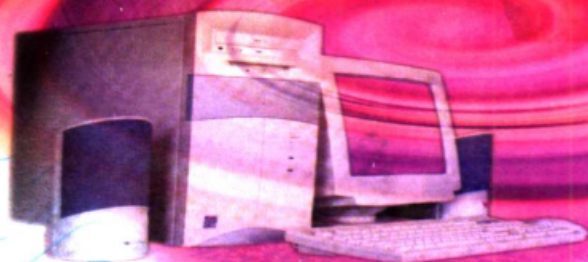


Visual Basic 6.0

高级编程技巧——

ADO数据访问篇

莫卫东 白鹏 张晓敏 张福萍 编著



西安交通大学出版社

Visual Basic 6.0

高级编程技巧

——ADO 数据访问篇

莫卫东 白 鹏 ~~张晓敏~~ 张福萍 编著

西安交通大学出版社

中国·西安

内 容 简 介

Visual Basic 6.0 是微软公司最新推出的 Visual Studio 6.0 可视化应用程序开发工具组件中的一员,也是当今世界上最流行的可视化编程工具。本书是“Visual Basic 6.0 高级编程技巧系列”丛书中的一本——ADO 数据访问篇。主要介绍在 Visual Basic 6.0 环境下,使用先进的 ADO 技术进行数据访问的程序设计和方法。除了讲解具有针对性的实用方法之外,并在其中插入了一些全新的实例,在这些实例中有许多是作者在实际设计中采用过的。

本书立意独特,体系完整,内容最新,图文并茂,使用最新的实例,讲解最实用的方法,奉献最有价值的技巧,使读者能在最短的时间内掌握使用 Visual Basic 6.0,开发出最好的数据访问应用程序的方法和技巧。

本书不仅适用于 Visual Basic 的初学者作为学习和开发 Windows 应用程序的理想参考书,更适用于所有软件开发人员作为常备手册性参考书。

图书在版编目(CIP)数据

Visual Basic 6.0 高级编程技巧.ADO 数据访问篇/
莫卫东编著.-西安:西安交通大学出版社,2000.1
ISBN 7-5605-1226-7

I.V… II.莫… III.BASIC 语言-程序设计
IV.TP312

中国版本图书馆 CIP 数据核字(2000)第 00668 号

*

西安交通大学出版社出版发行

(西安市咸宁西路 28 号 邮政编码:710049 电话:(029)2668316)

西安市德力彩印厂印装

各地新华书店经销

*

开本:787 mm×1 092 mm 1/16 印张:19.375 字数:468 千字

2000 年 1 月第 1 版 2000 年 1 月第 1 次印刷

印数:0 001~5 000 定价:28.00 元

若发现本社图书有倒页、白页、少页及影响阅读的质量问题,请去当地销售
部门调换或与我社发行科联系调换。发行科电话:(029)2668357,2667874

前 言

Visual Basic 6.0 是微软公司最新推出的 Visual Studio 6.0 可视化应用程序开发工具组件中的一员，也是当今世界上最流行的可视化编程工具。新的 Visual Basic 系统不但简化了可视化编程的方法，而且使得应用程序的开发流程更加流畅，功能也更加强大。特别是 Visual Basic 6.0 在数据库和 Internet 应用程序开发两方面的改进，使程序员能够在当前的环境中轻松自如地开发出功能不凡的各种数据库与多媒体应用程序。因此，Visual Basic 6.0 已经成为最受欢迎的 Windows 应用程序可视化开发工具。

为了帮助众多的程序设计人员尽快掌握 Visual Basic 6.0 程序设计的方法，有关 Visual Basic 6.0 的各类技术参考书不断问世。在这些参考书中不乏优秀之作，但更多地是介绍 Visual Basic 6.0 进行程序设计的基本方法和技巧，属于入门级的技术参考书。我们的这套“Visual Basic 6.0 高级编程技巧”丛书，试图从 Visual Basic 编程的各个角度，分别系统全面地介绍 Visual Basic 编程方法，更深入地分析编程中的策略与技巧，并通过大量的实例来讲解各种方法与技巧的实际应用，具有很强的实战效果。其中的许多示例都是作者在实际工作中采用过的。可见，本丛书是介绍 Visual Basic 编程方法与技巧的更高层次的技术参考书，广大的 Visual Basic 编程人员一定会在其中获得有益的帮助和启示。

这套丛书共有 4 本，分别以四个专题介绍 Visual Basic 程序设计中的更高层次的方法与技巧。它们是：

- 窗口、菜单篇
- 控件篇
- 多媒体通信篇
- ADO 数据访问篇

本书是该丛书之一，即《Visual Basic 6.0 高级编程技巧——ADO 数据访问篇》。全书立意独特，体系完整，内容充实，图文并茂，使用最新的实例，讲解最实用的方法，奉献最有价值的技巧。使读者能在最短的时间内掌握使用 Visual Basic 6.0 编程方法与技巧，以开发出最好的数据访问方面的应用程序。本书共有 7 章，具体的内容安排如下：

第 1 章 Visual Basic 数据访问原理概述

Visual Basic 6.0 所支持的 ADO 数据访问接口，可帮助我们对关系型数据库和操作系统中的文件、电子表格、电子邮件、多媒体数据以及目录服务信息访问。本章要点：Visual Basic 数据访问技术的基本框架，Microsoft 的一致数据访问技术（UDA），系统级数据访问接口——OLE DB，应用级数据访问接口——ADO，ADO 与 RAO 和 RDO 之间的关系，Visual Basic 6.0 有关数据访问的新内容。

