



程序员书库

陈文军 陈晓铭 编著

Visual Basic.NET 数据库编程



清华大学出版社

程序员书库

Visual Basic.NET 数据库编程

陈文军 陈晓铭 编著

清华大学出版社

北 京

内 容 简 介

本书首先介绍了 ADO.NET 的框架及数据库编程技术, 以便为 Visual Basic.NET 数据库应用程序开发奠定基础。接着以 5 个实例详细讲述了如何开发 Visual Basic.NET 数据库应用程序, 内容包括 Bug 跟踪管理系统、企业进销存管理系统、图书管理系统、考试报名管理系统、学生管理系统等, 这些实例分别从系统功能设计、系统模块设计、数据库设计、系统详细设计等方面进行讲解, 完整地讲述了设计一个项目所必须经过的所有步骤。

本书的实例源代码可通过 <http://www.tupwk.com.cn/downpage/index.asp> 下载。

本书主要面向有一定 Visual Basic.NET 基础的编程人员及工程人员。

版权所有, 翻印必究。举报电话: 010-62782989 13901104297 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。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术, 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

Visual Basic.NET 数据库编程/陈文军, 陈晓铭 编著. —北京: 清华大学出版社, 2005.1

(程序员书库)

ISBN 7-302-09807-7

I. V… II. ①陈… ②陈… III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 110136 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: 胡伟卷

文稿编辑: 刘金喜

封面设计: 王 永

版式设计: 康 博

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印 张: 21.75 字 数: 502 千字

版 次: 2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷

书 号: ISBN 7-302-09807-7/TP·6768

印 数: 1~4000

定 价: 32.00 元

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

微软.NET 框架的推出,使得广大 Visual Basic 开发人员拥有了一个全新的编码平台,与 Visual Basic 6.0 相比,它不仅带来了许多重要的变化,而且增加了很多新功能。Visual Basic.NET 在数据库编程方面通过新的 ADO.NET 数据访问方法增强了原有的 ADO 的功能,提供了一整套检索和操作数据的类。

本书首先从 ADO.NET 的框架以及一些重要的数据库编程技术讲起,接着以 5 个实例详细讲述了如何开发数据库应用程序,从系统分析、功能设计、数据库设计、模块设计,到模块详细设计,再到联机帮助的制作、软件发布,让读者了解 ADO.NET 对象模型以及它的工作原理,学习如何使用 ADO.NET 产生的类访问数据库,以及编写和使用存储过程、视图、触发器、数据绑定和事务处理等,以使读者能够理解和应用系统的设计与开发思想。同时,在实例开发中,对 Visual Basic.NET 中控件的使用、编程技巧、错误处理也作了相应叙述。

本书的读者对象是对 Visual Studio.NET 和 Visual Basic 先前版本有一定了解的中级用户。另外,读者还应具备面向对象程序设计(Object Oriented Programming, OOP)、ADO.NET 或 ADO、数据库设计的一些基本知识。本书中部分章节适合初学者阅读,而另外一些章节则适合水平更高的读者阅读。

本书分为 3 部分:

- 第 1 部分包括第 1 章和第 2 章。第 1 章介绍 ADO.NET 的框架、ADO.NET 数据库访问方法、构建自己的数据库访问对象。第 2 章主要介绍数据库编程技术,涉及 SQL、PL/SQL 基础、编写存储过程和包、事务处理等,并且通过示例详细讲述了数据绑定、使用存储过程、报表打印和数据输出技术。
- 第 2 部分包括第 3 章到第 7 章。该部分内容非常丰富,详细讲解了实际的数据库应用程序的开发过程。在 5 个实例中,使用 ADO.NET 分别连接和操作 Access、SQL Server 和 Oracle 9i 数据库,对 DataGrid、TreeView、ListView、水晶报表等控件的灵活使用以及编程技巧作了相应的描述。
- 第 3 部分包括第 8 章和第 9 章。该部分讲述了联机帮助的制作以及软件的发布。

