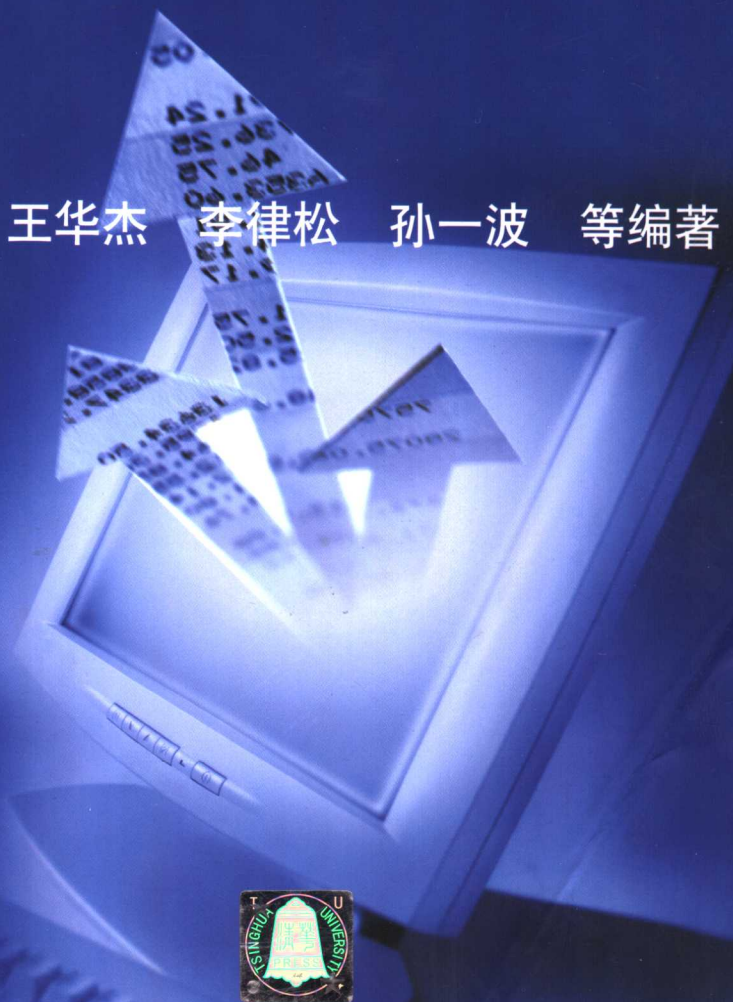
精通

C#

数据库开发

王华杰 李律松 孙一波 等编著

- ✓ 数据库访问基础ADO.NET
- ✓ WinForm个人日程管理
- ✓ 学生信息报表系统的实现
- ✓ 电子相册的设计与实现
- ✓ 网上选课系统
- ✓ ASP.NET实现论坛
- ✓ ASP.NET实现网上商城
- ✓ 电子图书在线出版系统



清华大学出版社

精通 C#数据库开发

王华杰 李律松 孙一波 等编著

清华大学出版社

北 京

内 容 简 介

本书通过大量的典型实例,循序渐进地介绍了使用 Visual C# .NET 2003 开发数据库应用程序的方法和技巧。全书共 8 章,第 1 章介绍了 .NET 平台数据库编程与 ADO.NET 的基础知识,第 2~8 章通过 7 个具体案例的设计与实现,详细地介绍了运用 C# 语言进行数据库编程的方法。

本书选材新颖,案例由浅到深,内容完整,涵盖了 Windows 窗体的数据库编程和 Web 窗体的数据库编程,以及 XML Web Service 等高级应用,是一本易懂、实用、极具参考价值的实例教程。本书主要面向有一定 C# 语言编程基础、准备学习 C# 数据库编程或者想提高 C# 编程技巧的读者。

本书所附光盘中包含本书所有实例的源代码,读者在实践时可直接使用。

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

精通 C# 数据库开发/王华杰,李律松,孙一波等编著.—北京:清华大学出版社,2004

ISBN 7-302-08424-6

I. 精… II. ①王…②李…③孙… III. C 语言程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 028023 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:孟毅新