第 2 章 Visual Basic 中的 ADO 数据访问原理

Visual Basic 6.0 最鲜明的变化就是缺省地采用了 ADO 作为它与各种数据源的接口，

从而使 Visual Basic 在数据访问技术方面有了新的突破，为 Visual Basic 更广泛的应用奠定了十分坚实的基础。ADO 数据访问方式是现在和未来 Visual Basic 乃至 Microsoft 的各种应用软件进行数据访问与应用的主流。因此，这里将最大限度地重点介绍 ADO 数据访问技术的原理、特点和 ADO 在 Visual Basic 编程应用中的方法与技巧。作者认为，ADO 应用是目前 Visual Basic 应用与开发的最前沿。

第 3 章 Visual Basic ADO 数据访问程序设计方法

本章系统地介绍了 ADO 数据访问原理，其中有许多的范例。但是，在 Visual Basic 6.0 又该如何具体地实现数据访问的应用呢？是不是像范例中所示范的那样，在 Visual Basic 中应用 ADO 数据访问接口模型进行数据库的访问，要求设计者一定要通过编程才能到达目的？如果是这样，那不是有回到了非 Visual Basic 程序设计了吗？那么，如何使用 Visual Basic 进行有关数据访问应用的程序设计，正是本章所要解决的问题，这就是 ADO Data 控件与数据环境等。

第 4 章 ADO 数据访问程序设计工具的应用方法与技巧

在 Visual Basic 6.0 系统中特别配备了一些用于 ADO 数据访问程序设计的工具。这些工具可以为在 Visual Basic 6.0 中进行数据访问应用系统的设计带来许多的方便，从而大大提高程序设计的生产效率。本章介绍 Visual Basic 6.0 中提供的有关数据访问系统设计工具的应用方法与技巧，以帮助读者能在最短的时间内设计出最好的数据访问应用。

第 5 章 ADO 数据访问中的数据库操作方法与技巧

ADO 数据访问不仅可访问各种数据库信息，也可以访问其他非数据库类型的数据源。但最常用的仍是对数据库访问应用，这一问题也是众多 Visual Basic 应用程序开发者经常需要解决的。本章系统全面地讲解使用 ADO 对象方法和 ADO Data 控件方法进行数据库访问的方法，以及 SQL 在 Visual Basic 6.0 数据访问程序设计中应用的有关问题。

第 6 章 Visual Basic 6.0 中的数据报表输出技术

数据的显示与打印输出几乎是所有数据访问应用系统必须的，其中数据报表的打印输出功能在许多应用中具有十分突出的地位和作用。因此，在数据访问应用系统中，有关报表打印输出模块的设计是必不可少的，也使之成为体现一个数据访问程序设计水平高低的一个重要标准。本章全面介绍 Visual Basic 6.0 最新的数据报表设计器和其他数据报表输出技术的特点以及应用方法与技巧。

第 7 章 ADO 数据访问应用实例

本章把作者在实际工作中的完成的 ADO 数据访问应用系统，从系统的规划到设计介绍给读者，期望能为读者提供一个完整的 ADO 数据访问程序设计的范例，从而能全面地掌握 Visual Basic 6.0 的 ADO 数据访问程序的方法与技巧。

作者

1999. 12

目 录

第 1 章 Visual Basic 数据访问原理概述

1.1 Visual Basic 6.0 数据访问技术的基本框架	1
1.1.1 Visual Basic 数据访问技术的发展	2
1.1.2 DAO 与 Microsoft Jet	2
1.1.3 RDO 与 ODBC 数据库驱动接口	3
1.1.4 ADO 与 OLE DB	4
1.2 Microsoft 一致数据访问技术	5
1.3 系统级数据访问接口 —— OLE DB	6
1.4 应用层数据访问接口 —— ADO	7
1.5 Visual Basic 6.0 数据访问策略	8
1.6 Visual Basic 6.0 有关数据访问技术的新内容	8

第 2 章 Visual Basic 中的 ADO 数据访问原理

2.1 再论 ADO 与 OLE DB	11
2.1.1 ADO/OLE DB 数据访问的特点	11
2.1.2 ADO/OLE DB 的基本应用方法	13
2.1.3 OLE DB 标准中定义的新概念	13
2.1.4 OLE DB 与 ODBC 的关系	14
2.2 OLE DB 数据访问对象模型	14
2.3 ADO 数据访问对象之一 —— Connection 对象	15
2.3.1 Connection 对象的基本功能	16
2.3.2 Connection 对象的基本属性	16
2.3.3 Connection 对象的基本方法	20
2.4 ADO 数据访问对象之二 —— Command 对象	24
2.4.1 Command 对象的基本性能	24
2.4.2 Command 对象的基本属性	25
2.4.3 Command 对象的基本方法	27
2.5 ADO 数据访问对象之三 —— Recordset 对象	29
2.5.1 Recordset 对象的基本性能	29
2.5.2 Recordset 对象应用的例子	31
2.5.3 Recordset 对象的基本属性	32
2.5.4 Recordset 对象的基本方法	40
2.6 ADO 数据访问对象之四 —— Error 对象	51
2.6.1 Error 对象的基本功能	51
2.6.2 Error 对象的基本属性	52
2.6.3 有关 Error 对象属性应用的范例	53

