

.NET开发丛书

Microsoft
.net

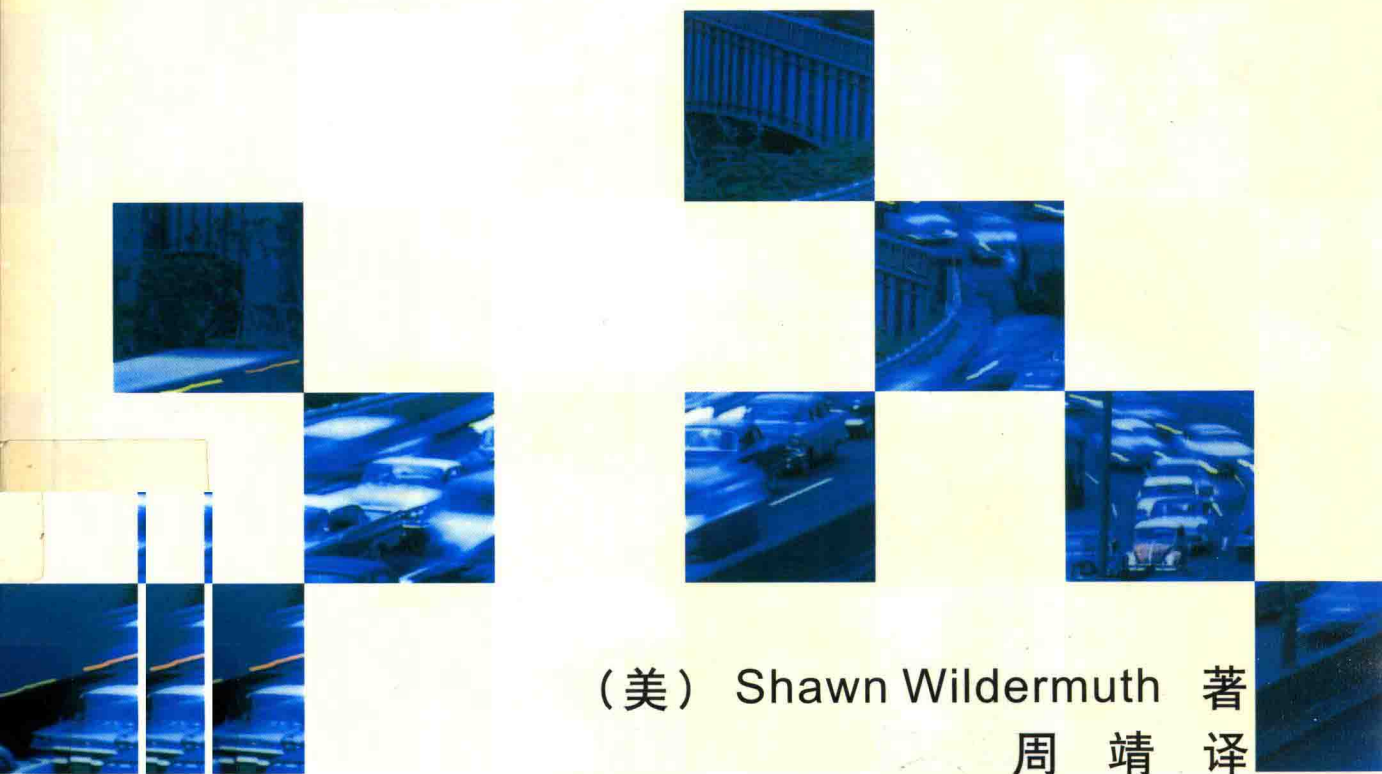


ADO.NET实用指南

——面向Internet世界的数据库访问技术

Pragmatic ADO.NET

Data Access for the Internet World



(美) Shawn Wildermuth 著
周 靖 译



清华大学出版社

.NET 开发丛书

ADO.NET 实 用 指 南

——面向 Internet 世界的数据库访问技术

(美) Shawn Wildermuth 著

周 靖 译

清华大学出版社

北 京

内 容 简 介

全书共 3 部分：“ADO.NET 基础”、“DataSet 详述”和“ADO.NET 实战”。本书提供了大量用 C# 写成的示例代码，作者在为本书提供的配套网站 www.adoguy.com/book 上，为读者提供了相应的 Visual Basic .NET 和 C# 的实现以及技术更新。本书附录同样具有参考价值，其中详细讲解了从 ADO 到 ADO.NET 的迁移策略。对于迫切需要了解 ADO.NET 的开发者而言，本书将是最适合的参考书。