本书编写过程中,张江涛、徐日强、赵磊、杜江、刘旭、周鸣扬、范翠丽、刘伟、陈策、郑艳华、谢振华、唐兵、张俊岭、尹建民、冉林仓、李东玉等也参加了部分内容的编写及素材整理工作,在此一并表示感谢。

由于时间仓促,作者水平有限,不当之处在所难免,欢迎读者批评指正。

读者可通过 <http://www.tupwk.com.cn/downpage/index.asp> 下载本书中的实例源代码。

读者可以通过 chenxm_03@163.com 或 wenwoods@163.com 与我们联系,我们将尽力回答关于此书的各种问题。

作 者

目 录

第 1 章 数据访问方法与对象	1
1.1 VB 中的数据访问方法	1
1.1.1 高级数据连接器(ADC)	1
1.1.2 Active 数据对象(ADO)	1
1.1.3 数据访问对象(DAO)	2
1.1.4 开放式的数据库连接(ODBC)	3
1.1.5 OLEDB 数据访问	4
1.1.6 远程数据对象(RDO)	4
1.2 ADO.NET 数据访问	4
1.2.1 ADO.NET 概述	5
1.2.2 ADO.NET 和 ADO 的比较	6
1.2.3 ADO.NET 对象模型	8
1.3 Connection 对象	10
1.3.1 连接字符串	10
1.3.2 使用 Connection 对象	13
1.4 DataAdapter 对象	13
1.4.1 填充数据集	13
1.4.2 填充架构	14
1.4.3 更新数据库	14
1.5 Command 对象	16
1.5.1 创建 Command	16
1.5.2 执行 Command	17
1.5.3 带存储过程的 Command	18
1.6 DataReader 对象	19
1.6.1 读取数据	19
1.6.2 返回多个结果集	20
1.7 DataSet 对象	20
1.7.1 在 DataSet 中使用关联	22
1.7.2 在 DataSet 中使用约束	24
1.7.3 更新 DataSet 数据	26
1.7.4 更新数据源数据	30
1.8 XML 集成	32
1.8.1 从 XML 中加载 DataSet	32
1.8.2 以 XML 数据形式编写 DataSet	34
1.9 建立 OleDbDatabase 对象	35

1.10	建立 SqlDatabase 对象	37
1.11	使用对象的简单例子	41
1.12	小结	43
第 2 章	数据库编程技术	44
2.1	SQL 语言	44
2.1.1	SQL	44
2.1.2	PL/SQL	50
2.2	数据绑定	56
2.2.1	使用向导进行数据绑定	56
2.2.2	编写代码进行数据绑定	60
2.3	使用存储过程	61
2.3.1	在 VS.NET 中创建存储过程	62
2.3.2	返回单个记录集	63
2.3.3	返回多个数据集	64
2.3.4	SqlParameter 对象	65
2.3.5	输入参数	66
2.3.6	输出参数	67
2.3.7	返回参数	69
2.3.8	调用 PL/SQL 包	71
2.4	事务和锁定	75
2.4.1	ACID 属性	75
2.4.2	事务命令	76
2.4.3	使用事务	78
2.4.4	锁定和隔离	79
2.4.5	有效使用事务和锁定	80
2.5	数据输出	81
2.5.1	输出到文本文件	81
2.5.2	输出到 HTML 文件	82
2.5.3	输出到 Excel	84
2.5.4	输出到 Word	86
2.5.5	输出报表	88
2.6	小结	92
第 3 章	Bug 跟踪管理系统	93
3.1	综述	93
3.2	系统功能设计	94
3.2.1	Bug 描述	95
3.2.2	Bug 状态转换	96