2.7 ADO 数据访问对象之五 —— Field 对象	55
2.7.1 Field 对象的基本功能	55
2.7.2 Field 对象的属性详解	56
2.7.3 Field 对象的基本方法详解	58
2.8 ADO 数据访问对象之六 —— Parameter 对象	60
2.8.1 Parameter 对象的基本功能	60
2.8.2 Parameter 对象的基本属性	60
2.8.3 Parameter 对象的基本方法	62
2.9 ADO 数据访问对象之七 ——Property 对象	63
2.9.1 Property 对象的基本功能	63
2.9.2 Property 对象的基本属性	64
2.10 ADO 集合对象	65
2.10.1 Errors 集合对象的基本功能	65
2.10.2 Errors 集合对象的基本属性	66
2.10.3 Errors 集合对象的基本方法	66
2.10.4 Parameters 集合对象的基本功能	67
2.10.5 Parameters 集合对象的基本属性	68
2.10.6 Parameters 集合对象的基本方法	68
2.10.7 Properties 集合对象的基本功能	70
2.10.8 Properties 集合对象的基本属性	70
2.10.9 Properties 集合对象的基本方法	70
2.10.10 Fields 集合对象的基本功能	71
2.10.11 Fields 集合对象的基本属性	71
2.10.12 Fields 集合对象的基本方法	72
2.11 ADO 对象事件	73
2.11.1 ADO 事件类型以及在应用中注意的问题	73
2.11.2 ADO 事件过程分析	74
2.11.3 ADO 事件参数点评	76
2.11.4 ADO 事件详解	78
2.12 ADO 的高级特性	87

第3章 Visual Basic ADO 数据访问程序设计方法

3.1 Visual Basic 6.0 数据访问的基本方案	92
3.1.1 本地数据访问的解决方案	93
3.1.2 远程数据访问的解决方案	95
3.2 在 Visual Basic 6.0 创建一个数据访问应用的基本步骤	97
3.2.1 步骤1: 连接数据源	97
3.2.2 步骤2: 创建SQL查询命令对象	98
3.2.3 步骤3: 执行命令	99

3.2.4 步骤4: 处理数据.....	100
3.2.5 步骤5: 更新数据源.....	101
3.2.6 步骤6: 结束更新数据源.....	102
3.3 Visual Basic 6.0 中的 ADO 数据访问控件及其应用.....	102
3.3.1 ADO Data 控件的概述.....	103
3.3.2 ADO Data 控件的用途.....	103
3.3.3 将 ADO Data 等其它 ADO 数据访问控件加入到工具箱.....	104
3.3.4 使用 ADO Data 控件创建一个前端数据库应用程序.....	105
3.3.5 在程序中设置 ADO Data 等其它数据绑定控件的属性.....	114
3.3.6 ADO Data 控件与数据访问有关的属性详解.....	114
3.3.7 ADO Data 控件事件速查.....	121
3.4 ADO 数据绑定控件——DataList 和 DataCombo 控件及其应用.....	122
3.4.1 DataList 和 DataCombo 控件有关数据访问的属性详解.....	122
3.4.2 DataList 控件功能与应用方法.....	127
3.4.3 DataCombo 控件的功能及其应用方法.....	129
3.4.4 DataGrid 控件的功能及其应用.....	131
3.5 MSFlexGrid 和 MSHFlexGrid 控件.....	139
3.5.1 MSFlexGrid 和 MSHFlexGrid 控件的功能比较.....	139
3.5.2 MSHFlexGrid 控件与 DataGrid 控件选用的原则.....	140
3.5.3 MSHFlexGrid 控件的应用方法详解.....	140
3.5.4 MSHFlexGrid 控件的基本应用技术——分层与带区.....	148
3.5.5 MSHFlexGrid 控件应用的高级技巧——表头的设置.....	152
3.5.6 MSHFlexGrid 控件应用的特别技巧——数据的合并与排序.....	153
3.6 连接不同ADO数据源的技术分析与范例.....	156
3.6.1 连接 ADO 数据源的几种途径.....	156
3.6.2 与 Microsoft Jet/Microsoft Access 的数据交互技术与范例.....	157
3.6.3 与 SQL Server 的数据交互技术与范例.....	158
3.6.4 与 ASCII 文本文件中的数据交互技术与范例.....	165

第4章 ADO 数据访问程序设计工具的应用方法与技巧

4.1 Visual Basic 6.0 数据访问设计工具概述.....	174
4.2 “数据视图”窗口与数据库工具.....	175
4.1.1 表、视图、存储过程、触发器.....	176
4.1.2 在“数据视图”中管理数据库.....	178
4.3 数据环境设计器.....	184
4.3.1 数据环境设计器的功能与特点.....	184
4.3.2 数据环境的设计.....	185
4.3.3 数据环境的操作.....	193
4.3.4 在编程中数据环境的应用.....	197
4.4 可视化数据库管理.....	198

4.4.1 数据库设计器的功能与特点.....	198
4.4.2 创建和操作数据库图表.....	199
4.4.3 创建和添加数据库表.....	200
4.4.4 建立和管理数据库表间的关系.....	201
4.4.5 管理数据库表的索引和约束.....	204
4.4.6 管理数据库表的列.....	206
4.5 查询设计器.....	209
4.5.1 查询设计器的功能与特点.....	209
4.5.2 查询设计器操作界面.....	210
4.5.3 创建查询的步骤与方法.....	211
4.5.4 查询设计器操作与应用详解.....	212