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Pragmatic ADO.NET: Data Access for the Internet World, 1st

Edition by Shawn Wildermuth, Copyright © 2003

EISBN: 0-201-74568-2

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Shawn Wildermuth.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内（不包括中国香港、澳门特别行政区）出版发行。

北京市版权局著作权合同登记号 图字：01-2003-1764

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签，无标签者不得销售。

图书在版编目 (CIP) 数据

ADO.NET 实用指南——面向 Internet 世界的数据库访问技术/ (美) 韦德牧著；周靖译.—北京：清华大学出版社，2003

(.NET 开发丛书)

书名原文：Pragmatic ADO.NET: Data Access for the Internet World

ISBN 7-302-07199-3

I. A… II. ①韦… ②周… 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 (2003) 第 078449 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

文稿编辑：文开棋

封面设计：立日新设计公司

印 刷 者：北京牛山世兴印刷厂

发 行 者：新华书店总店北京发行所

开 本：185×230 印张：21 字数：448 千字

版 次：2003 年 9 月第 1 版 2003 年 9 月第 1 次印刷

书 号：ISBN 7-302-07199-3/TP·5241

印 数：1~4000

定 价：39.00 元

读者对本书的赞誉

★★★★★ ADO.NET 的权威指南

“《ADO.NET 实用指南》出色地介绍了 ADO.NET 数据库编程。作者使用浅显易懂的语言，为初学者阐明了 ADO.NET 的‘大道理’，突出了‘实用’之功效。但与此同时，本书具有一定的深度，可满足中级程序员的需求。书中从介绍数据库连接开始，一直描述到 XML Dataset 数据库编程最佳实践。其间所涉及的每个主题涵盖了大量细节。通过对本书的阅读，我可以这么说，它是您构建数据库系统的‘门户’。作为适合初学者阅读的 ADO.NET 图书，本书提供的简要概述将帮助经验欠缺的程序员跨入 .NET 的殿堂。”

这是我一直在读的少数几本编程类图书，其中的示范代码清楚地说明了书中所涉及的概念。阅读本书时，我不再像以前那样，必须静下心来揣摩书中的示范程序。显然，Shawn Wildermuth 在阐述 ADO.NET 时，确实做到了深入浅出，通过他的解释，非常复杂的观点变得非常容易理解且引人入胜。总的说来，我觉得本书很精彩，很容易懂，而且让我从中学到不少知识。”

——宾西法尼亚州 Jaminson 读者：James J Edelen IV

★★★★★ ADO.NET 最佳实践是全书的亮点

“关于 NET 的书很多，实在令人眼花缭乱，很难做出正确的选择。不过，本书不会让您后悔。它系统地论述了 .NET 数据库编程，解释了哪些该做，哪些不该做。这是我所买的第二本 ADO.NET 书。在此之前，我读过 Bob Beauchemin 的 Essential ADO.NET（中译本《ADO.NET 本质论》，清华大学出版社 2003 年 9 月出版），当时我就认为自己对 ADO.NET 已经了如指掌，不需要再买 ADO.NET 方面的书了。事实证明，我错了。利用 .NET 来完成数据库任务的方式有很多，Shawn Wildermuth 则精心地引导您选择最好的备选方案，他不仅标识出最佳方式，还解释了其缘由。”

这里有一个简单的例子可以证明：数据的扩展性（第 11 章）。Shawn 解释了不同的缓存方案（每用户、每应用程序和每服务器），分析了各种方案的不足，讨论了如何向上扩展 Dataset，以及是应该复制还是隔断。随后，他给出了以下建议（和以前一样，同样附有

详细的解释):

- (1) 使用 Dataset 架构;
- (2) 使用类型化的 Dataset 创建业务规则层;
- (3) 减少到数据库的往返行程;
- (4) 尽早、尽可能频繁地缓存数据;
- (5) 使用 DataReader;
- (6) 使用连接工厂;
- (7) 不要硬编码连接字符串。