3.2.3	人员描述	96
3.2.4	角色定义	97
3.2.5	Bug 提交以及查询	97
3.3	系统模块设计	98
3.4	数据库设计	101
3.5	模块详细设计	105
3.5.1	主界面模块	107
3.5.2	测试数据库连接模块	111
3.5.3	信息修改模块	114
3.5.4	角色定义模块	115
3.5.5	状态定义模块	119
3.5.6	人员管理模块	124
3.5.7	项目组管理模块	125
3.5.8	Bug 提交模块	127
3.5.9	Bug 分配模块	130
3.5.10	Bug 查询模块	135
3.6	小结	143
第 4 章	企业进销存管理系统	144
4.1	系统功能设计	144
4.2	系统模块设计	145
4.3	数据库设计	147
4.3.1	创建数据库	147
4.3.2	数据表设计	147
4.3.3	建立表关系	152
4.3.4	编写存储过程	153
4.4	系统详细设计	157
4.4.1	主窗体模块	158
4.4.2	用户登录模块	160
4.4.3	用户信息管理模块	161
4.4.4	仓库信息管理模块	166
4.4.5	客户信息管理模块	168
4.4.6	商品类目管理模块	173
4.4.7	商品信息管理模块	175
4.4.8	商品交易信息管理模块	178
4.4.9	商品流通信息管理模块	183
4.4.10	财务管理模块	186
4.5	小结	190

第 5 章	图书管理系统	191
5.1	系统功能设计	191
5.2	系统模块设计	193
5.3	数据库设计	195
5.4	系统详细设计	197
5.4.1	登录模块	198
5.4.2	主界面模块	199
5.4.3	图书管理模块	202
5.4.4	读者管理模块	206
5.4.5	添加图书模块	207
5.4.6	添加读者模块	209
5.4.7	借书模块	210
5.4.8	还书模块	214
5.4.9	查询模块	217
5.4.10	打印报表模块	218
5.4.11	统计排名模块	222
5.4.12	设置模块	223
5.5	小结	226
第 6 章	考试报名管理系统	227
6.1	系统功能设计	227
6.2	系统模块设计	228
6.3	数据库设计	229
6.3.1	创建数据表	229
6.3.2	创建包	234
6.4	系统详细设计	242
6.4.1	主窗体模块	243
6.4.2	考试项目登录模块	245
6.4.3	考试项目管理模块	246
6.4.4	学生信息管理模块	250
6.4.5	考场信息管理模块	258
6.4.6	考场设置模块	262
6.4.7	打印准考证模块	267
6.5	小结	271
第 7 章	学生管理系统	272
7.1	系统功能设计	272
7.2	系统模块设计	274
7.3	数据库设计	275

7.4	系统详细设计	277
7.4.1	主界面模块	278
7.4.2	数据库连接模块	279
7.4.3	用户登录模块	281
7.4.4	学院管理模块	282
7.4.5	专业管理模块	284
7.4.6	班级管理模块	287
7.4.7	课程管理模块	289
7.4.8	新生注册模块	291
7.4.9	学生查询模块	295
7.4.10	选课登记模块	298
7.4.11	成绩登记模块	301
7.4.12	成绩查询模块	303
7.4.13	报表输出模块	304
7.5	小结	305
第 8 章	制作联机帮助	306
8.1	Help Workshop 介绍	306
8.2	使用 HtmlHelp Workshop 创建工程	307
8.3	创建目录	311
8.4	创建索引和关键字	313
8.5	编译生成 CHM 帮助文件	315
8.6	在应用程序中提供帮助	316
8.6.1	显示弹出帮助	316
8.6.2	提供上下文帮助	317
8.6.3	启动 CHM 文件	318
8.7	小结	318
第 9 章	软件发布	319
9.1	创建应用程序	319
9.2	创建部署项目	320
9.3	将应用程序添加到安装程序	321
9.4	创建快捷方式	322
9.5	创建文件关联	323
9.6	添加注册表项	324
9.7	添加自定义安装对话框	325
9.8	添加启动条件	326
9.9	创建数据库	327
9.10	小结	334

第 1 章 数据访问方法与对象

本章介绍 Visual Basic.NET(以下简称 VB.NET)数据库编程的基础知识,着重讲述 ADO.NET 数据库访问方法,ADO.NET 的体系结构,ADO.NET 常用对象及其使用。最后,在此基础上,构建了两个对象 OleDbDatabase 和 SqlDatabase,并编写了一些简单的测试例子。

