

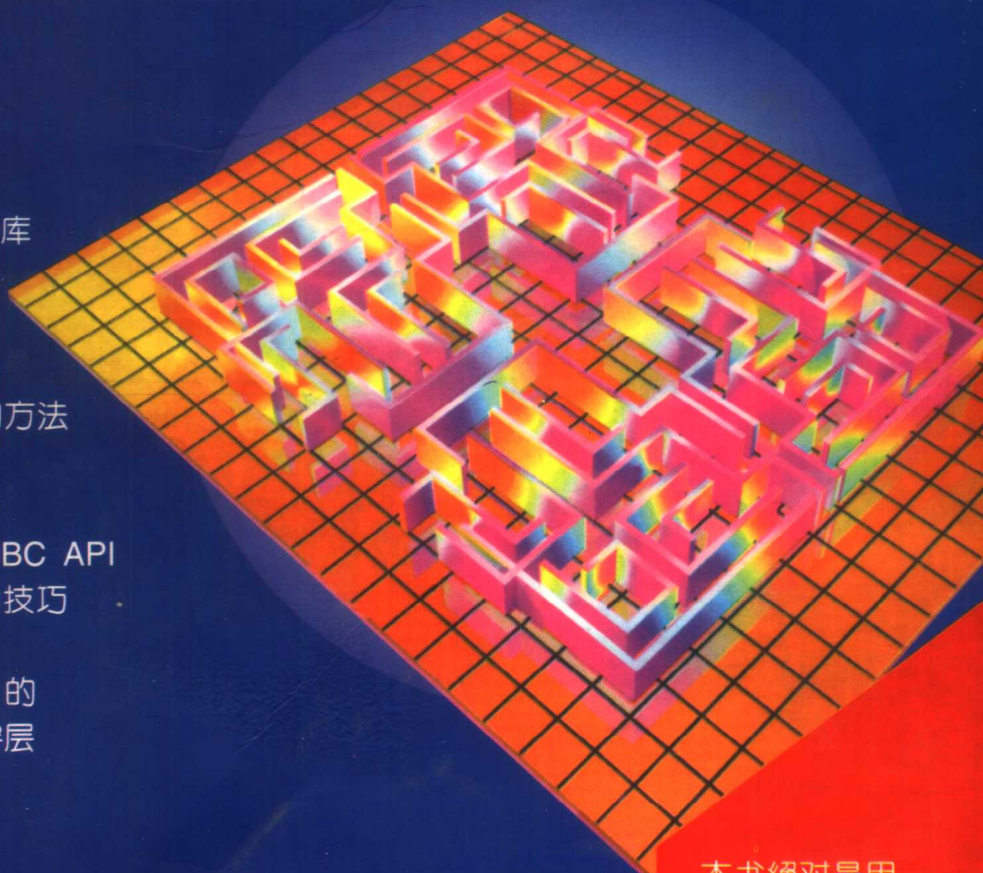
Visual Basic

数据库开发指南

[美] Roger Jennings 著

寇学坤 赵 炜 等译
寿 标 王秩男
韩 柯 审校

- 深入探讨Windows数据库开发技术
- 将其它数据库应用程序转换为Visual Basic 的方法
- 应用OLE 2.0、OLE Automation、ODBC API 和SQL Passthrough的技巧
- 全面讨论Visual Basic 的Access 2.0数据库兼容层



本书绝对是用
Visual Basic开发
数据库应用软件
人员必备的参考书

SAMS
PUBLISHING



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.co.cn>

Visual Basic 数据库开发指南

[美] Roger Jennings 著

寇学坤 赵 炜 等译
寿 标 王秩男

韩 柯 审校

电子工业出版社

内 容 提 要

本书主要讲述利用 Visual Basic 开发数据库应用程序的方法。主要包括: Visual Basic 数据访问对象与控件;数据库和查询设计的概念;数据库前端设计;多用户数据库应用程序;分销数据库应用程序等。本书基本包括了使用 Visual Basic 开发应用程序的各个方面,特别是对核心概念的介绍较为清楚。Visual Basic 是目前深受软件开发人员喜爱的软件,本书会对他们的开发起到良师益友的作用。

本书适合使用 Visual Basic 开发数据库应用软件的工程技术人员及大专院校师生阅读。

Copyright © 1994 by Sams Publishing.

Chinese Copyright © 1996 by Publishing House of Electronics Industry, PRC.