从这个例子可以看出, 本书的确能帮助 .NET 数据库开发人员更好地理解 .NET 数据库接口及其最佳实践。”

——挪威 Kolbotn 读者: Vagif Abilov

★★★★★ 通俗易懂, 宜作参考书

“我发现本书很流畅。书中的示范程序非常容易理解, 并适合用作参考书。书中详细解释了 ADO.NET 所涉及的概念, 以及与之相关的方方面面。拿到这本书, 我就将其随身携带, 以便于工作时参考。各章内容中, 我尤其偏爱第 9 章 ‘ADO.NET 和 XML’。从第 1 章开始, 作者就解释了 ADO.NET 概念。这一章提供了大量知识, 帮助我们透彻地理解 ADO.NET 是如何工作的, 数据访问技术是如何揉合到 ADO.NET 中的。第 2 章和第 3 章提供了详细的解释和示范程序, 演示了如何连接到数据库, 以及如何执行 SQL 命令。第 3 章还提供了丰富的信息和示范, 解释了 SQL 命令的运行结果。第 4 章透视了 DataReader, 并介绍了如何使用其大部分功能。第 5 章至第 7 章则完整地描述了 Dataset, 具体包括如何构造 Dataset, Dataset 的重要价值以及一个全新的编程模型 Typed Dataset (类型化的数据集)。第 8 章详细解释了一些比较好的示范程序, 以便说明如何更新数据库, 如何管理并发性 (不管 Dataset 是连接的, 还是断开的)。第 9 章阐述了 ADO.NET 和 XML 之间的关系, 描述了 XML 与 ADO.NET 是如何完美集合的。第 10 章讨论了数据绑定, 这对我来说, 倒比较新鲜, 但我觉得其中的示范程序都非常棒。第 11 章则对全书进行了总结, 提供极富借鉴意义的建议。”

——SoCalNETug.org 南加州 .NET 用户组: Annie W

资料来源: http://www.devasp.net/net/store/shop155/pd0201745682/Books_Software/Books/

序

第一次通过电子邮件结识 Shawn 是在许多年前。他和我参加了同一个邮件组，他经常发表很多有趣的东西。所以，当我做项目时，一旦需要帮助，就立即想到了他。我记得当时是深夜发送的电子邮件，本来预计第二天才会有回复，谁知道 45 分钟之后，就收到了 Shawn 的邮件，要知道他那里比我这里晚 3 个时区！他热情地和我讨论了这个项目，并提供了许多宝贵的经验和思路。那是我们后来长达 3 年的友谊的开始，而且我相信这段友谊还会继续下去。

我们合作的项目结束后，又邀请 Shawn 帮助另一个项目，这次的请求基本来源于 Microsoft 数据库社区，Shawn 受邀向普通大众讲授 Microsoft 最新的数据库技术 ADO.NET。Shawn 热心地建立了自己的 ADO.NET 论坛，并积极地参加 DevelopMentor 新的 .NET 邮件组，解答潮水般涌来的数据库问题。Shawn 从实际的网络世界积累到的 ADO.NET 知识日益精进，终于有一天他问我是否应该写一本书。我告诉他如果想在空闲时间做点事情，就应该去做。要写好一本书（虽然我知道他最终肯定能做到），实在是太难了。除非您有强烈的欲望、坚定的决心和大量的空闲时间，否则是不可能完成的。

令人欣慰的是，Shawn 最后还是应验了“有志者，事竟成”这句老话。他的长时间工作并没有白费，您现在捧在手上的这本书正是他辛勤工作的结晶。您过去可能在网上请 Shawn 帮您解决过数据库方面的问题，也可能这是您第一次接触他的作品，无论如何，正如我一直坚信的那样，他最终都能使您满意！

Chris Sells

www.sellsbrothers.com

前 言

ADO.NET 不是 ADO，这是本书要强调的一个重要事实。要真正理解 ADO.NET，不能只知道如何获取数据库数据，或者只知道如何更新数据库，而是要理解为什么设计 ADO.NET。ADO.NET 用于访问数据库的方式与传统的 Microsoft 数据访问策略完全不同。它和 ADO 完全是两码事。