文稿编辑:许书明

封面设计:久久度企划

版式设计:康 博

印 刷 者:北京通州大中印刷厂

装 订 者:三河市化甲屯小学装订二厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:27.25 字数:646 千字

版 次:2004 年 4 月第 1 版 2004 年 4 月第 1 次印刷

书 号:ISBN 7-302-08424-6/TP·6057

印 数:1~4000

定 价:48.00 元(含光盘)

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704。

前 言

近年来,随着.NET平台的崛起,C#语言作为此平台上的主流语言,在编程中的应用越来越广泛。C#从C和C++演变而来,是一种简单易懂、面向对象、类型安全的编程语言。它结合了 Visual Basic 编程的高效率和 C++的强大功能和灵活性,使用方便、语法优雅,受到了广大程序员的好评和喜爱。同时,随着应用系统越来越复杂,需要处理的数据越来越多,数据库的使用也相应的广泛起来。任何稍有规模的应用几乎都离不开后台数据库的支持。如何有效的编写数据库应用程序显得非常重要了。

本书从程序开发的角度,深入介绍了使用 C#语言开发数据库应用程序的基本知识、技术特性和实用技巧,并将 C#语言的特性和数据库编程的技术难点融入到具体的实例中,使读者更容易掌握使用 C#语言开发数据库应用程序的方法和技巧。由于在书中还引入了 MVC、XML Web Services 等概念,在增强读者编程经验的同时,也利于对这些概念的掌握和理解。

本书使用了大量示例代码以便读者理解和实际操作,特别针对数据库的操作进行了非常详细的讲解,对于刚刚迈入 C#语言大门和刚刚接触数据库编程的读者来说,本书无疑是一本理想的学习教材和参考用书,建议将本书与 C#语言的基础教材结合使用,效果更佳。

本书共分为 8 章。为了方便读者阅读本书,这里先简单介绍一下本书的组织结构和主要内容。

第 1 章介绍数据库访问基础——ADO.NET 的基本内容,包括 ADO.NET 设计目标、结构、核心组件以及简单示例。

第 2 章通过介绍个人日程管理程序,具体说明使用 Visual C# .NET 开发 WinForm 程序的具体过程,以及在 WinForm 程序中调用数据库的方法。

第 3 章是数据库编程的非常典型的应用,基于一个比较简单的学生信息报表系统来介绍 VS.NET 中水晶报表的使用方法。

第 4 章介绍 Java 世界中最为流行、基本的一种设计模式——MVC 结构,它适合大型的 GUI 管理,我们把这种思想在 C#中实现。本章介绍的实例是电子相册的设计与实现,它的功能简单,结构清晰。

第 5 章介绍了 XML,它是整个.NET平台的基石,而它又与数据库编程联系非常紧密,本章通过一个网上选课系统的设计和实现,重点讲述了使用 C#语言处理 XML 数据的方法。

第 6 章通过 ASP.NET 实现论坛,介绍了一个简单的网上论坛的设计和实现的过程,主要使用到 ASP.NET 技术和 ADO.NET 数据库访问技术。

第 7 章通过 ASP.NET 实现网上商城,特别对数据库中使用到的存储过程的设计和实现进行了具体的讲解。

第 8 章为 ASP.NET 高级应用,用 C#编写多层的分布式应用——电子图书在线出版系统,它涵盖了 XML Web Service 的重要内容,程序员往往拿它同 Sun 的 J2EE 比较。本章

旨在通过介绍这个系统的设计和实现,讲解使用.NET 平台开发 XML Web Service 以及 N 层结构的分布式应用程序的方法。

每一章大体都分为如下几个部分。

- 案例简介。从需求的角度简单介绍了所选实例的功能,和我们通过本章的讲解要达到的目的。
- 应用程序概述。站在程序员的角度对应用程序进行功能模块的划分。
- 应用程序方案设计。对应用程序本身和数据库进行详细地设计。对应用程序来说具体到了某个类的设计,对数据库来说,具体到每一张表和每一个存储过程的设计。
- 方案实现。详细介绍系统的实现,包括界面的实现和关键代码的分析。
- 小结。对本章的主要内容、关键技术以及所要注意的问题进行总结。