书 名: Visual Basic 数据库开发指南

著 者: [美] Roger Jennings 著

译 者: 寇学坤 赵 炜 寿 标 王秩男

审校者: 韩 柯

责任编辑: 张丽华

排版制作: 电子工业出版社排版室

印 刷 者: 北京大中印刷厂印刷

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.co.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 45.5 字数: 1160 千字

版 次: 1997 年 4 月第 1 版 1997 年 4 月第 1 次印刷

印 数: 0-3000 册

书 号: ISBN 7-5053-3685-1
TP·1543

定 价: 78.00 元

著作权合同登记号 图字: 01-1995-561

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

译者序

Visual Basic 功能强大,直观易学,具有面向对象的特点并具有很好的开放性和可扩充性,它一经问世就迅速成为 Windows 平台开发工具的主流产品之一。正是由于 Visual Basic,特别是 3.0 版本 Visual Basic 的出现,才使得 Windows 真正成为高生产率的开发支撑平台。

目前,在世界范围内桌面数据库应用系统和客户-服务器数据库系统前端应用的需求量很大,有报道说,占软件总开发工作量的 80% 左右都集中在数据库应用方面。可见,好的数据库应用开发平台和工具对于开发高质量的应用产品具有重要意义。Visual Basic 作为通用的开发工具,通过使用 OLE Automation 和第三方厂商提供的数据库感知控件和工具,结合极为灵活的数据处理能力,也很适用于开发桌面数据库应用系统和客户-服务器数据库系统前端应用。对于大批有一定开发经验的 Visual Basic 用户,以及希望将自己 xBase 数据库应用移到 Windows 平台或客户-服务器环境上的程序员,Visual Basic 都是值得认真考虑选择的高质量数据库应用的开发工具。虽然国内已经出版了很多 Visual Basic 的参考书,但是,专门深入细致地介绍 Visual Basic 在桌面数据库应用系统和客户-服务器数据库系统前端应用各个开发阶段中的使用,目前还没有。

本书作者对于桌面数据库应用系统和客户-服务器数据库系统前端应用,以及 Visual Basic 本身都有深入的研究。作者将桌面数据库应用系统和客户-服务器数据库系统前端应用设计原则,与 Visual Basic 提供的解决方法有机地结合起来,配合大量具有典型意义的开发实例,完整地介绍了采用 Visual Basic 开发桌面数据库应用系统和客户-服务器数据库系统前端应用各个方面的问题,对于 Visual Basic 数据库系统开发人员来说,具有特别有效的指导意义。我们深信本书对于国内大量的小型桌面数据库应用系统和客户-服务器数据库系统前端应用开发人员和决策规划人员,都是一本不可多得的参考书。

参加本书翻译工作的人员还有:张立清、王卫平、陈华平、汤书昆、陈韵韵和项莉等。

对本书的翻译有异意者,望提出宝贵的意见。

译者

1996 年 3 月

目 录

引言	(1)
0.1 从数据库开发人员的角度看 Visual Basic	(1)
0.2 本书的使用对象	(2)
0.3 如何有效地利用本书	(3)
0.4 本书的结构	(4)
0.5 本书的排印惯例	(6)
0.6 Visual Basic 和数据库的参考书目	(8)
0.7 跟上 Visual Basic 的发展	(10)
 第一部分 Visual Basic 3.0 数据访问对象与控件	
第一章 将 Visual Basic 推向桌面数据库市场	(15)
1.1 选择 Visual Basic 3.0 作为数据库开发平台	(15)
1.2 使用 Visual Basic 3.0 作为数据库前端	(16)
1.3 数据库前端生成器	(19)
1.4 Visual Basic 和 SQL	(20)
1.5 数据库前端应用程序的分类	(21)
1.6 Visual Basic 3.0 数据库管理系统的分类	(26)
1.7 放弃传统的数据库编程语言	(31)
1.8 Visual Basic 3.0 的数据存取对象	(34)
1.9 OLE 2.0 和 Visual Basic for Application	(38)
1.10 本章小结	(43)
第二章 理解数据存取对象	(44)
2.1 定义数据相关对象的特征	(44)
2.2 定义数据库对象	(49)
2.3 建立并应用 Recordset 对象类的成员	(72)
2.4 索引集和 ListIndexes 方法	(80)
2.5 本章小结	(85)
第三章 运用 Visual Basic 数据存取控件	(86)
3.1 用数据控件创建数据存取对象	(86)
3.2 运用带有数据控件对象的 Visual Basic 代码	(96)
3.3 本章小结	(107)

第二部分 数据库和查询设计的概念

第四章 优化关系数据库的设计	(111)
4.1 数据库系统的分类	(111)
4.2 建立数据模型	(119)
4.3 关系数据库设计规则	(123)
4.4 为提高性能及域完整性进行表索引	(130)
4.5 本章小结	(134)
第五章 学习结构化查询语言	(135)
5.1 回顾 SQL 的形成	(135)
5.2 编写 ANSI SQL 语言	(139)
5.3 比较 Access SQL 方言和 Visual Basic 3.0	(160)
5.4 本章小结	(164)
第六章 理解 Access 数据库引擎	(166)
6.1 Access 数据库引擎的结构	(166)
6.2 结合 Visual Basic3.0 使用 Access 数据库	(171)
6.3 使用 Visual Basic3.0 支持的其他数据库类型	(184)
6.4 根据查询直接创建 Dynaset 和 Snapshot	(194)
6.5 本章小结	(196)
第七章 运用开放数据库连接 API	(197)
7.1 理解 ODBC API 的结构	(197)
7.2 VB.INI 和 APPNAME.INI 中[ODBC]部分的条目	(207)
7.3 通过 Visual Basic 使用桌面 ODBC 数据源	(209)
7.4 使用 Microsoft 查询和桌面数据库驱动程序	(214)
7.5 本章小结	(221)
第八章 运行交叉表和动作查询	(222)
8.1 使用交叉表查询提供汇总数据	(222)
8.2 使用 TRANSFORM 和 PIVOT 建立交叉表查询	(225)
8.3 定义动作查询	(229)
8.4 使用 SQL 动作查询	(232)
8.5 本章小结	(234)