本书既不是 ADO.NET 的入门参考手册，也不是讲解 ADO.NET 内部工作原理的高阶书。相反，它的目的是帮助开发者快速完成自己的工作。本书的宗旨是为 ADO.NET 的使用者提供清楚、明确的建议。

面向的读者

本书面向具有 .NET 和数据库基础知识的开发者。它的目的是帮助繁忙的开发者快速吸取 ADO.NET 知识，然后用它实际完成自己的工作。本书不要求读者具有 ADO 的经验。当然，如果有的话更佳。

第 1 部分 ADO.NET 基础

第 1 部分帮助您掌握一些必要的基础知识。

第 1 章 ADO.NET 存在的原因

解释如何操纵 ADO.NET 之前，本书解释了 ADO.NET 问世的原因。本章介绍了 Microsoft 数据访问的历史，解释了 ADO.NET 的历史由来。

第 2 章 用 ADO.NET 建立连接

对数据库进行任何操作之前，首先必须知道如何连接它。第 2 章解释如何通过 ADO.NET 来连接一个数据库。我们讨论了连接字符串、许多托管提供程序中都有的连

接池、连接创建模式以及如何尽可能减少与数据库的连接。

第 3 章 操纵数据库

我们与数据库惟一的接触是执行命令，并处理那些命令的结果。本章重点讲解了如何使用 `Command` 对象让数据库为我们工作。我们解释了 `Command` 对象与简单 SQL 语句、参数化查询、存储过程以及批处理查询配合使用的情况。

第 4 章 读取数据

`DataReader` 是 ADO.NET 用于从数据库中获取表的方法。在第 4 章，本书解释了如何使用 `DataReader` 来访问数据库信息，并展示了如何写一个类，该类将 `DataReader` 用作自己的数据源。

第 2 部分 DataSet 详述

`DataSet` 是 ADO.NET 的核心。要想有效地使用 `DataSet`，必须理解如何用它解决日常问题。

第 5 章 构建 DataSet

本章解释了 `DataSet` 是什么，以及为什么要使用它们。本章介绍了许多生成 `DataSet` 的方法，包括使用 `DataAdapter`、XML 以及代码来创建它们的方法。本章还全面地解释了如何使用 `DataSet` 架构。

第 6 章 类型化 DataSet

类型化 `DataSet` 是一种有用的工具，有了它，您的代码能适应不断变化的架构。第 6 章探讨了这种新的编程模型，解释了如何创建类型化 `DataSet`，以及如何将其用作业务逻辑层的基础（一般情况下，业务逻辑层只能人工编码）。本章还展示了如何从 Visual Studio .NET 和命令行创建类型化 `DataSet`。

第 7 章 操纵 DataSet

知道如何创建 `DataSet` 之后，接着要掌握如何用它们来操纵数据，以及如何在 `DataSet` 中组织数据。本章向读者解释了 `DataSet` 模型的工作原理，以及如何用它来完成工作。

第 8 章 更新数据库

在 DataSet 中完成对数据的操纵之后，接着必须根据那些更改来更新数据库。本章指导您如何规划断开式并发性，用实例演示了如何使用内建的乐观并发性，以及如何实现悲观和破坏并发性。除此之外，本章还讨论了如何处理违反并发性的情况，并用具体的例子阐明解决方案。

第 3 部分 ADO.NET 实战

掌握如何访问数据、操纵数据以及更新数据库之后，接着要理解如何让 ADO.NET 与 .NET 框架的其他部分进行交互。

第 9 章 ADO.NET 和 XML

XML 就是数据。ADO.NET 是用于 .NET 的一个数据框架。正是因为这两个事实，所以 ADO.NET 与 XML 框架紧密地集成，使不同的数据库数据能混合使用。本章探讨了如何以通用的方式来对待每一种类型的数据。

第 10 章 用 ADO.NET 进行数据绑定

.NET 有两种基于表单的技术：Windows 窗体和 Web 窗体。幸运的是，ADO.NET 容器（DataReader, DataSet 和 DataTable）都支持直接数据绑定。本章解释了具体的操作过程。