第5章 ADO 数据访问中的数据库操作方法与技巧

5.1 通过数据访问对象对数据库的访问.....	225
5.1.1 打开数据库.....	225
5.1.2 定位记录.....	228
5.1.3 筛选、索引和排序记录.....	232
5.1.4 添加、编辑和删除记录.....	234
5.1.5 访问多个记录.....	241
5.1.6 输出记录.....	242
5.2 结构化查询语言 SQL 的基础与应用.....	243
5.2.1 SQL 的概念与原理.....	243
5.2.2 SELECT 语句的应用.....	243
5.2.3 DELETE 语句的应用.....	247
5.2.4 INSERT 语句的应用.....	248
5.2.5 UPDATE 语句的应用.....	248
5.3 Visual Basic 6.0 中的 SQL 应用方法与技巧.....	248
5.3.1 如何在 Visual Basic 编程中使用 SQL 语句.....	248
5.3.2 Command 对象与 SQL 语句.....	249
5.3.3 Recordset 对象与 SQL 语句.....	250
5.3.4 快速编写 SQL 语句 —— SQL 生成器的应用.....	251

第6章 Visual Basic 6.0 中的数据报表打印输出技术

6.1 Visual Basic 6.0 报表打印输出设计概述.....	253
6.2 编程直接打印输出设计的方法、技巧和实例.....	254
6.2.1 通过编程打印输出技术特点和应用方法.....	255
6.2.2 编程打印输出设计的实例点评.....	256
6.3 通过 MS Office 实现报表打印输出技术.....	259
6.3.1 向 Excel 传递数据并打印报表.....	259
6.3.2 向 Word 传递数据并打印报表.....	261
6.4 应用数据报表设计器的报表打印输出技术.....	263

6.4.1 数据报表设计器的功能与特点	263
6.4.2 数据报表设计器设计报表的基本方法	264
6.4.3 输出由数据报表设计器设计的报表	266
6.4.4 数据报表对象、方法、控件与快捷菜单	267
6.4.5 数据报表设计高级技术与技巧	270

第7章 ADO 数据访问应用实例

7.1 实例的功能以及设计思想	278
7.1.1 系统功能规划	278
7.1.2 系统设计的技术分析	281
7.2 设计方法与技巧点评之一 —— “处方划价系统”	282
7.2.1 系统功能分析	282
7.2.2 设计方法点评	284
7.3 设计方法与技巧点评之二 —— “入库管理系统”	298
7.3.1 系统功能分析	298
7.3.2 设计方法点评	299

第 1 章

Visual Basic 数据访问 原理概述

Visual Basic 采用事件驱动编程，以其优异的面向对象程序设计的特性赢得了广大程序设计人员的青睐。新的 Visual Basic 程序设计系统，不但具有更强大的功能和丰富的编程素材，对数据库的操作与应用更加趋于完善，为 Visual Basic 的程序开发提供了更加广阔的应用空间。特别是在信息高度发展的今天，任何一种应用都离不开对数据的访问，而数据的访问的主要方法是对数据库的操作，当然还包括对其他包含有各类信息数据的文件的操作。Visual Basic 6.0 所支持的 ADO 数据访问接口，可帮助我们对关系型数据库和操作系统中的文件、电子表格、电子邮件、多媒体数据以及目录服务信息的访问。

本章要点：

- Visual Basic 6.0 数据访问技术的基本框架
- Microsoft 一致数据访问技术
- 系统级数据访问接口 —— OLE DB
- 应用层数据访问接口 —— ADO
- Visual Basic 6.0 数据访问策略
- Visual Basic 6.0 有关数据访问技术的新内容

1.1 Visual Basic 6.0 数据访问技术的基本框架

为了掌握 Visual Basic 6.0 进行数据访问的技术，我们需要对 Visual Basic 6.0 所支持的各种数据访问技术有全面的了解，也就是说不但要对 Visual Basic 6.0 的数据访问技术的基本框架有一定的概念，主要是还能够对框架中的各种数据访问技术的特点有准确的理解，只有这样，我们才能在实际的应用程序设计中选用最恰当的数据访问方法。

1.1.1 Visual Basic 数据访问技术的发展

Visual Basic 在发表之初就支持了与数据库的连接,在 Visual Basic 5.0 之前,Visual Basic 一直采用 Microsoft Jet 引擎作为与数据库连接的纽带, Visual Basic 通过一系列的数据访问对象 DAO (Data Access Object) 来访问 Microsoft Jet 引擎的功能, 让用户能创建新的数据库、表、查询、索引和关系等等。此后又推出了新的数据库接口技术 RDO —— 远程数据对象, 使用户对数据的访问更加简单和迅速; 而新的 Visual Basic 6.0 使用了最新的数据访问技术 ADO (ActiveX Data Object) —— ActiveX 数据对象, 使 Visual Basic 对数据的访问技术上升了一个新的高度, 不但使 Visual Basic 对数据的访问更加简单和灵活, 而且访问的数据类型更加丰富, 特别在 Internet 和 Intranet 方面的应用可极大地提高系统的性能。

一般认为, Visual Basic 数据访问技术到今天 Visual Basic 6.0 经历了 RAO 到 RDO, 以及现在的 ADO 三个重要的阶段。但是, 为了使 Visual Basic 数据库的开发具有向后的兼容性和工程的可维护性, Visual Basic 6.0 仍然支持数据访问对象 DAO 和远程数据对象 RDO, 从而可见, DAO、RDO 和 ADO 构成了现在 Visual Basic 数据访问的基本框架。