第三部分 数据库前端设计引论

第九章 决策支持应用程序设计	(237)
9.1 在决策支持应用程序背后组织数据	(237)
9.2 设计用户界面	(245)
9.3 创建并使用按钮条和工具条	(251)

9.4 根据累计数据创建图形	(255)
9.5 用网格控件显示细节信息	(261)
9.6 本章小结	(264)
第十章 创建自己的数据存取控件	(266)
10.1 定义约束和非约束数据存取元素	(266)
10.2 创建自填充的约束和非约束组合框及列表框	(274)
10.3 模拟约束及非约束选项组	(286)
10.4 创建等价的连接子表单和子表单上的控件对象	(288)
10.5 模拟约束及非约束 Access OLE 对象帧	(293)
10.6 本章小结	(293)
第十一章 利用交叉表查询生成的摘要数据	(294)
11.1 组织基于图表的决策支持应用程序	(294)
11.2 为复杂的决策支持程序确定导航(Navigation)方法	(301)
11.3 实现前两级图表	(309)
11.4 为 MDI 子表单加细节数据网络	(317)
11.5 生成第三级饼图	(323)
11.6 编写在图形和图表之间导航(Navigate)的代码	(326)
11.7 本章小结	(327)
第十二章 使用 CRYSTAL REPORTS 定制控件打印报表	(329)
12.1 由图形决策支持应用程序打印报表	(329)
12.2 根据记录集对象打印报表	(344)
12.3 使用 Crystal Reports 定制控件	(351)
12.4 本章小结	(365)

第四部分 Visual Basic 数据存取对象的高级编程

第十三章 数据存取对象的进一步使用	(369)
13.1 使用MouseDown 和 Double-Click 事件激活下钻	(369)
13.2 用展开的饼图选择下钻路径	(372)
13.3 用选中的数据填充网络	(382)
13.4 用 Spread/VBX 电子表格代替网络	(390)
13.5 从二级图表直接设置下钻路径	(394)
13.6 本章小结	(398)
第十四章 设计连机事务处理应用程序	(399)
14.1 事务处理应用程序的分类	(399)
14.2 在事务处理中使用 SQL 语句或记录集(recordset)方法	(400)
14.3 处理多表(Multitable)事务	(415)
14.4 本章小结	(420)
第十五章 数据库应用程序和 OLE 与 DDE 的结合	(421)

15.1 理解对象连接与嵌入	(421)
15.2 了解 OLE 2.0	(425)
15.3 安装 OLE 2.0 应用程序和注册数据库	(429)
15.4 使用 Visual Basic 3.0 OLE 容器控件	(435)
15.5 处理存放于 Access OLE Object 字段之中的对象	(442)
15.6 把 Visual Basic 用作 DDE 客户和服务端	(456)
15.7 通过 WINGate 与 DOS 数据库应用程序通信	(461)
15.8 本章小结	(462)
第十六章 在应用程序中使用 OLE Automation 和 Visual Basic	(463)
16.1 利用 OLE Automation 的优点	(464)
16.2 比较 Visual Basic for Application 和 Visual Basic 3.0	(479)
16.3 本章小结	(487)
第十七章 将 Access Basic 代码转换为 Visual Basic 3.0 代码	(488)
17.1 确定转化后 Visual Basic 应用程序的结构	(488)
17.2 把宏动作转换为 Visual Basic 3.0	(490)
17.3 把 Access Basic 代码转入 Visual Basic 模块	(494)
17.4 把 Access1.x 的 DDE 应用程序转化为 Visual Basic 3.0	(499)
17.5 本章小结	(508)

第五部分 多用户数据库应用程序

第十八章 在网络上运行 Visual Basic 数据库应用程序	(511)
18.1 理解网络拓扑和操作	(511)
18.2 在网络环境下保持数据库的安全性	(527)
18.3 探讨 Access 安全系统	(533)
18.4 本章小结	(539)
第十九章 为客户-服务器数据库创建前端	(540)
19.1 使用 Microsoft SQL Server for Windows NT 的 4.2 版本	(540)
19.2 根据客户-服务器数据库创建 ODBC 数据库	(549)
19.3 在 ODBC 数据源中创建表	(554)
19.4 用 GATOR.EXE 显示 ODBC 驱动器的能力	(560)
19.5 用 SQL Passthrough 操作改善应用程序性能	(561)
19.6 本章小结	(563)
第二十章 与 Microsoft Mail、Schedule+ 和 Telephones 交互	(564)
20.1 理解 Microsoft Mail	(564)
20.2 用 Visual Basic 创建工作流应用程序	(570)
20.3 使用 Microsoft Electronic Forms Designer	(574)
20.4 获得并使用 Schedule+ Access 库	(579)
20.5 Windows Telephony API 和 Visual Basic 3.0	(582)