本章要点:

- VB.NET 数据访问方法
- ADO.NET 体系结构
- ADO.NET 常用对象
- OleDbDatabase 对象
- SqlDatabase 对象

1.1 VB 中的数据访问方法

VB 中常用的数据访问方法有 ADC、ADO、DAO、ODBC、OLEDB 和 RDO,下面对这些访问方法逐一简单介绍。

1.1.1 高级数据连接器(ADC)

ADC(Advanced Data Connector)用于将 ADO 数据源绑定到窗体的数据绑定控件上。ADC 是一种直接访问或者通过 ADO 访问远程 OLEDB 对象的技术,它也支持应用在微软 Internet Explorer 浏览器上的数据绑定控件。它是专门为 Web 浏览器相关的应用程序设计的。

1.1.2 Active 数据对象(ADO)

ADO(Active Data Objects)实际是一种用于访问各种数据类型的连接机制。ADO 是一种极简单的数据访问格式,它通过 ODBC 的方法建立的数据库接口,可以使用任何

一种 ODBC 数据源访问数据库中的数据,即不只适用于 SQL Server、Oracle、Access 等数据库应用程序,也适用于 Excel 表格、文本文件、图形文件和无格式的数据文件。ADO 是基于 OLEDB 之上的技术,因此 ADO 通过其内部的属性和方法提供统一的数据访问接口方法。

ADO 是 DAO/RDO 的后继产物。ADO 2.0 在功能上与 RDO 更相似,而且一般来说,这两种模型之间有一种相似的映射关系。ADO “扩展”了 DAO 和 RDO 所使用的对象模型,这意味着它包含较少的对象、更多的属性、方法(和参数)以及事件。

作为较新的数据库访问模式,ADO 也是简单易用的,所以微软已经明确表示今后把重点放在 ADO 上,对 DAO/RDO 不再作升级,所以 ADO 已经成了当前数据库开发的主流。

要使用 ADO,必须清楚 ADO 的对象层次结构,其大体上分为以下 7 个对象层次。

- Command 对象:包含关于某个命令的信息,例如查询字符串、参数定义等。Command 对象在功能上和 RDO 的 rdoQuery 对象相似。
- Connection 对象:包含关于某个数据提供程序的信息。Connection 对象在功能上和 RDO 的 rdoConnection 对象是相似的,并且包含了关于结构描述的信息。它还包含某些 rdoEnvironment 对象的功能,例如 transaction 控件。
- Error 对象:包含数据提供程序出错时的扩展信息。Error 对象在功能上和 RDO 的 rdoError 对象相似。
- Field 对象:包含记录集中数据的某单个列的信息。Field 对象在功能上和 RDO 的 rdoColumn 对象相似。
- Parameter 对象:包含参数化的 Command 对象的某单个参数的信息。该 Command 对象有一个包含其所有 Parameter 对象的 Parameters 集合。Parameter 对象在功能上和 RDO 的 rdoParameter 对象相似。
- Property 对象:包含某个 ADO 对象的提供程序定义的特征。没有任何等同于该对象的 RDO,但 DAO 有一个相似的对象。
- Recordset 对象:用来存储数据操作返回的记录集。此对象和 Connection 对象是所有对象中最重要的两个对象。

1.1.3 数据访问对象(DAO)

DAO(Data Access Objects)是一种面向对象的接口。通过 DAO/Jet 功能可以访问 ISAM 数据库,使用 DAO/ODBCDirect 功能可以实现远程 RDO 功能。使用 DAO 的程序编码非常简单,DAO 提供丰富的游标(Cursor)类型的结果集和非游标(Cursor-Less)类型的结果集,同 DDL(数据描述语言)的功能很类似。

DAO 模型是设计关系数据库系统结构的对象类的集合。它们提供了完成管理这样一个系统所需的全部操作的属性和方法,包括创建数据库,定义表、字段和索引,建立表间的关系,定位和查询数据库等工具。