本书由李律松、孙一波和王华杰共同执笔编写。此外,蓝荣香、王昊亮、喻波、马天一、魏勇、郝荣福、李光龙、孙明、李大宇、武思宇、牟博超、李冰、付鹏程、高翔、朱丽云、崔凌、张巧玲、李辉、李欣、柏宇、郭强、金春范、程梅、黄霆、钟华、高海峰、王建胜、张浩、刘湘和邵蕴秋等同志在整理材料方面给予了编者很大的帮助。在此,编者对他们表示衷心的感谢。向以上人员致以诚挚的感谢。

由于编者水平有限,且编写时间仓卒,书中难免有疏漏和不足之处,恳请广大读者批评指正。

编 者

2003 年 12 月

目 录

第 1 章 数据库访问基础 ADO.NET	1
1.1 ADO.NET 概述	1
1.1.1 ADO.NET 设计目标	1
1.1.2 ADO.NET 结构	2
1.1.3 ADO.NET 示例应用程序	4
1.2 .NET 数据提供程序	5
1.2.1 使用 ADO 连接到数据源	6
1.2.2 Command 命令	11
1.2.3 使用 DataReader 检索数据	15
1.2.4 使用 DataAdapter	18
1.3 创建和使用 DataSet	27
1.3.1 创建 DataSet	28
1.3.2 向 DataSet 添加 DataTable	28
1.3.3 添加表间关系 DataRelation	28
1.3.4 导航表间关系	29
1.3.5 DataSet 同数据源中的数据一起使用	32
1.3.6 合并 DataSet 内容	32
1.3.7 复制 DataSet 内容	34
1.3.8 使用 DataSet 事件	35
1.3.9 使用类型化的 DataSet	35
1.4 DataSet 和 XML	41
1.4.1 DiffGram	41
1.4.2 从 XML 中加载 DataSet	43
1.4.3 编写 DataSet 的 XML 数据形式	45
1.5 创建和使用 DataTable	46
1.5.1 创建数据表	47
1.5.2 定义数据表的架构	48
1.5.3 在数据表中操作数据	52
1.6 创建和使用 DataView	59
1.6.1 创建 DataView	59
1.6.2 使用 DataView 对数据排序和筛选	60
1.6.3 使用 DataView 查看数据	61

1.6.4 使用 DataView 修改数据	63
1.6.5 使用 DataView 事件	65
1.6.6 使用 DataViewManager 设置默认表视图	65
1.7 小结	67
第 2 章 WinForm 实现个人日程管理	68
2.1 案例简介	68
2.2 应用程序概述	68
2.3 方案设计	70
2.3.1 应用程序前端的设计	70
2.3.2 后端数据库表的设计	70
2.3.3 后端数据库表间逻辑	73
2.3.4 后端数据库存储过程的设计	73
2.4 方案实现	77
2.4.1 数据库访问公用类	77
2.4.2 日程浏览模块的实现	86
2.4.3 日程管理模块的实现	90
2.4.4 数据库操作	97
2.5 小结	100
第 3 章 学生信息报表系统的实现	102
3.1 案例简介	102
3.2 水晶报表简介	103
3.3 应用程序概述	104
3.4 方案设计	105
3.4.1 应用程序的结构	105
3.4.2 数据库设计	105
3.5 方案实现	107
3.5.1 添加数据集文件	107
3.5.2 使用 Crystal 报表专家生成报表	110
3.5.3 使用 Crystal 报表设计器	116
3.5.4 实现程序关键代码	119
3.5.5 运行结果	124
3.6 小结	127
第 4 章 电子相册的设计与实现	128
4.1 案例简介	128
4.2 MVC 简介	129
4.3 应用程序概述	130