20.6 本章小结	(586)
-----------------	-------

第六部分 分销数据库应用程序产品

第二十一章 为数据库应用程序编制文档	(589)
21.1 准备数据字典	(589)
21.2 使用 VBXRef 为 Visual Basic 代码和表单编制文档	(611)
21.3 为数据库应用程序编制手册	(612)
21.4 本章小结	(616)
第二十二章 为数据库应用程序创建帮助文件	(617)
22.1 理解 Visual Basic 3.0 与 WinHelp 的交互方式	(618)
22.2 使用帮助编写工具	(629)
22.3 用 Visual Basic 代码调用 WinHelp() 函数	(638)
22.4 本章小结	(642)
第二十三章 用 Visual Basic Setup 工具包制作发行软盘	(643)
23.1 用 Setup Wizard 创建分发磁盘	(643)
23.2 分发盘接收者眼中的设置应用程序	(656)
23.3 创建自己的 Setup 应用程序	(658)
23.4 本章小结	(662)
附录 A 有关 Visual Basic 数据库应用程序的资源	(663)
附录 B Visual Basic 对象和变量的命名规则	(682)
附录 C 词汇表	(689)

引 言

对于当前桌面型(desktop)数据库市场的主要软件如 Microsoft Access 1.1、FoxPro 2.5 For Windows、Borland 公司的 Paradox for Windows, Microsoft 将存取数据库引擎加至 Visual Basic (VB) 3.0, 无疑使这种多功能窗口编程语言成为其中强有力的竞争者。尽管 VB 2.0 为 VB 程序员引进了对象变量和 Variant (变体) 数据类型, 但 Database (数据库) 对象却仅限于与那些与开放数据库连接 (ODBC) 的应用程序接口 (API) 的 Microsoft SQL Server 客户-服务器数据库一起使用。若要为 xBase、Paradox 和其它流行的桌面数据库建立前端, 开发人员就需要运用第三方提供的专用控件、动态链接库 (DLL) 或用 VB 写的附加库。而现在 VB 运用 Access 数据库引擎或 ODBC API 向用户提供与 20 多个桌面及客户-服务器 RDBMS 相连的内嵌对象连接。

0.1 从数据库开发人员的角度看 Visual Basic

在 Microsoft Access 1.1、FoxPro 2.5 for Windows、Paradox for Windows 和 Visual Basic (VB) 中选择适合的数据库开发基本平台不是一个简单问题, 对于那些习惯使用基于字符的 FoxPro、dBASE、Clipper 或 Paradox 应用软件的开发者来说更是这样。FoxPro 2.5 for Windows 能够与 xBase 语言兼容, 而且毫无疑问是所有窗口桌面数据库软件中运行速度最快的。Access 1.1 具有独特的用户界面, 能够最方便地运行从简单、中等直至复杂程度的数据库应用程序。Paradox for Windows 桌面 RDBMS 提供了比任何其它 RDBMS 不论是 Windows 上的还是 DOS 上的, 包括客户-服务器数据库系统更丰富的字段数据类型。上述每一种软件均有它自己的特点, 当然也有它的局限, 而只有当用户第一次开发大规模的客户-服务数据库前端应用程序时, 才会发现这些局限性。第一章“将 Visual Basic 推向桌面数据库市场”将 Visual Basic 与 Windows 数据库市场中其它主要竞争者作了比较。

如果你已经有了几年开发 xBase 的经验或 PAL 编程技巧, 并且不愿意为一种新的面向对象的事件驱动的窗口程序语言放弃 xBase RDBMS 或 Paradox, 这并不足为奇。COBOL 程序员曾经也面临相同的问题, 他们的雇主主要求将大型机老的数据库缩减到在由 PC 机组成的客户-服务器系统上运行。幸运的是, 相对于 COBOL 相似于 C 及其他 PC 机上常用的编程语言的程度, Visual Basic 与 xBase 及 PAL 的相似程序更大一些。不过, 对 PAL 程序员来说, Paradox for Windows 的 Object PAL 编程语言同基于字符的 Paradox 3.5 的 PAL 及其前身却没有多少联系。但是, xBase、PAL 的 Visual Basic 都根源于最初的 Dartmouth BASIC, 因此, Visual Basic 应用程序的结构与现在所编写的 xBase 和 PAL 程序很相似。

如果选择 FoxPro for Windows 和 Paradox for Windows 仅仅是因为编程者习惯于用 xBase 和 PAL 代码编程, 这不象是能解决窗口数据库开发平台困境的有效途径 (虽然 J·M·Keynes 认为“从长远看来, 人们都将死去”)。如果要依靠 Windows 应用程序来谋生, 无论公司雇员还是独立开发人员, 都必须提供给公司和用户一些体现当今新技术的应用程序。现在你需要准备 OLE 2.0 (对象的链接和嵌入), 包括它的内部响应 (activation) 和 OLE Automation (OA) 及面向应用程序的 Visual Basic Application (VBA)——过程间通讯的新