数据访问对象用来显示 Microsoft Jet 数据库引擎(最早是被 Microsoft Access 使用,现在已经支持其他数据库),并允许开发者通过 ODBC 直接连接到其他数据库一样,直接连接到 Access 表。DAO 最适用于单系统应用程序或在小范围本地分布使用。其内部已经对 Jet 数据库的访问进行了加速优化,而且其使用也是很方便的。所以,如果数据库是 Access 数据库且是本地使用,建议使用这种访问方式。

VB 已经把 DAO 模型封装成了 Data 控件,分别设置相应的 DatabaseName 属性和 RecordSource 属性,就可以将 Data 控件与数据库中的记录源连接起来,以后就可以使用 Data 控件来对数据库进行操作。

1.1.4 开放式的数据库连接(ODBC)

ODBC(Open Database Connectivity)是一种公认的关系数据源的接口。它快而轻,并且提供统一的接口,ODBC 对任何数据源都未作优化。

ODBC 也是微软公司开放服务结构(Windows Open Services Architecture, WOSA)中有关数据库的一个组成部分,它建立了一组规范,并提供了一组对数据库访问的标准 API(应用程序编程接口)。这些 API 利用 SQL 来完成其大部分任务。ODBC 本身也提供了对 SQL 语言的支持,用户可以直接将 SQL 语句发送给 ODBC。

一个基于 ODBC 的应用程序对数据库的操作不依赖任何 DBMS,不直接与 DBMS 打交道,所有的数据库操作由对应的 DBMS 的 ODBC 驱动程序完成。也就是说,不论是 FoxPro、Access,还是 Oracle 数据库,均可用 ODBC API 进行访问。由此可见,ODBC 的最大优点是能以统一的方式处理所有的数据库。

一个完整的 ODBC 由以下几个部件组成。

1. 应用程序(Application)

- ODBC 管理器(Administrator)。该程序位于 Windows 控制面板(Control Panel)的 32 位 ODBC 内,其主要任务是管理安装的 ODBC 驱动程序和管理数据源。
- 驱动程序管理器(Driver Manager)。驱动程序管理器包含在 ODBC32.DLL 中,对用户是透明的,是 ODBC 中最重要的部件。其任务是管理 ODBC 驱动程序。

2. ODBC API

ODBC 驱动程序是一些 DLL,提供了 ODBC 和数据库之间的接口。

数据源包含了数据库位置和数据库类型等信息,实际上是一种数据连接的抽象。

ODBC 连接目前仅仅限于关系型数据库,对于其他数据源比如 Excel、文本文件都不能进行访问,而且有很多 DBMS(数据库管理系统)都不能充分支持其所有的功能。相比之下,OLEDB 可以存取任何形式的数据,所以其功能相当强大,它也指引了目前技术发展的方向。

1.1.5 OLEDB 数据访问

OLEDB 是一种底层数据访问接口，是用于第三方驱动程序商家开发输出数据源到 ADO 技术的应用程序或用于 C++ 的开发者开发定制的数据库组件。OLEDB 不能被 VB 直接调用。

OLEDB 是用于访问数据的重要的系统级编程接口，它是 ADO 的基础技术，同时也是 ADO.NET 的数据源。OLEDB 是用于访问所有类型的数据的开放式标准，这些数据既包括关系数据又包括非关系数据：大型机 ISAM/VSAM 和分层数据库，电子邮件和文件系统存储区，文本、图形和地理数据以及自定义的业务对象。

OLEDB 提供对数据一致的、高性能的访问，并支持各种开发需要，包括使用与关系数据库或其他存储区中数据的活连接来创建前端数据库客户端和中间层业务对象。

1.1.6 远程数据对象(RDO)

RDO(Remote Data Objects)为使用代码来创建和操作一个远程 ODBC 数据库系统的各个部件提供了一个框架。对象和集合都具有描述数据库的各个部件特征的属性以及用来操作这些部件的方法。可以在对象和集合之间建立起关系，这些关系就代表了数据库系统的逻辑结构。RDO 是 ODBC API 的一个浅层接口，是专为访问远程 ODBC 关系数据源而设计的。