4.4	方案设计	131
4.4.1	应用程序的结构	132
4.4.2	数据库设计	132
4.5	方案实现	134
4.5.1	Observer 类的实现	134
4.5.2	Treeltem 类的实现	135
4.5.3	Model 的实现	135
4.5.4	View 的实现	142
4.5.5	Controller 的实现	151
4.6	小结	157
第 5 章	网上选课系统	158
5.1	案例简介	158
5.2	应用程序概述	158
5.3	关键技术概述	159
5.3.1	XML 文档对象模型	159
5.3.2	遍历 DataSet	160
5.3.3	TreeView 控件动态绑定数据	160
5.4	方案设计	161
5.4.1	客户端的设计	161
5.4.2	数据库表的设计	162
5.4.3	数据库存储过程的设计	166
5.5	方案实现	176
5.5.1	公共类的实现	176
5.5.2	学生子系统的实现	176
5.5.3	管理员子系统的实现	193
5.6	小结	218
第 6 章	ASP.NET 实现论坛	220
6.1	案例简介	220
6.2	应用程序概述	221
6.3	方案设计	221
6.3.1	论坛的结构设计	221
6.3.2	论坛的数据库设计	223
6.3.3	存储过程设计	227
6.4	方案实现	231
6.4.1	数据库访问类	231
6.4.2	用户自定义控件	237

6.4.3 进入论坛	240
6.4.4 帖子管理	243
6.4.5 论坛管理	255
6.4.6 用户管理	260
6.4.7 出错处理	268
6.5 服务器部署	269
6.6 小结	271
第7章 ASP.NET 实现网上商城	272
7.1 案例简介	272
7.2 应用程序概述	272
7.3 方案设计	273
7.3.1 商城前台的方案设计	273
7.3.2 数据库表的定义	274
7.3.3 数据库表间逻辑	278
7.3.4 数据库存储过程的设计	279
7.4 方案实现	291
7.4.1 数据信息类	292
7.4.2 数据库访问类	293
7.4.3 用户自定义控件	311
7.4.4 进入商城	315
7.4.5 获取商品列表	318
7.4.6 查看商品细节	322
7.4.7 查询商品	323
7.4.8 购物车管理	325
7.4.9 订单管理	330
7.4.10 用户管理	334
7.5 服务器部署	336
7.6 小结	338
第8章 电子图书在线出版系统	339
8.1 案例简介	339
8.2 Web Service 概述	340
8.3 应用程序概述	340
8.4 应用程序方案设计	341
8.4.1 书稿撰写人服务和前端网站的设计	342
8.4.2 后端数据库表的设计	342
8.4.3 后端数据库存储过程的设计	346

8.5 方案实现	357
8.5.1 书稿撰写人服务的实现	360
8.5.2 前端网站的实现	386
8.6 服务器配置	420
8.7 小结	422

第1章 数据库访问基础ADO.NET

ADO.NET 提供对 Microsoft SQL Server 等数据源以及通过 OLE DB 和 XML 公开的数据源的一致访问。数据共享使用者应用程序可以使用 ADO.NET 来连接到这些数据源,并检索、操作和更新数据。

ADO.NET 包含用于连接到数据库、执行命令和检索结果的 .NET 数据提供程序,用户可以直接处理检索到的结果,也可以将其放入 DataSet 对象中。使用 DataSet 对象方便与来自多个源的数据或在层之间进行远程处理的数据组合在一起,以特殊方式向用户公开,它也可以独立于 .NET 数据提供程序使用,用于管理应用程序本地的数据或源自 XML 的数据。

1.1 ADO.NET 概述

ADO.NET 是微软 .NET 框架中的一种新的数据访问技术,它能用于访问关系型数据库系统,如 SQL Server 2000,及很多其他已经配备了 OLE DB 供应器的数据源。以前,数据处理主要依赖于基于连接的双层模型。当数据处理越来越多地使用多层结构时,程序员正逐渐向断开方式转换,以便为应用程序提供更佳的可缩放性。ADO.NET 技术顺应了这种需求,虽然在某种程度上,ADO.NET 代表了最新版本的 ADO 技术,然而,ADO.NET 引入了一些重大变化和革新,它专门用于结构松散的、本质非链接的 Web 应用程序。