“工业标准”。Windows 在其中起了很大作用,通过销售二千五百万套 Windows 3+,Microsoft 公司能够产生足够大的市场影响,使 OLE 2.0、ODBC 和 VBA 成为世界上桌面计算机的标准(不管整个行业是否认同)。一些软件商想在近期内提出另一种替代标准来与 Microsoft 的 OLE 2.0、ODBC 和 VBA 这样庞大的系统竞争几乎是不可能成功的。

对习惯于使用 xBase、PAL、C 和其它语言的程序员,他们需编写成千行代码来建立基于字符的 RDBMS,Windows 桌面数据库的应用是一个挑战。因为用数据控件对象和有也是很少的 Visual Basic 代码可以编出简单但实用的 VB 应用程序。Microsoft Access 也提供类似的能力。但如果使用 FoxPro、xBase 或 Paradox 的 Object PAL,代码将会很长。不过,实际情况是无论用任何一种软件开发商用数据库产品,编程者都必须写很多的代码。关键问题不是代码的多少,而是语言的选择。下面是一些会影响你的职业前途和机会的编程语言因素:

- xBase 和 PAL 是当今最流行的桌面数据库编程语言。但 Microsoft 公司推出了 Object Basic 语言,比如 Access Basic,Visual Basic 3.0 及面向程序的 Visual Basic 的变体,像 Excel Basic,Project Basic,Word Basic。面对如此挑战,xBase 和 Object PAL 是否能最终仍被广泛使用呢?
- 面向应用程序的 Visual Basic 能够同样在 PC 和 Macintosh 机上良好地运行,那么 Microsoft Visual Basic 3.0 会在多长时间内推出 Macintosh 的版本呢?
- Windows NT 可以运行 32 位的 Windows 实用程序,而这些程序也可以在 Windows 3.1、DOS 和没有 DOS 的 Chicago 环境下的 win32s API 下运行(Chicago 是 Microsoft 为即将推出的 Windows 4.0 起的名字),谁将率先应用 32 位的 Windows 桌面数据库管理程序呢?
- 据报导,Microsoft 将为 Windows NT 开发一种具有 RDBMS 特性的对象数据库管理系统(ODBMS),代码名为 Cumulus,那么 Microsoft 公司会选取何种语言开发 Cumulus 应用程序?

本书并不打算直接回答上述这些问题,不过读者将会在看此书后得出自己的结论。

在这本书的写作过程中,Visual Basic 是唯一能支持 OLE 2.0 和 OLE Automation 的数据库开发平台。另外,VBA 也是建立于 Visual Basic 的基础上的。一旦你学习了 Visual Basic,你将会熟练使用 Excel、Project 和 Word Basic。第十六章描述了怎样将 OLE 2.0、OLE Automation 和 VBA 运用到决策支持数据库的前端处理中。

不管最终选择何种语言,都必须能适应来自 Windows 图形用户界面(GUI)的事件驱动方法。另外还有一个事实,那就是 Windows 应用程序的运行速度将不及直接在 DOS 状态下运行 Clipper 和 FoxPro 应用程序。如果有,那也只是少数几个 Windows 程序能在速度竞赛中与 DOS 的相应程序抗衡,不过,当 RISC 工作站上出现 Windows NT 且 Windows 桌面数据库成为 32 位 Windows 产品之后,情况就会改变。

0.2 本书的使用对象

本书主要是为以下人员编撰的,不过也适于其他读者阅读:

- Visual Basic 开发人员。可利用 Visual Basic 3.0 数据库的连接能力来为各种桌面及顾
- 2 •

客-服务器数据库建立产品级的图形前端处理程序。

- Access 1.x 开发人员,与现版本 Microsoft Access 相比,他们需要更多的一些对数据显示及编辑格式的控件。Visual Basic 3.0 的数据库应用程序也比相应的 Access 1.x 程序少用 Windows 的资源。
- 基于字符的 DOS 数据库应用程序开发人员。他们的用户或上司已经决定采纳 Windows 3.1 GUI 或宣布要采用 Windows NT(Windows NT 也许不是一个主要的操作系统,但采用 Windows NT 的公司却是数据库咨询人员的主要顾客)。
- 想花不到用 C 或 C++ 编写数据库应用程序 1/4 的时间和精力的程序员,习惯于用 C++ 的程序员能很快用 Visual Basic 建立起原型数据库应用软件(令人惊讶的是,有很多原型 Visual Basic 应用程序变成生产数据库前端处理程序)。
- 因为公司要将应用从大型机移到微机而面临危险者,尤其是 COBOL 程序员。他们只有通过获取 Windows 数据库开发技巧才能继续就业。
- 客户-服务器数据库开发前端应用的用户。他们厌倦了对服务器上连接的每个用户工作站支付使用许可费。
- Paradox for Windows 开发人员。他们已对 IDAPI 短期可用性和 Paradox for Windows 的无版税运行版失去信心。
- 信息处理负责人和管理信息服务管理员。他们即将对采用何种窗口前端处理生成器作为该组织的标准设备做出决策。
- 其他一些感兴趣人员。他们希望开发人员能更舒适地使用具有商业用途的 Visual Basic 3.0 应用程序。