1.1.2 DAO 与 Microsoft Jet

DAO (Data Access Objects) 数据访问对象是 Visual Basic 最先采用的面向对象的数据数据库访问接口。Visual Basic 正是通过一系列的 DAO 对象来访问 Microsoft Jet 数据库引擎 (Microsoft Jet DataBase Engine), 使得用户能方便快速地创建新的数据库、表、查询、索引和关系, 并对数据库进行修改和数据的编辑等。其基本思路是通过设置 DAO 对象的属性和附加到这些对象中的方法让用户来操作数据库中的信息, 包括对数据库中的数据以及记录进行添加、编辑和删除。

Microsoft Jet 引擎是 Visual Basic 与数据库连接的中间层, 它为 Visual Basic 提供了进行数据库访问的基本方法。Microsoft Jet 是 Microsoft 公司为其产品开发的一个共同的数据库接口模块, 不但 Visual Basic 可以使用, Microsoft Access 等其他产品也使用此 Jet 引擎与数据库进行连接。Jet 引擎包含在一组动态连接库 (DLL) 文件中, 在运行时, 这些文件被连接到 Visual Basic 程序中。Jet 引擎负责把 Visual Basic 应用程序通过 DAO 对象对数据库访问的请求翻译成对 Access 数据库 (.mdb) 文件或其他数据库的物理操作, 即对数据库进行真正的读取、写入和修改, 并处理有关数据库的所有其他的事物, 如索引、锁定、安全性和引用完整性等。

除此之外, Jet 引擎还提供了多种功能, 最大限度地支持 Visual Basic 对数据库的访问。如 Jet 引擎支持 SQL, 使用户可以在 Visual Basic 使用 SQL 语句对数据库进行标准的数据库查询、检索、添加、删除和更新; Jet 引擎同时还提供有安全功能, 以便应用程序的开发者在进行数据库访问时让用户输入用户标识和口令; Jet 引擎也支持使用许可功能以及对单个表和查询的访问级别限制的功能, 这样便可在应用程序中方便地对用户或用户组设置不同的访问限制。

把 DAO 与 Microsoft Jet 的关系用图表示, 可更清楚地看出它们之间的关系, 并可

更好地理解 Visual Basic 进行数据访问的基本原理。图 1.1 是 Visual Basic、DAO、Microsoft Jet 和数据库之间的关系图。

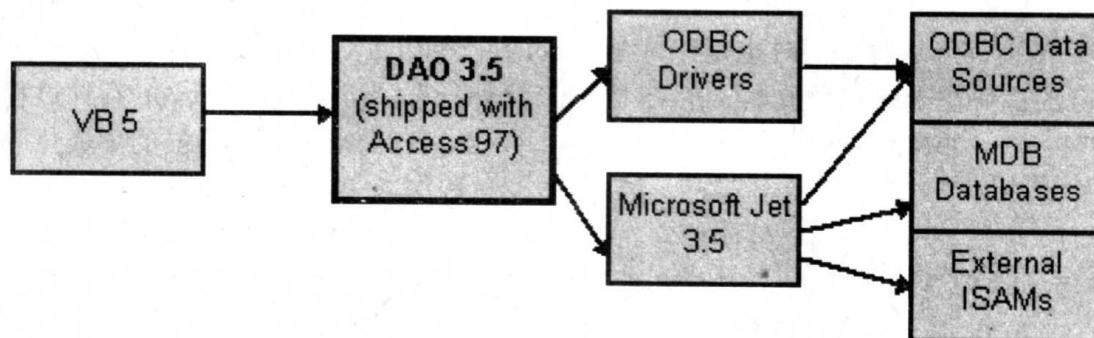


图 1.1 DAO 与 Microsoft Jet 数据访问图

可见，DAO 接口借用 Microsoft Jet，使 Visual Basic 开发者如同直接连接到其他数据库一样，直接连接到 Access 表或其他数据库。DAO 最适用于单系统应用程序或小范围本地分布使用。

1.1.3 RDO 与 ODBC 数据库驱动接口

RDO (Remote Data Objects) 远程数据对象是一个到 ODBC 的面向对象的数据访问接口，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 (顺序索引查找方法) 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、属性和方法。

RDO 是位于 ODBC API 之上的一个对象模型薄层，它依赖 ODBC API、选定的 ODBC 驱动程序以及后端数据库引擎实现大部分的智能和功能，因此短小 (大约 250 K)、快速、强健。RDO 具备基本的 ODBC 处理方法，所以可以直接执行大多数 ODBC API 函数。RDO 包含在 Visual Basic 4.0 之后的各个版本中，它由一个动态连接库 MSRD032.DLL 实现。

当 Visual Basic 版本还只是 2.0 时，从 VB 访问 MS SQL Server 的手段只有 API 函数集 (VBSQL/DBLib 或 ODBC API)。Visual Basic 发展到 3.0 时，其 DAO/Jet (Microsoft Access 的数据库引擎) 功能非凡，具备面向对象的程序接口，但处理速度尚不尽如人意。另一方面，尽管 VBSQL/DBLib、ODBC API 在速度方面性能很好，但它需要花费很大的精力进行繁琐的编程工作。当 MS SQL Server 发展到 6.0、Visual Basic 发展到 4.0 时，开发了新的对象模块，由此产生了 RDO，可以访问 SQL Server、Oracle；RDO 2.0 则是由 VB5.0 提供的。值得一提的是，它实现了“事件驱动型数据库访问的编程方式”。

要讨论 RDO，就必然要谈到 DAO。DAO/Jet 是为了实现从 VB 访问 Access 数据库而开发的程序接口 (对象)。RDO 是从 DAO 派生出来的，但两者很大的不同在于其数据库模式。DAO 是针对 [记录 (Records)] 和 [字段 (Fields)]，而 RDO 是作为 [行 (Rows)] 和 [列 (Columns)]