1.1.1 ADO.NET 设计目标

随着应用程序开发的发展演变,新的应用程序已基于 Web 应用程序模型越来越松散地耦合。如今,越来越多的应用程序使用 XML 来编码要通过网络连接传递的数据。Web 应用程序将 HTTP 用作在层间进行通讯的结构,因此它们必须显式地处理请求之间的状态维护。这一新模型大大不同于连接、紧耦合的编程风格,此风格曾是客户端/服务器时代的标志。在此编程风格中,连接会在程序的整个生存期中保持打开,而不需要对状态进行特殊处理。

因此,微软认识到需要为数据访问提供全新的编程模型,此模型是基于 .NET 框架生成的。基于 .NET 框架这一点将确保数据访问技术的一致性——组件将共享通用的类型系统、设计模式和命名约定。

设计 ADO.NET 的目的是为了满足这一新编程模型的以下需要:

- 断开式数据结构

- 与 XML 的紧密集成
- 能够组合来自多个、不同数据源的数据的通用数据表示形式
- 与数据库交互而优化的功能

微软在创建 ADO.NET 时, 具有以下设计目标。

(1) 尽可能同 ADO 保持一致

ADO.NET 的设计满足了当今应用程序开发模型的多种要求。同时, 该编程模型尽可能地与 ADO 保持一致, 这使当今的 ADO 开发人员不必从头开始学习全新的数据访问技术。ADO.NET 是 .NET 框架的固有部分, 因此对于 ADO 程序员不是完全陌生的。

大多数基于 .NET 的新应用程序将使用 ADO.NET 来编写, 但是 .NET 程序员仍然可以通过 .NET COM 互操作性服务来使用 ADO, 使得 ADO.NET 与 ADO 共存。

(2) 支持 N 层编程模式

ADO.NET 为断开式 N 层编程环境提供了高级的支持, 许多新的应用程序都是为该环境编写的。使用断开式数据集这一概念已成为编程模型中的焦点。N 层编程的 ADO.NET 解决方案就是使用 ADO.NET 的 DataSet 组件。

(3) 内置 XML 支持

.NET 框架不仅支持 Web 标准——它还是完全基于 Web 标准生成的, 作为 .NET 框架一部分, ADO.NET 内置 XML 支持, 而且内置在非常基本的级别上。XML 的特点是: XML 和数据访问紧密联系在一起, 即 XML 的全部内容都是有关数据编码的, 因此数据访问越来越多的内容都与 XML 有关。

1.1.2 ADO.NET 结构

ADO.NET 提供两个核心组件: DataSet 和 .NET 数据提供程序(包括 Connection、Command、DataReader 和 DataAdapter 4 个对象), 设计 ADO.NET 组件的主要目的是为了从数据操作中分解出数据访问。

1. ADO.NET DataSet 组件

ADO.NET DataSet 组件为 ADO.NET 提供断开式结构服务, DataSet 的设计是为了实现独立于任何数据源的数据访问。因此, 它可以用于多种不同的数据源, 用于 XML 数据, 或用于管理应用程序本地的数据。DataSet 包含一个或多个 DataTable 对象的集合, 这些对象由数据行和数据列以及主键、外键、约束和有关 DataTable 对象中数据的关系信息组成。

实际上, ADO.NET 正是借用 XML 的力量来提供对数据的断开式访问。ADO.NET 的设计与 .NET 框架中 XML 类的设计都是同一个结构的组件——DataSet 对象。无论 DataSet 是文件还是 XML 流, 它都可以使用来自 XML 源的数据来进行填充。无论 DataSet 中数据的数据源是什么, DataSet 都可以写为符合 WWW 联合会 (W3C) 的 XML, 并且将其架构包含为 XML 架构定义语言 (XSD) 架构。由于 DataSet 固有的序列

化格式为 XML, 它是在层间移动数据的优良媒介, 这使 DataSet 成为以远程方式向 XML Web services 发送数据和架构上下文以及从 XML Web services 接收数据和架构上下文的最佳选择。

2. .NET 数据提供程序

ADO.NET 结构的另一个核心元素是 .NET 数据提供程序, 它的设计是为了实现数据操作和对数据的快速、只进、只读访问。.NET 数据提供程序包括 Connection、Command、DataReader 和 DataAdapter 4 种对象。各对象的作用如下。