本书适用于曾经使用过 Visual Basic 3.0, Access Basic 或任何一种 Windows 环境下的传统编程语言,比如 Microsoft 或 Boland C++, Turbo Pascal for Windows 或 Windows 版本的 Smalltalk。本书并不介绍 Visual Basic 编程技巧,已经有很多有关的教学和参考书可以满足读者这部分需要(书后的参考文献中列出了一些适用于初级至中级 Visual Basic 3.0 程序员使用的好书和其它信息来源)。故本书一开始就回顾 Visual Basic 如何在桌面数据库市场中占有一席之地,接下来直接开始讨论 Visual Basic 3.0 中的 object(对象)变量。本书的全部内容都旨在介绍建立有用的 Visual Basic 3.0 数据库应用程序,其中每一个 Visual Basic 3.0 代码例子均连接到一个或多个数据库。

0.3 如何有效地利用本书

你需要专业版本的 Visual Basic 3.0 来重写或修改第八章及其后章节中出现的示例。尽管可以用标准版本的 Visual Basic 3.0 中提供的数据库控件及约束控件对象来编写简单的数据库程序,但是专业版本对复杂的数据库开发是必须的。由于磁盘容量的限制,附盘上没有可执行版本的示范程序。

对于那些要用 Access 后缀为 .MDB 的数据库文件进行商业开发的程序员来说,很有必要买一份 Access 1.1。用 Access 1.1 来建立新数据库或增加表要比用 Visual Basic 3.0 的数据库管理程序容易而且快速。另外,用 Access 1.1 创建 .MDB 数据库文件还有其他的好处,它能够在数据库而不是 VB 代码的层次上实现参照一致性,能够实现一些运用 Visual Basic

无法解决的数据库设计及文档管理。第四章和第二十一章介绍了一些 Access 数据库新的设计和文档管理工具。另外还需要用 Access 1.1 创建一些能设置口令保护的多用户数据库。Visual Basic 内部并不提供实现和修改 Access 数据库的用户注册 ID 和口令保护的功能。第十八章讨论了如何在 Visual Basic 数据库应用层处理安全问题。

如果没有连接 Microsoft 或 Sybase 版本的 SQL Server 或 Oracle 数据库服务器,却又想试用 Microsoft 的 ODBC API,那么可以购买一套 Microsoft ODBC 桌面数据库驱动系统。这套产品包括了 Access 数据库支持的数据库类型的 ODBC 驱动程序。第十九章的主要内容之一就是运用 Access 数据库模仿客户—服务器和大型机数据库。

通过购买一套 Microsoft 电子制表软件(MEF),建立可通邮的 Visual Basic 数据库应用就是一件简单的事情。MEF 属于 Visual Basic 的扩展应用,它提供了一些工具能开发 Windows 下的 Microsoft 邮件 e—form。尽管专业版本 Visual Basic 3.0 的 MAPI 用户控件可以应用于邮件服务,但 EFD 将从标准的信息模板中自动生成许多代码。

用 Visual Basic 3.0 进行商用数据库开发者需要第三方数据感知(data—aware)用户控件的某些功能。以 Visual Basic 3.0 的数据控制为中介,数据感知控件将使你能控制数据库表中的域。本书例程中就运用了一些第三方数据感知控件。这些用户控件的来源则在附录 A 中介绍。

大部分 Visual Basic 3.0 数据库应用程序可以在 4M RAM 的 80386 机器上正常运行,如果要用 Access 1.1 就必须有 8M RAM。

0.4 本书的结构

本书分为六个部分,二十四章。每个部分都针对数据库应用设计的一个主题,这些部分的安排也依照了典型的数据库开发过程。以下介绍本书各部分和各章节的内容。

第一部分 Visual Basic 3.0 数据存取对象及控件

第一部分介绍了 Visual Basic 作为 Windows 环境下进行数据库系统开发的能力。第一章“将 Visual Basic 推向桌面数据库市场”分析了 Visual Basic 提供给数据开发人员的性能和这种语言能够按照 Microsoft 公司的战略占领桌面数据库市场的原因。第二章“理解数据库存取对象”通过示例详细描述了如何建立和掌握 Visual Basic 数据库对象和 Access 的 MDB 数据库集合。第三章“运用 Visual Basic 数据存取控件”介绍了如何运用数据控件对象和与数据库当前记录相关的控件对象编写简单的数据库应用程序。

第二部分 数据库及查询设计概念

第二部分主要讨论关系数据库的设计并介绍运用 SQL 完成选择和查询功能。这些功能运用了 Access 数据库和 ODBC API。第四章“优化关系数据库的设计”指出如何规范化数据从而减少应用数据的冗余。第五章“学习结构化的查询语言”讨论了 ANSI SQL—89 和 SQL—92,还有 Access SQL 与客户—服务器和主机数据库所用的“标准”SQL 的不同之处。第六章“理解 Access 数据库引擎”讨论了利用 Access 数据库引擎和 Visual Basic 代码创建和修改数据库的方式。第七章“运用开放数据库连接 APL”描述了 ODBC 理论及其实际应用,及如何在单用户环境下用 ODBC 管理程序建立 ODBC 数据源。第八章“运行交叉表和动作查询”超