来处理。也就是说 DAO 是 ISAM 模式, RDO 是关系模式。此外 DAO 是访问 Access 的 Jet 引擎(Jet 是 ISAM)的接口, 而 RDO 则是访问 ODBC 的接口。

可见, RDO 是综合了 DAO/Jet、VBSQL/DBLib 以及 ODBC 的优点的对象(Object)。需要强调的是, RDO 是包裹着 ODBC API 的一层薄薄的外壳, 被设计成在后台(服务器端)有数据库存在的前提下运行, 同时也是针对 SQL Server 和 Oracle 而特别设计的。

RDO 的优势在于它完全被集成在 VB 之中。此外, 直接访问 SQL Server 存储过程、完全支持 T-SQL、T-SQL 调试集成在开发环境中、Visual Database Tools 的集成化等, 也是 RDO 的长处。

然而, Microsoft 公司已经明确宣布今后不会对 VBSQL/DBLib 进行升级, 而 ODBC API 函数一级的编程方式也不为人们所喜爱。RDO 自身今后将被以 ActiveX 技术为基础的 ADO(ActiveX Data Objects)所替代。Visual Basic 6.0 的发表已经表明, ADO 将开始替代 RDO 成为 Visual Basic 进行数据访问的接口。

为什么需要 ADO 呢? RDO 是以 ODBC 为基础, 而 ADO 则是基于全新的 OLE DB 技术。ODBC 本身是以 SQL Server、Oracle 等关系数据库作为访问对象; OLE DB 则不仅限于此, 而将是对电子邮件、文本文件、复合文件、数据表等各种各样的数据通过统一的接口进行存取的技术。DAO、RDO 当然不会一下子消失, 但随着新技术的利用, 它们的作用将逐步淡化。

1.1.4 ADO 与 OLE DB

ADO (ActiveX Data Object) —— ActiveX 数据对象是 DAO/RDO 的后继产物, 也是 Visual Basic 6.0 开始最新支持的数据访问对象。最新的 ADO 2.0 在功能上与 RDO 更相似, 而且一般来说, 在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)和事件。例如, ADO 没有与 rdoEngine 和 rdoEnvironment 对象相等同的对象, 便可以显露 ODBC 驱动程序管理器和 hEnv 接口。

ADO 是为 Microsoft 新的数据访问接口 OLE DB 而设计的, 是一个便于使用的应用程序层的数据访问接口。ADO 通过 OLE DB 所提供的高性能的数据访问方法, 不仅可方便快速地对各种数据源进行操作访问, 而且可使之成为一个高性能的轻量接口。ADO、OLE DB 可访问的数据源包括关系或非关系数据库、电子邮件和文件系统、文本和图形、自定义业务对象等等。

不过, 也应当注意, ADO 主要包括 RDO 风格的功能性, 以便和 OLE DB 数据源交互, 另外还包括远程和 DHTML 技术。但 ADO 当前并不支持 DAO 的所有功能。所以, 要把已有的基于 DAO 的 Visual Basic 应用程序直接移植到 ADO 是有一定的困难, 通常只有重写代码。

由于 ADO 代表了未来 Visual Basic 数据访问技术的发展, 所以本书重点介绍 ADO 数据访问技术的性能、特点和应用。关于 ADO 与 OLE DB 的详细介绍参见下面几节的内容。

1.2 Microsoft 一致数据访问技术

Microsoft 新近推出的 UDA(Universal Data Access, 一致数据访问技术)为关系型或非关系型数据访问提供了一致的访问接口,为企业级 Intranet 应用多层软件结构提供了数据接口标准。一致数据访问包括两层软件接口,分别为 ADO(ActiveX Data Object)和 OLE DB,对应于不同层次的应用开发,ADO 提供了高层软件接口,可在各种脚本语言(Script)或一些宏语言中直接使用;OLE DB 提供了底层软件接口,可在 C/C++语言中直接使用。ADO 以 OLE DB 为基础,它对 OLE DB 进行了封装。一致数据访问技术建立在 Microsoft 的 COM(Component Object Model, 组件对象模型)基础上,它包括一组 COM 组件程序,组件与组件之间或者组件与客户程序之间通过标准的 COM 接口进行通讯。

随着网络技术和数据库技术的不断发展,现在的应用系统对数据集成的要求越来越高,这些数据有可能分布在不同的地方,并且使用不同的格式,例如关系型数据库和操作系统中的文件、电子表格、电子邮件、多媒体数据以及目录服务信息等等。传统的解决方案是使用大型的数据库系统,把所有这些数据都移到数据库系统中,然后按照操作数据库的办法对这些数据进行访问,这样做虽然能够按统一的方式对数据进行各种操作,但这种间接访问方式带来了很多问题,比如数据更新不及时、空间资源的冗余和访问效率低等等。

Microsoft 公司推出的一致数据访问技术则较好地解决了这些问题,它使得应用通过一致的接口来访问各种各样的数据,而不管数据驻留在何处,也不需要数据进行转移或复制、转换,在实现分布式的同时也带来了高效率。并且 UDA 技术在统一数据访问接口的同时,它的多层结构使数据使用方有了更多的选择机会,而它强大的扩展能力也给数据提供方留下了更多的扩展余地,这种开放型的软件结构使它具有极强的生命力,所以,这种技术从一推出便获得了广泛的欢迎,可以说,UDA 技术是继 ODBC 之后的又一数据访问技术的飞跃。