第 11 章 扩展性和性能

本章归纳了一系列有用的设计建议，便于您构建易扩展的、高性能的系统。除此之外，本章还列出了使用 ADO.NET 和数据库开发的一系列常用最佳实践。

附录 A ADO.NET 迁移策略

我们不能想当然地认为所有 .NET 代码都是新开发的。有许多开发者需要将代码迁移到 .NET。本附录针对混合型系统提供了大量可行的策略，便于您让 ADO.NET 数据结构来读取 ADO 结构，或者让 ADO 数据结构来读取 ADO.NET。

目 录

第 1 部分 ADO.NET 基础

第 1 章 ADO.NET 存在的原因	1
1.1 统一数据访问简史	1
1.2 为什么说 ADO.NET 是一个更好的数据访问层	5
1.3 初学 ADO.NET	7
1.3.1 ADO.NET 命名空间	8
1.3.2 ADO.NET 数据结构	9
1.3.3 ADO.NET 托管提供程序对象模型	13
1.4 小结	17
第 2 章 用 ADO.NET 建立连接	19
2.1 第一印象	19
2.2 连接	20
2.2.1 连接字符串	22
2.2.2 理解集成安全性	26
2.2.3 更改数据库	27
2.2.4 连接池	29
2.2.5 连接事件	34
2.2.6 连接工厂	37
2.3 获取 OLE DB 数据库架构信息	40
2.4 .NET 错误处理	41
2.5 小结	45
第 3 章 操纵数据库	47
3.1 命令	47
3.1.1 创建 Command 对象	47
3.1.2 命令类型	48
3.2 执行命令	49
3.2.1 定义的结果集	50
3.2.2 使用参数	51

3.3	ADO.NET 中的数据库事务处理	63
3.3.1	隔离级别	66
3.3.2	SQL Server 事务处理保存点	67
3.3.3	COM+和企业服务	68
3.4	批处理查询	69
3.5	小结	72
第4章	读取数据	73
4.1	读取数据	73
4.2	DataReader	76
4.2.1	构造 DataReader	77
4.2.2	DataReader 的工作原理	78
4.2.3	从 DataReader 中访问数据	78
4.2.4	什么是结果集	85
4.2.5	处理多个结果集	85
4.2.6	处理 DataReader 元数据	86
4.3	综合运用	89
4.3.1	数据库访问	90
4.3.2	数据对象	90
4.3.3	Windows 窗体代码	95
4.4	小结	96

第2部分 DataSet 详述

第5章	构建 DataSet	97
5.1	什么是 DataSet	97
5.1.1	DataSet 的构成	98
5.1.2	DataSet 和托管提供程序	99
5.2	填充 DataSet	99
5.2.1	DataAdapter 简介	99
5.2.2	根据数据库来创建 DataSet	100
5.2.3	根据 XML 来创建 DataSet	106
5.2.4	以程序化方式创建一个 DataSet	108
5.3	定义 DataSet 架构	109
5.3.1	为什么要在 DataSet 中使用数据库架构	109

5.3.2	用 DataAdapter 来推断架构	110
5.3.3	用 XSD 来定义 DataSet 架构	112
5.3.4	以程序化方式创建 DataSet 架构	113
5.3.5	列架构	126
5.4	小结	133
第 6 章	类型化 DataSet	135
6.1	什么是类型化 DataSet	135
6.2	生成类型化 DataSet	138
6.2.1	用 Visual Studio .NET 创建一个类型化 DataSet	139
6.2.2	用 XSD.exe 来创建类型化 DataSet	147
6.2.3	用批注来自定义生成的代码	148
6.3	使用类型化 DataSet	151
6.4	业务对象层的简化	152
6.5	小结	166
第 7 章	操纵 DataSet	167
7.1	更改数据	167
7.1.1	添加行	167
7.1.2	删除行	169
7.1.3	读和写 DataRow 的值	170
7.1.4	行的版本	177
7.1.5	行的状态	179
7.2	DataSet 导航	181
7.2.1	用关系来导航	181
7.2.2	DataView	184
7.3	搜索 DataSet	188
7.3.1	用 DataTable.Select 进行搜索	188
7.3.2	用 DataView 进行搜索	190
7.4	合并 DataSet	191
7.5	小结	193
第 8 章	更新数据库	195
8.1	断开式数据的麻烦	195
8.2	ADO.NET 并发性	196
8.2.1	DataAdapter 如何更新 DataSet	196
8.2.2	实现乐观并发性	197