越了简单的 SQL 选择查询模式,它描述了如何编写查询,使其包括 TRANSFORM、PIVOT、INTO 及另外一些较少使用的 SQL 保留字以修改表中的数据。

第三部分 数据库前端设计引论

第三部分主要讨论为数据库建立商业化的决策支持前端处理程序。第九章“决策支持应用程序设计”描述了将原始数据转变为可用 Visual Basic 表格显示出来的易于理解的信息的基本原理。第十章“创建自己的数据存取控件”介绍了如何复制控件对象,比如那些在 Visual Basic 的标准控件对象不能直接完成的记录捡拾(pick)列示组合框和 Access 已链上的子表格。第十一章“利用交叉表查询生成的摘要数据”举例说明了如何运用 Visual Basic 图形用户控件来产生适合在管理中使用粗线条汇总。第十二章“使用 CRYSTAL REPORTS 定制控件打印报表”描述如何设计报表及如何将报表生成与数据库应用程序天衣无缝地结合起来。

第四部分 Visual Basic 数据存取对象高级编程

第四部分将你带入更深一层的商业数据库应用开发中。第十三章“数据存取对象的进一步使用”指出如何运用图形用户控件和 Pinnacle Publishing 的 ChartBuilder 用户控件来创建基于选择项的可在图表上按点热点的透视应用程序。第十四章“设计联机事务处理应用程序”描述了如何设计一些标题下拉(heads-down)的高速数据录入表格,并说明了如何用 Visual Basic 事务处理中的保留字来加快大批量表格内容的更新。第十五章“数据库应用程序和 OLE 与 DDE 的结合”解释了如何用 OLE 的通用源程序进行内部激活以及如何运用 DDE(动态数据交换)实现一些不支持 OLE 的应用程序的通讯。本章中 Excel 5.0 和 Word for Windows 6.0 被用作 OLE 2.0 的源程序。第十六章“在应用程序中使用 OLE Automation 和 Visual Basic”描述了 OLE Automation 的原理,并举例说明了 Visual Basic 3.0 如何控制 Excel 的工作表(Worksheet)对象。第十六章还说明了 VBA 和 Visual Basic 3.0 之间的密切关系。第十七章“将 Access Basic 代码转换为 Visual Basic 3.0 代码”总结了将 Access 1.1 应用程序向 Visual Basic 3.0 转换的方法。

第五部分 多用户数据库应用程序

本书前四部分都是为单用户设计的个人应用。第五部分提供 Visual Basic 数据库应用于客户—服务器结构和网络能力的背景和例证,示例中采用了 Windows for Workgroup 3.1+, Windows NT 高级服务器和 Windows NT 的 SQL Server。第十八章“在网络上运行 Visual Basic 数据库应用程序”介绍了如何用同级的和网络服务器在工作组的各成员间或一个组织中共享数据库。第十九章“为客户-服务器数据库创建前端处理程序”介绍了如何用 Visual Basic 应用程序建立并联结用户—服务器和主机的数据源,其中包括使用 Windows NT 下的 Microsoft SQL Server 的示例。第二十章“与 Microsoft Mail、Schedule+、Telephone 交互”详细介绍了 MAPI 用户控件及 Schedule Access Library(SAL),并提供了用 Microsoft 电子制表软件 EFD 来创建可传递邮件的 Visual Basic 应用程序的例子。

第六部分 数据库应用产品的发行

没有完整的文档及联机用户帮助系统的数据库应用产品是不完整的。第二十一章“为数据库应用程序文档编制”指出如何运用 Visual Basic 的数据库对象汇总创建文本文件格式的数据词典。你也可以将它调入其它软件中,如 Word for Windows 6.0 的字处理及 Excel 5.0。本章还包括了一个为 Access 1.1、Total Access 建立商用文件库的范例。第二十二章“为数据

库应用程序创建帮助文件”介绍了如何运用 Word for Windows 2+和商用 WinHelp 的辅助手段,如 Doc-ToHelp 和 RoboHelp 来加快为 Visual Basic 应用程序增添与上下文有关的帮助系统。最后一章用 Visual Basic Setup 工具包制作发行盘”指出如何能建立起一套专业安装程序,它的性能和界面可与 Microsoft 为 Visual Basic 和其它主流 Windows 应用程序建立的 Setup 文件相媲美。

附录和词汇表

附录提供了本书正文中解释资料的参考数据。附录 A“有关 Visual Basic 数据库应用程序的资源”列出了常规控件,一些具有新功能的附加产品和大致与 Visual Basic 及数据库有关的期刊出版者。销售商和出版商均按产品分类,每个条目均包括他们的地址、电话、传真号和简单的产品介绍。附录 B“Visual Basic 对象和变量的命名规则”介绍了如何根据本书中使用的前缀来识别对象和变量的数据类型。这里的命名惯例依据了 Stan Leszgnski 和 Greg 为 Access 1.x 对象及变量命名而开发的系统。本书第二次印刷增加了两部分“HOT OFF THE PRESS”和运用“Access 2.0 对 Visual Basic 3.0 的兼容能力”,其中指出了如何使 Visual Basic 应用程序适用于 Access2.0 中的.MDB 文件。