UDA 技术包括 OLE DB 和 ADO 两层标准接口,OLE DB 是系统级的编程接口,它定义了一组 COM 接口,这组接口封装了各种数据系统的访问操作,这组接口为数据使用方和数据提供方建立了标准,OLE DB 还提供了一组标准的服务组件,用于提供查询、缓存、数据更新、事务处理等操作,因此,数据提供方只需实现一些简单的数据操作,在使用方就可以获得全部的数据控制能力。

ADO 是应用层的编程接口,它通过 OLE DB 提供的 COM 接口访问数据,它适合于各种客户机/服务器应用系统和基于 Web 的应用,尤其在一些脚本语言中访问数据库操作是 ADO 的主要优势。ADO 是一套用自动化技术建立起来的对象层次结构,它比其他的一些对象模型如 DAO(Data Access Object)、RDO(Remote Data Object)等具有更好的灵活性,使用更为方便,并且访问数据的效率更高。图 1.2 显示了统一数据访问的软件层次模型。

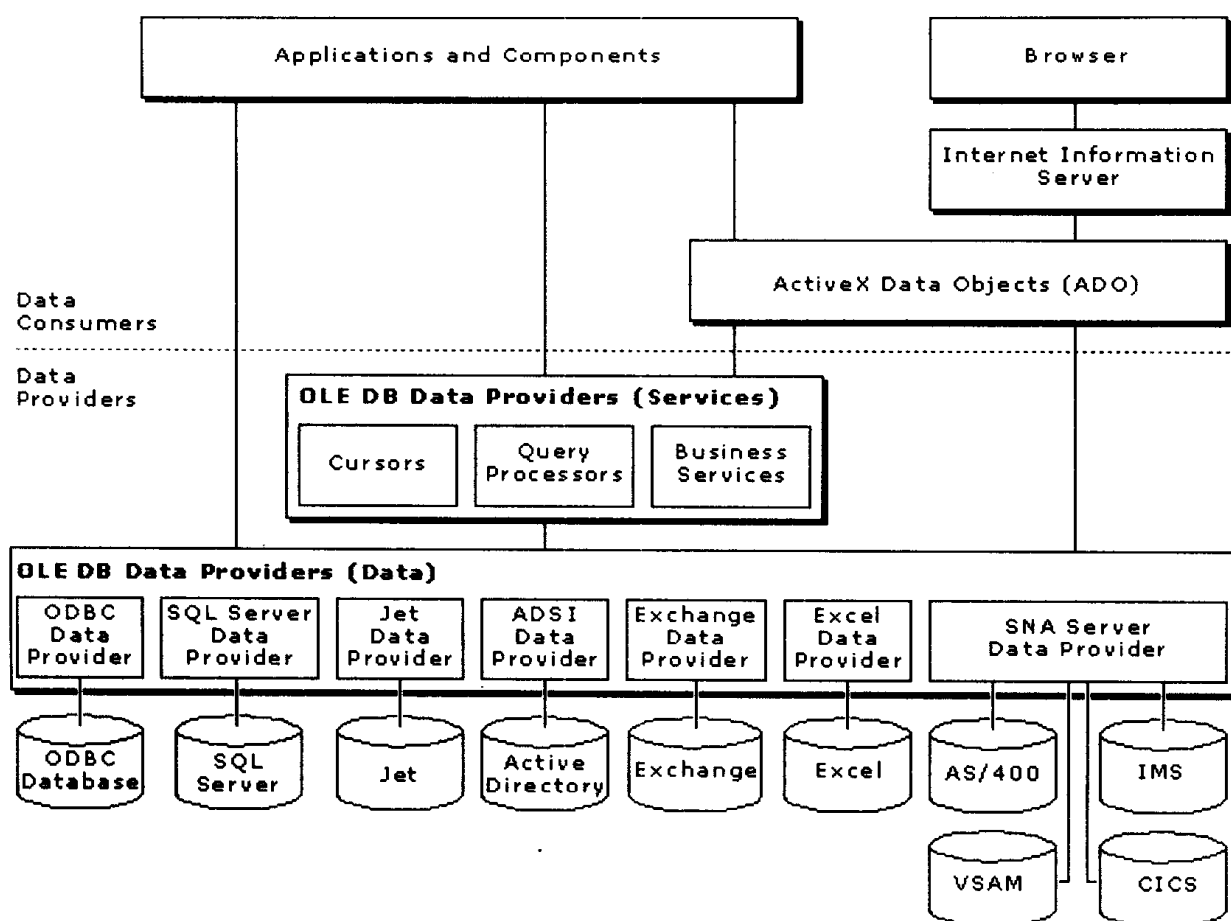


图 1.2 UDA 的层次结构图

1.3 系统级数据访问接口 —— OLE DB

OLE DB 是 Microsoft 开发的新一代数据访问接口，它是现在和未来 Microsoft 产品数据访问应用的重要的接口技术之一，它与 ADO 的结合将为 Intranet 和 Intranet 的网络应用创造新的机会和更广阔的应用前景。

简单地说，OLE DB 是一种技术标准，目的是提供一种统一的数据访问接口，这里所说的“数据”，除了标准的关系型数据库中的数据之外，还包括邮件数据、Web 上的文本或图形、目录服务(Directory Services)，以及主机系统中的 IMS 和 VSAM 数据。OLE DB 标准的核心内容就是要求对以上这些各种各样的数据存储(Data Store)都提供一种相同的访问接口，使得数据的使用者(应用程序)可以使用同样的方法访问各种数据，而不用考虑数据的具体存储地点、格式或类型。