RDO 同易于使用的 DAO style 组合在一起，提供了一个接口，形式上展示出所有 ODBC 的底层功能和灵活性。尽管 RDO 在很好地访问 Jet 或 ISAM 数据库方面受到限制，而且它只能通过现存的 ODBC 驱动程序来访问关系数据库。但是，RDO 已被证明是许多 SQL Server、Oracle 以及其他大型关系数据库开发者经常选用的最佳接口。RDO 提供了用来访问存储过程和复杂结果集的更多和更复杂的对象、属性以及方法。和 DAO 一样，在 VB 中也把其封装为 RDO 控件了，其使用方法与 DAO 控件的使用方法完全一样。

1.2 ADO.NET 数据访问

2001 底，Microsoft 公司向用户推出了 Visual Studio.NET 开发环境，.NET 平台引入了 C++ 中支持类、对象、继承、重载等面向对象的编程机制，特别是在数据访问领域彻底改进了 VB 6.0 中的 ADO 基于 COM+ 的技术特征，ADO.NET 支持脱机记录集与 XML 数据文档格式，把模式(Schema)和数据视为分离的可替换元素，这些技术使记录集很容易在分布式、异种平台构成的环境中使用，使数据访问变得更加简单和高效。

1.2.1 ADO.NET 概述

ADO.NET 是由一系列的数据库相关类和接口组成的, 它的基石是 XML 技术, 所以通过 ADO.NET 不仅能访问关系型数据库中的数据, 而且还能访问层次化的 XML 数据。

ADO.NET 提供了两种数据访问的模式: 一种为连接模式(Connected), 另一种为非连接模式(Disconnected)。运用过 ADO 技术的编程人员对前一种模式应该是非常熟悉的, 而后一种模式则是 ADO.NET 中才具有的。相比于传统的数据库访问模式, 非连接的模式提供了更大的可升级性和灵活性。在该模式下, 一旦应用程序从数据源中获得所需的数据, 它就断开与原数据源的连接, 并将获得的数据以 XML 的形式存放在主存中。在应用程序处理完数据后, 它再取得与原数据源的连接并完成数据的更新工作。

ADO.NET 中的 DataSet 类是非连接模式的核心, 数据集对象(DataSet)以 XML 的形式存放数据。我们既可以从一个数据库中获取一个数据集对象, 也可以从一个 XML 数据流中获取一个数据集对象。而从用户的角度来看, 数据源在哪里并不重要, 也是无需关心的。这样一个统一的编程模型就可被运用于任何使用了数据集对象的应用程序。

ADO.NET 有效地从数据操作中将数据访问分解为多个可以单独使用或一前一后使用的非连续组件。ADO.NET 包含用于连接到数据库、执行命令和检索结果的 .NET Framework 数据提供程序。用户可以直接处理检索到的结果, 或将其放入 ADO.NET DataSet 对象, 以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。ADO.NET DataSet 对象也可以独立于 .NET Framework 数据提供程序使用, 以管理应用程序本地的数据或源自 XML 的数据。

在 ADO.NET 中, 可以使用新的 .NET Framework 数据提供程序来访问数据源。这些数据提供程序包括:

- SQL Server .NET Framework 数据提供程序;
- OLE DB .NET Framework 数据提供程序;
- ODBC .NET Framework 数据提供程序;
- Oracle .NET Framework 数据提供程序。

这些数据提供程序可以满足各种开发要求, 包括中间层业务对象(它们使用与关系数据库和其他存储区中的数据的活动连接)。

ADO.NET 是专为基于消息的 Web 应用程序而设计的, 同时还能为其他应用程序结构提供较好的功能。通过支持对数据的松耦合访问, ADO.NET 减少了与数据库的活动连接数目(即减少了多个用户争用数据库服务器上的有限资源的可能性), 从而实现了最大程度的数据共享。