- Connection 对象提供与数据源的连接。
- Command 对象使用户能够访问用于返回数据、修改数据、运行存储过程以及发送或检索参数信息的数据库命令。
- DataReader 对象从数据源中提供高性能的数据流。
- DataAdapter 对象提供连接 DataSet 对象和数据源的桥梁。并且 DataAdapter 使用 Command 对象在数据源中执行 SQL 命令, 以便将数据加载到 DataSet 中, 并使对 DataSet 中数据的更改与数据源保持一致。

ADO.NET 两种组件间的关系如图 1-1 所示。

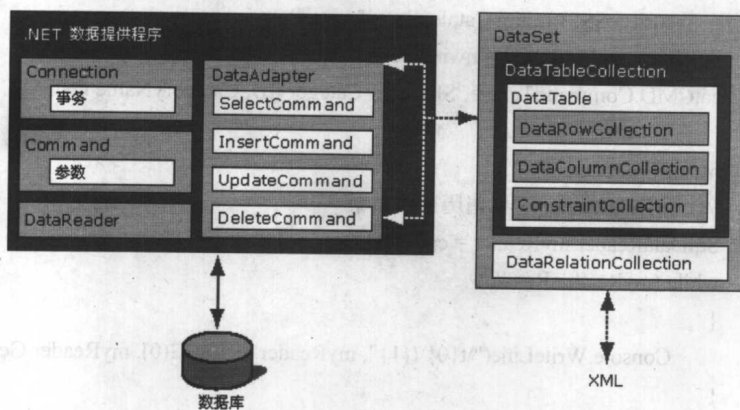


图 1-1 ADO.NET 结构图

3. ADO.NET 平台要求

支持微软 .NET 框架 SDK(包括 ADO.NET) 的平台包括: Windows 2000、Windows NT 4(带 Service Pack 6a)、Windows ME、Windows 98 和 Windows SE。如果要使用 ADO.NET 中 SQL Server .NET 数据提供程序或 OLE DB .NET 数据提供程序, 则必须安装 Microsoft Data Access Components 2.6 或更高版本。

ADO.NET 的定义包含在 System.Data 命名空间中, 例如, SQL Server 提供程序位于 System.Data.SqlClient 命名空间, 而 OleDb 提供程序位于 System.Data.OleDb 命名空间。在 C# 编写的应用程序中使用 ADO.NET 需要包含如下代码:

```
using System.Data;
```

ADO.NET 类存在于 System.Data.dll 中, 并且与 System.Xml.dll 中的 XML 类集成。当编译使用 System.Data 命名空间的代码时, 将引用 System.Data.dll 和 System.Xml.dll。

1.1.3 ADO.NET 示例应用程序

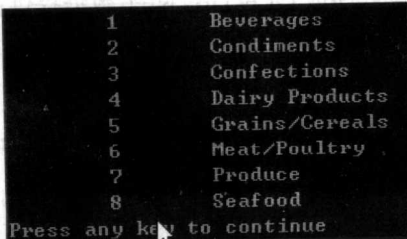
以下代码示例使用 SQL Server .NET 数据提供程序, 并使用 Connection、Command、DataReader 等对象从 SQL Server 的 Northwind 数据库检索 Categories 表, 最后从控制台输出。

```
using System;
using System.Data;
using System.Data.SqlClient;

class Sample
{
    public static void Main()
    {
        SqlConnection nwindConn = new SqlConnection("Data Source=localhost;Integrated
            Security=SSPI;Initial Catalog=northwind");
        SqlCommand catCMD = nwindConn.CreateCommand();
        catCMD.CommandText = "SELECT CategoryID, CategoryName FROM Categories";

        nwindConn.Open();
        //使用 DataReader 对象遍历查询结果
        SqlDataReader myReader = catCMD.ExecuteReader();
        while (myReader.Read())
        {
            Console.WriteLine("\t{0}\t{1}", myReader.GetInt32(0), myReader.GetString(1));
        }
        myReader.Close();
        //关闭连接
        nwindConn.Close();
    }
}
```