OLE DB 标准的具体实现是一组 C++ API 函数，就像 ODBC 标准中的 ODBC API 一样。不同的是，OLE DB 的 API 是符合 COM 标准、基于对象的 API，ODBC API 则是简单的 C API。使用 OLE DB API，可以编写能够访问符合 OLE DB 标准的任何数据源的应用程序，也可以编写针对某种特定数据存储的查询处理程序(Query Processor)和游标引擎(Cursor

Engine), 因此 OLE DB 标准实际上是规定了数据使用者和提供者之间的一种应用层的协议。

最新发布的微软最新数据库产品 SQL Server 7.0 的查询处理程序就是使用 OLE DB API 编写的, 所以在 SQL Server 7.0 的 Transact-SQL 语法中, 可以支持异种数据库的查询, 当然, 这些异种数据库必须也提供相应的 OLE DB 访问接口。

1.4 应用层数据访问接口 —— ADO

ADO 最早被用于 Microsoft Internet Information Server 中访问数据库的接口, 与一般的数据库接口相比, ADO 可更好地用于网络环境, 通过优化技术, 它尽可能地降低网络流量; ADO 的另一个特性是使用简单, 不仅因为它是一个面向高级用户的数据库接口, 更因为它使用了一组简化的接口用以处理各种数据源。这两个特性使得 ADO 必将取代 RDO 和 DAO, 成为最终的应用层数据接口标准。

参见图 1.3 的 ADO 与 OLE DB 的关系, 可以看到了 ADO 实际上是 OLE DB 的应用层接口, 这种结构也为一致的数据访问接口提供了很好的扩展性, 而不再局限于特定的数据源, 因此, ADO 可以处理各种 OLE DB 支持的数据源。

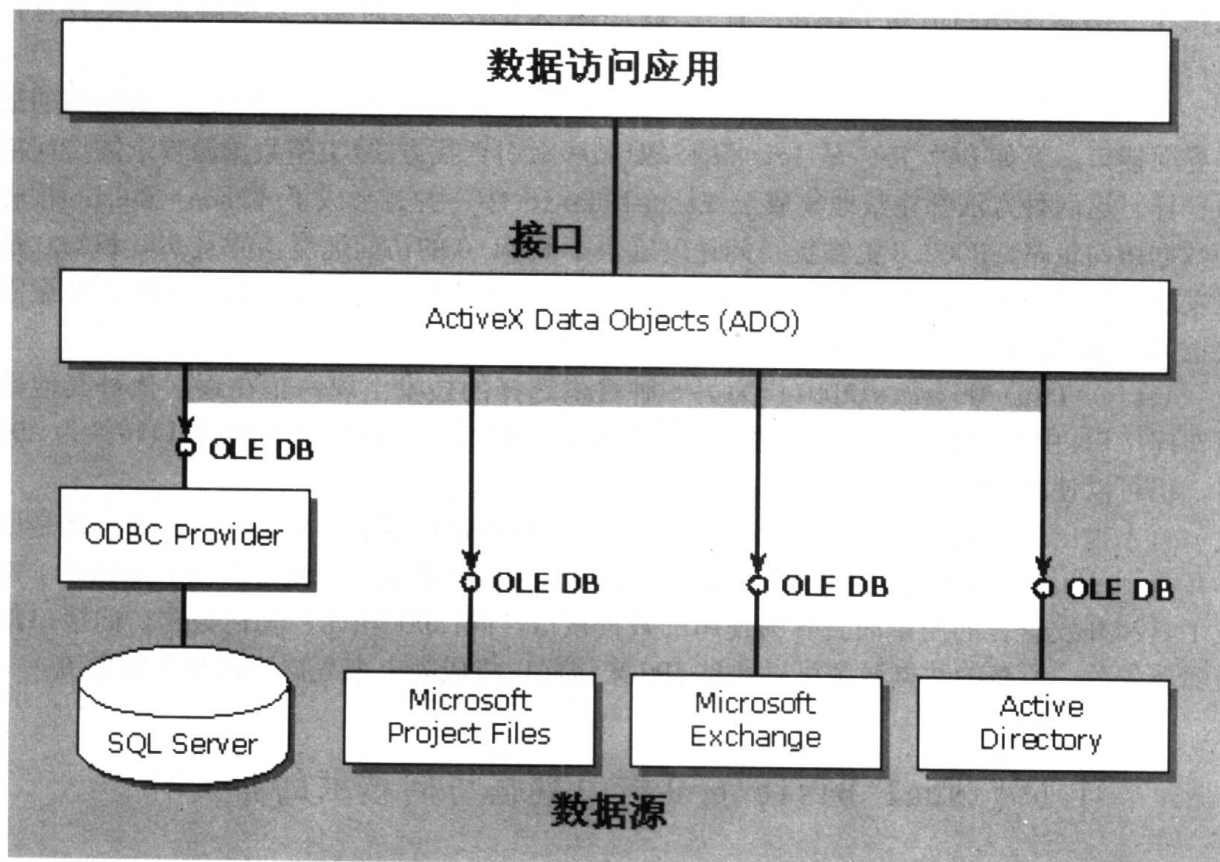


图 1.3 ADO 对象模型

ADO 对象模型定义了一组可编程的自动化对象, 可用于 Visual Basic、Visual C++、Java 以及其他各种支持自动化特性的脚本语言。