8.2.3	实现悲观并发性	218
8.2.4	实现破坏式并发性	223
8.3	常见的更新问题	227
8.3.1	处理含有多个表的 DataSet	227
8.3.2	用本地事务处理来更新数据库	230
8.3.3	从 SQL Server 检索新行标识	233
8.4	小结	235

第 3 部分 ADO.NET 实战

第 9 章	ADO.NET 和 XML	237
9.1	.NET 和 XML	237
9.2	DataSet 和 XML	238
9.2.1	将 DataSet 数据转换成 XML	238
9.2.2	将 DataSet 另存为 XML	246
9.2.3	DataSet 命名空间	249
9.2.4	用 XML 来填充 DataSet	249
9.2.5	DiffGram 策略	253
9.2.6	DataSet 架构	255
9.3	XmlDataDocument 类	259
9.3.1	用 XPath 表达式来搜索 DataSet	262
9.3.2	用 XSLT 来转换 DataSet	264
9.4	小结	265
第 10 章	用 ADO.NET 进行数据绑定	267
10.1	.NET 中的数据绑定	267
10.2	Windows 窗体中的数据绑定	267
10.2.1	简单数据绑定	270
10.2.2	复杂绑定	271
10.2.3	DataGrid 绑定	273
10.2.4	主控/详细绑定	276
10.2.5	使用 CurrencyManager 类	278
10.3	ASP.NET 中的数据绑定	280
10.3.1	简单绑定	281
10.3.2	复杂绑定	281

10.3.3 数据控件绑定	282
10.3.4 与 DataReader 绑定	283
10.3.5 性能问题	284
10.4 小结	285
第 11 章 扩展性和性能	287
11.1 是否应该关心	287
11.1.1 设计易于扩展的系统	288
11.1.2 在系统中耦合组件	289
11.2 在 ADO.NET 问世之前	289
11.2.1 连接式数据的问题	289
11.2.2 扩展数据库服务器	290
11.3 ADO.NET 能提供什么帮助	291
11.3.1 在 Web 服务器上进行数据缓存	292
11.3.2 扩展数据库数据	292
11.3.3 实战	295
11.4 DataReader 是否易于扩展	296
11.5 ADO.NET 的性能	296
11.5.1 与数据库交互	297
11.5.2 与 DataSet 交互	297
11.6 最佳实践	297
11.6.1 使用 DataSet 架构	298
11.6.2 使用类型化 DataSet 来创建业务规则层	298
11.6.3 减少到数据库的往返行程	298
11.6.4 提前和经常缓存数据	298
11.6.5 起用 DBA	299
11.6.6 将开发者与数据库隔离	299
11.6.7 在 ASP.NET 中有限度地使用 DataReader	300
11.6.8 使用连接工厂	300
11.6.9 不要硬编码连接字符串	300
11.6.10 将用户排除在数据库之外	301
11.7 小结	301
附录 A ADO.NET 迁移策略	303
A.1 总体规划迁移到 ADO.NET	303
A.1.1 重新结构化 ADO 应用程序	303

A.1.2	ADO.NET 不支持什么	310
A.2	ADO.NET 与 ADO 对象的对应关系	310
A.2.1	将 ADO 映射到 .NET 数据类型	310
A.2.2	提供程序和托管提供程序	311
A.2.3	迁移连接	312
A.2.4	迁移 Command 对象	313
A.2.5	迁移 Recordset	314
A.2.6	在 ADO.NET 中接受 ADO Recordset	316
A.3	小结	316

第 1 部分 ADO.NET 基础

第 1 章 ADO.NET 存在的原因

欢迎进入 ADO.NET (ActiveX Data Objects for .NET) 的世界。对我们而言, ADO.NET 幸好并非只是一个数据访问层, 它还是考虑数据访问的一种更好方式。但在深入 ADO.NET 之前, 首先有必要为大家上一堂简单的历史课。