程序执行结果如图 1-2 所示。



```
1 Beverages
2 Condiments
3 Confections
4 Dairy Products
5 Grains/Cereals
6 Meat/Poultry
7 Produce
8 Seafood
Press any key to continue
```

图1-2 示例程序执行结果

注意:

使用不同的 .NET 数据提供程序所包含的命名空间不同, 例如上面示例使用 `using System.Data.SqlClient` 包含 SQL Server .NET 数据提供程序的命名空间。

1.2 .NET 数据提供程序

.NET 框架中的数据提供程序用于连接到数据源, 检索和修改数据源中的数据。开发人员可以直接处理检索到的结果, 也可以将其放入 ADO.NET DataSet 对象, 在实际应用中, .NET 数据提供程序在数据源和 ADO.NET DataSet 之间起着桥梁的作用。NET 数据提供程序在设计上是轻量的, 它在数据源和代码之间创建了一个最小层, 因此在不以牺牲功能为代价的前提下提高了性能。表 1-1 列出了基于 .NET 框架中的 .NET 数据提供程序。

表1-1 .NET 数据提供程序列表

.NET 数据提供程序	说 明
SQL Server .NET 数据提供程序	使用 SQL Server 7.0 或更高版本
OLE DB .NET 数据提供程序	使用 OLE DB 公开的数据源
ODBC .NET 数据提供程序	使用 ODBC 公开的数据源
ORACLE .NET 数据提供程序	使用 ORACLE 数据源

注意:

.NET 框架包括 SQL Server .NET 数据提供程序和 OLE DB .NET 数据提供程序。包括开放式数据库连接(ODBC) .NET 数据提供程序以及 ORACLE .NET 数据提供程序在内的其他数据提供程序需要从 <http://msdn.microsoft.com/downloads> 下载。不过, 它们实现的结构与 SQL Server .NET 数据提供程序和 OLE DB .NET 数据提供程序相同。因此, 本节提供的信息也可应用于其他 .NET 数据提供程序。

SQL Server .NET 数据提供程序位于 `System.Data.SqlClient` 命名空间, 它针对 SQL Server 访问进行了优化, 与通过 OLE DB 使用 ADO 相比, 它能够提供更好的性能。此外, 该提供程序还具有一个智能连接缓冲池机制, 因此, 与以前的 OLE DB 或 ODBC 相比, 能够提供更快的缓冲数据库连接访问。

OLE DB .NET 提供程序位于 `System.Data.OleDb` 命名空间, 可以通过 OLE DB 与数据库进行交互。当前大多数的数据库系统可以由 ADO.NET 通过现有的 OLE DB 提供程序和 OLE DB .NET 数据提供程序来进行访问、交互。

SQL Server .NET 数据提供程序和 OLE DB .NET 数据提供程序之间的对比如图 1-3 所示。

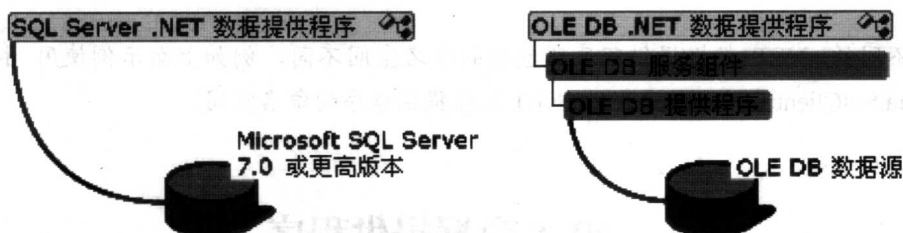


图 1-3 SQL Server .NET 数据提供程序和 OLE DB .NET 数据提供程序之间的比较

Connection、Command、DataReader、DataAdapter 组成 .NET 数据提供程序的 4 个核心对象，表 1-2 概括了上述核心对象。