在有些情况下, Web 应用程序或 XML Web Services 需要访问多个源中的数据, 或者需要与其他应用程序(包括本地和远程应用程序)进行互操作, 或者可受益于保持

和传输缓存结果,这时使用数据集将是一个明智的选择。作为一种替换方法,ADO.NET 提供数据命令和数据读取器以便与数据源直接通信。使用数据命令和数据读取器直接进行的数据库操作包括:运行查询和存储过程,创建数据库对象,使用 DDL 命令直接更新和删除。

ADO.NET 还通过对分布式 ADO.NET 应用程序的基本对象“数据集”(Dataset)提供基于 XML 的持久性和传输格式,来实现最大程度的数据共享。数据集是一种关系数据结构,可使用 XML 进行读取、写入或序列化。ADO.NET 数据集使得生成在应用程序层与多个 Web 站点之间进行松耦合数据交换的应用程序变得很方便。

1.2.2 ADO.NET 和 ADO 的比较

ADO.NET 和 ADO 的差别主要体现在以下几个方面。

1. 数据在内存中的表示形式

在 ADO 中,数据在内存中的表示形式为记录集。在 ADO.NET 中,它为数据集。它们之间重要的差异。

2. 表的个数

记录集看起来像单个表。如果记录集将包含来自多个数据库表的数据,则它必须使用 JOIN 查询,将来自各个数据库表的数据组合到单个结果表中。

相反,数据集是一个或多个表的集合。数据集内的表称为数据表。明确地说,它们是 DataTable 对象。如果数据集包含来自多个数据库表的数据,它通常将包含多个 DataTable 对象,即每个 DataTable 对象通常对应于单个数据库表或视图。这样,数据集可以模仿基础数据库的结构。

数据集通常还包含关系。数据集内的关系类似于数据库中的外键关系,即它使多个表中的行彼此关联。例如,如果数据集包含一个有关投资者的表和另一个有关每个投资者的股票购买情况的表,则数据集可能还包含一个关系来连接投资者表的各个行和购买表的对应行。

由于数据集可以保存多个独立的表并维护有关表之间关系的信息,因此它可以保存比记录集丰富得多的数据结构,包括自关联的表和具有多对多关系的表。

3. 数据导航和游标

在 ADO 中,使用 ADO MoveNext 方法顺序扫描记录集的行。在 ADO.NET 中,行表示为集合,因此可以像依次通过任何集合那样依次通过表,或通过序号索引或主键索引访问特定行。DataRelation 对象维护有关主记录和详细资料记录的信息,并提供方法使用户可以获取与正在操作的记录相关的记录。例如,从 Investor 表的 Nate Sun 的行开始,可以定位到 Purchase 表中描述其购买情况的那组行。

“游标”是数据库元素，它控制记录导航、更新数据的能力和其他用户对数据库所作更改的可见性。ADO.NET 不具有固有的游标对象，而是包含提供传统游标功能的数据类。例如，在 ADO.NET DataReader 对象中提供只进、只读游标的功能。有关游标功能的更多信息，请参见数据访问技术。

4. 将打开连接的时间降至最低

在 ADO.NET 中，打开连接的时间仅足够执行数据库操作，例如“选择”(Select) 或“更新”(Update)。可以将行读入数据集中，然后在不保持与数据源连接的情况下使用它们。在 ADO 中，记录集可以提供不连接的访问，但 ADO 主要是为连接的访问设计的。

ADO 和 ADO.NET 中的不连接处理之间存在一个显著差异。在 ADO 中，通过调用 OLEDB 提供程序来与数据库通信。但在 ADO.NET 中，通过数据适配器(OleDbDataAdapter、SqlDataAdapter、OdbcDataAdapter 或 OracleDataAdapter 对象)与数据库通信，这将调用 OLEDB 提供程序或基础数据源提供的 API。ADO 和 ADO.NET 之间的主要区别在于：在 ADO.NET 中，数据适配器允许控制将对数据集所作的更改传输到数据库的方式，方法是实现性能优化、执行数据验证检查或添加其他任何额外处理。