1.1 统一数据访问简史

为了更清楚地体会 Microsoft 在 ADO.NET 上的设计思路, 您需要了解以前的数据库 API (Application Program Interface, 应用程序编程接口)。在过去 10 年里, Microsoft 多次尝试解决统一数据访问的问题。开发者一直都希望在简化性和性能之间取得平衡。遗憾的是, 这两个方面经常发生冲突。

20 世纪 90 年代初, Microsoft 知道对于自己的开发者来说, 数据库访问是一个大问题, 所以他们制定了一个计划。Microsoft 知道开发者创建的应用程序可能需要与其他平台 (主要是 UNIX 小型机或大型机) 上的数据库进行通信。他们的方案是使用“开放数据库连接” (Open Database Connectivity, ODBC) 应用程序编程接口 (API)。这种 API 允许开发者访问这些后端系统中的数据。

但与此同时, Microsoft 还知道许多开发者正在寻求桌面数据库的解决方案。所以在 1992 年, 他们发布了 Visual Basic 2.0 和 VT Objects, 这使 Visual Basic (VB) 也能访问

ODBC API。VT Objects 是一个非常简单的接口，虽然它允许访问这些后端数据库，但仅此而已。Microsoft 没有揭示出完整的 ODBC API，只揭示了能满足 VB 开发者的数据访问需求的那一部分。

Microsoft 还知道开发者迫切需要采用桌面数据库的形式，将关系数据转变成本地存储。为了支持桌面数据库，Microsoft 发布了 Access 1.0，它包括 Jet（一种自主的数据访问技术）以及一个新版本的 VT Objects，只是现在改名为“数据访问对象”（Data Access Objects, DAO）。这个版本的 DAO 引入了 Connection 和 Recordset 的概念，这些概念一直延用至今。随着技术的进步，DAO 的问题也越来越突出。例如，由于 DAO 不是线程安全的，所以在高负荷或者大型客户机 - 服务器环境中容易失效。

Microsoft 试图用 OLE DB（OLE for Database）来再次改造数据访问。OLE DB 将数据库开发直接与 Microsoft 的组件对象模型（Component Object Model, COM）绑定到一起。OLE DB 本质上是一系列 COM 接口，它揭示了一个消费者/提供者模型，用于在整个企业范围内访问数据。OLE DB 不关心数据具体是在桌面数据库（比如 Access）中，还是在企业数据库（比如 SQL Server 或 Oracle）中，还是在一个除数据库之外的其他数据源（比如 Excel 电子表格）中。如果开发者想将自己的数据揭示给 OLE DB，就需要开发一个 OLE DB 提供程序（提供者）。遗憾的是，事实已经证明，OLE DB 提供程序的开发极其困难，只有最能干的 C++ 开发者才可能胜任。在 OLE DB 中开发数据消费程序（消费者）则要简单得多，但开发者必须精通 Visual C++，这使其失去了“与任何语言无关”这一优势。

之后，Microsoft 又开发了 ActiveX 数据对象（ActiveX Data Object, ADO）。它是围绕 OLE DB 而建立的一个 COM 包装器（wrapper）。ADO 的宗旨是使 OLE DB 更易使用。ADO 发布后不久，Internet 获得爆炸性增长。开发者渴望一种易于使用的 API，使自己的网站和 Web 应用程序能由数据库来驱动。ADO 的脚本编程能力能与 Microsoft 的 Internet 信息服务（IIS）和活动服务器页（ASP）完美地配合。ADO 不久就成为 Internet 网站的“交际语”（lingua franca）。对于小型网站，ADO 的优势是明显的，它很容易理解和编程。但遗憾的是，ADO 不能很好地支持规模较大的网站。大量网页会频繁地访问数据库，一遍又一遍地检索同样的信息，每天可能达到数百万次，这一下子就暴露出 ADO 对于数据库连接的严重依赖。

使用 ADO，您需要创建一个到数据库的连接，然后向数据库查询信息。在程序 1.1 中，您可看到连接具有和查询操作一样的生存期。