表 1-2 .NET 数据提供程序的 4 个核心对象

对 象	说 明
Connection	建立与特定数据源的连接
Command	对数据源执行命令。公开 Parameters，并且可以从 Connection 在 Transaction 的范围内执行
DataReader	从数据源中读取只进且只读的数据流
DataAdapter	用数据源填充 DataSet 并解析更新

除表 1-2 列出的核心对象之外，.NET 数据提供程序还包含表 1-3 列出的类。

表 1-3 .NET 数据提供程序提供的其他类

对 象	说 明
Transaction	在数据源的事务中登记命令
CommandBuilder	自动生成 DataAdapter 的命令属性或将从存储过程导出参数信息并填充 Command 对象的 Parameters 集合
Parameter	定义命令和存储过程的输入、输出和返回值参数
Exception	在数据源中遇到错误时返回。对于在客户端遇到的错误，.NET 数据提供程序会引发 .NET 框架异常
Error	公开数据源返回的警告或错误中的信息
ClientPermission	为 .NET 数据提供程序代码访问安全属性而提供

1.2.1 使用 ADO 连接到数据源

在 ADO.NET 中，使用 Connection 对象来连接到指定的数据源。若要连接到 Microsoft SQL Server 7.0 或更高版本，建议使用 SQL Server .NET 数据提供程序的 SqlConnection 对象。若要连接到 OLE DB 数据源或者 Microsoft SQL Server 6.x 等较早版本，建议使用 OLE DB .NET 数据提供程序的 OleDbConnection 对象。

1. 使用 SQL Server .NET 数据提供程序连接到 SQL Server

SQL Server .NET 数据提供程序使用 `SqlConnection` 对象提供了与 Microsoft SQL Server 7.0 或更高版本的连接。SQL Server .NET 数据提供程序通过 `SqlConnection.ConnectionString` 属性支持类似于 OLE DB (ADO) 连接字符串格式, 该连接字符串包含源数据库名称和建立初始连接所需的其他参数, 默认值为空。

以下代码示例演示使用 `SqlConnection` 创建和打开与 SQL Server 数据库连接, 其中, 示例 1 对连接字符串属性单独赋值。并且, 示例 1 采用 SQL Server 集成安全(见连接字符串中 `Integrated Security=SSPI;`), 而示例 2 则采用明确的信任机制, 使用用户 ID 和密码建立连接。对比两段代码, 可以发现使用 SQL Server 集成安全(也叫做可信任连接)是连接到 SQL Server 数据库更可靠的方法, 因为它不会在连接字符串中暴露用户 ID 以及密码。

示例 1 代码如下所示。

```
SqlConnection nwindConn = new SqlConnection("Data Source=localhost; Integrated  
Security=SSPI;" + "Initial Catalog=northwind");  
nwindConn.Open();
```

示例 2 代码如下所示。

```
SqlConnection myConnection = new SqlConnection();  
myConnection.ConnectionString = "user id=;password=;initial catalog=northwind;  
data source=local;Connect Timeout=30";  
myConnection.Open();
```

为了将数据连接返回数据库连接池, 每次使用完 `Connection` 后都必须将其关闭。关闭数据连接可以使用 `Connection` 对象的 `Close` 或 `Dispose` 方法来实现。当 `Connection` 对象处于范围之外或者已通过垃圾回收时, 连接不会隐式释放。

注意:

不要在用户类的 `Finalize` 函数中, 调用 `Connection`、`DataReader` 或其他对象的 `Close` 或 `Dispose` 方法, 因为根据垃圾回收原理, `Finalize` 函数只负责回收用户类自身不再引用的资源。

2. 使用 OLE DB .NET 数据提供程序连接到 SQL Server

OLE DB .NET 数据提供程序通过 `OleDbConnection` 对象提供了与使用 OLE DB 公开的数据源的连接, 对 SQL Server 6.x 或较早版本, 使用 SQL Server 的 OLE DB 提供程序 `SQLOLEDB` 连接。

对于 OLE DB .NET 数据提供程序, 连接字符串格式与 ADO 中使用的连接字符串格式基本相同, 但存在以下例外:

- `Provider` 关键字是必需关键字。
- 不支持 `URL`、`Remote Provider` 和 `Remote Server` 关键字。