提示：

数据适配器、数据连接、数据命令和数据读取器是组成 .NET Framework 数据提供程序的组件。Microsoft 和第三方供应商可能会提供其他提供程序，这些提供程序也可集成到 Visual Studio 中。

5. 在应用程序间共享数据

在应用程序间传输 ADO.NET 数据集比传输 ADO 不连接的记录集要容易得多。若要将 ADO 不连接的记录集从一个组件传输到另一个组件，可使用 COM 封送。若要在 ADO.NET 中传输数据，可使用数据集，它可以传输 XML 流。

6. 更丰富的数据类型

COM 封送提供一组有限的数据类型(由 COM 标准定义的那些类型)。由于 ADO.NET 中的数据集传输基于 XML 格式，所以对数据类型没有限制。因此，共享数据集的组件可以使用这些组件一般会使用的任何丰富的数据类型集。

7. 性能

传输大型 ADO 记录集或大型 ADO.NET 数据集会使用网络资源。随着数据量的增长，施加于网络的压力也在增加。ADO 和 ADO.NET 都可以最大限度地降低所传输的数据。但 ADO.NET 还提供另一个性能优势：ADO.NET 不需要数据类型转换。而需要 COM 封送传输记录集的 ADO，则需要将 ADO 数据类型转换为 COM 数据类型。

8. 穿透防火墙

防火墙可以影响试图传输不连接的 ADO 记录集的两个组件。记住, 防火墙通常配置为允许 HTML 文本通过, 但防止系统级请求(如 COM 封送)通过。

因为组件使用 XML 交换 ADO.NET 数据库, 所以防火墙可以允许数据集通过。

9. 访问数据库的步骤

通过 ADO 访问数据库, 一般要通过以下 4 个步骤:

- (1) 创建一个到数据库的链路, 即 ADO.Connection;
- (2) 查询一个数据集, 即执行 SQL, 产生一个 Recordset;
- (3) 对数据集进行需要的操作;
- (4) 关闭数据链路。

在 ADO.NET 里, 这些步骤有很大的变化。ADO.NET 的最重要概念之一是 DataSet。DataSet 是不依赖于数据库的独立数据集。所谓独立就是: 即使断开数据链路, 或者关闭数据库, DataSet 依然是可用的。如果在 ASP 中使用过非连接记录集合(Connectionless Recordset), 那么 DataSet 就是这种技术的最彻底的替代品。

有了 DataSet, ADO.NET 访问数据库的步骤就相应地改变了。

- (1) 创建一个数据库链路;
- (2) 请求一个记录集合;
- (3) 把记录集合暂存到 DataSet;
- (4) 如果需要, 返回第 2 步(DataSet 可以容纳多个数据集);
- (5) 关闭数据库链路;
- (6) 在 DataSet 上做所需要的操作。

DataSet 在内部是用 XML 描述数据的。由于 XML 是一种平台无关、语言无关的数据描述语言, 而且可以描述复杂数据关系的数据, 比如父子关系的数据, 所以 DataSet 实际上可以容纳具有复杂关系的数据, 而且不再依赖于数据库链路。

1.2.3 ADO.NET 对象模型

在具体介绍如何实现通用数据库编程前, 先简要的介绍一下 ADO.NET 的体系结构。在 ADO.NET 体系结构中, 一个非常重要的部分就是数据提供者对象(Data Provider), 它是访问数据库的必备条件。通过它, 可以产生相应的数据集对象, 同时它还提供了连接模式下的数据库访问支持。

图 1-1 显示了 ADO.NET 总体的体系结构。

ADO.NET 中的数据提供者对象包括了数据库连接接口(IDbConnection)、数据库命令接口(IDbCommand)、数据读取器接口(IDataReader)和数据适配器接口(IDbDataAdapter)等不同种类的接口。通过这些接口, 应用程序就可以访问数据库、执行相关的命令操作并获取结果, 获取的结果可以以 XML 数据的形式存放在数据集对