本书附录之后是一个专业术语词汇表。对于未接触过 Windows 编程、关系数据库应用、PC 网络和 Windows NT 的人员,表中列出了一些他们也许觉得陌生的词汇,其重点在于定义本书中使用的专业术语、缩略语和缩写词。

0.5 本书的排印惯例

为了区别注释性文字、Windows 对话框中的输入、初始化文件(后缀为.INI)的元素和输入进 Visual Basic 编辑窗口的代码,本书采用了多种编排形式。下面将具体介绍本书的编排原则。

字集、字型 and 字体

在 Windows 3+的文档和帮助文件中,字体(font)指排印工和图形美工(graphic artists)所称的字型(type face)。而在图形美工术语中,字集指字型的集合。字集包括不同形式的字型(一般罗马字体、斜草体)、字重(重、黑体、半黑体、book 和轻)、字宽(扁型、长型)和属性(下划线、双下划线、贯穿线)。一个字体仅指一种特定大小、形式、字重、属性的字型,比如长型加粗斜体 12 点 Futura。本书在和字型有关的注释材料中,无论是计算机屏幕输出单元 VDU 还是打印输出均采用图形美工的术语。

键位组合、菜单选择和对话框输入

加速(ALT+KEY)和切换(CTRL+KEY)组合常用来代替鼠标控制,本书中这些组合由“+”号来表示,比如,CTRL+C 就表示将一块选择拷贝到 Windows Clipboard,ALT+H 表示在对话框中选择了帮助(HELP)选项。一些应用软件,如 Word for Windows,用多键位组合,如用 CTRL+SHIFT+KEY 来激活宏(macros)。

用加速组合选择菜单就是用 ALT 加菜单中黑色的字母,比如File。一般菜单选项的排列均按该操作在完成全过程中的位置决定,如File Open。本书中没有出现菜单选择后引导出现对话框的椭圆。

用户需要在对话框或 Windows 文档插入处的输入都用黑体表示。菜单选择中要求输入的地方,比如在对话框中输入文件名,文件名用大写形式标出,以符合 Windows 3+ 设定标准,比如,File Open VB3DEMO.MAK。输入完成后,可用 TAB 键将光标移至其它对话框或用回车键(ENTER)表示确认。

需要连续击键来完成某一操作的可能性不大,如果有,那么这些需要击的键将使用逗号连接且中间没有空格,比如 ALT,X,Y。

Visual Basic 代码、SQL 语句和其它语言源代码。

Visual Basic 代码、SQL 语句和其它语言源代码的例子都采用等宽字体。Visual Basic 中的保留字为加粗等宽字体。ANSI SQL 的保留词、关键词和 xBase 编程语言均用大写表示。下面是一个编排 SQL 命令示范:

```
SELECT name,Address,City,Zip-code FROM Customers WHERE Zip-Code >= 90000
```

用来表示同一意思的 xBase 语言是这样的形式:

```
USE customers
```

```
LIST name,address,city,zip-code WHERE zip-code >= 90000
```

在 SQL 命令示例中,一些 Access SQL 中可用的 Visual Basic 功能没有用加粗等宽字体。

有一些 SQL 中的特殊实现不符合 ANSI SQL-92 标准,比如 Microsoft 查询中有用来表明时标型变量的 {ts Date Variable},出现在 SQL 的应用对话框中。而一些在 SQL-92 标准中未出现的命令,如 Access SQL 中的 PIVOT 和 TRANSFORM 命令,仍用大写形式。

初始化文件和注册数据库文件中的项

Windows Visual Basic、Microsoft Access 1.x 初始化文件的项均另起一行。后缀.INI 的文件部分用方括号加选项以示区别,比如:

```
[Options]
```

```
SystemDB=c:\vbapps\system.mda
```

用注册数据库编辑器 RegEdit 生成的注册数据库,其项目也用等宽字体以示区别,这有利于显示注册数据库项目的树状层次(如同文件目录)。

Visual Basic 代码示例和代码片段

正如以前提到的,所有 Visual Basic 和其它 Object Basic 语言的示范程序均用等宽字体。本书中,Object Basic 包括 Visual Basic 和 Visual Basic 应用,仅除去 Word for Windows 6.0 中使用的 Word Basic。夹在正文中的代码片段也单独成行。其字型和字重的原则如下:

- Visual Basic 中的保留词、算子、符号及所有对象方法(包含对象功能)都设为黑体。Visual Basic 的其它文法成分,如非保留词的关键词、属性、集合名和成员集合名均用一般字体,举个例子:Dim cboFindname As Control。
- 除了 Visual Basic 内部常数 True、False、Null 以外,其它符号常数名均用大写。

- 可替换的变量名、命令和参数均用斜体。但是一些能显示变量所属 Visual Basic 基本数据类型和对象类型的前缀则不用斜体,如 Dim xlObjectVar As Object。
- 可替换的数据类型用黑色斜体,如,Dim anyVariable As Datatype。
- 黑色方括号([])所包含的内容是对象、属性和方法,否则这些内容仅在被双引号引起时才符合 Visual Basic 的语法规则。这方面的例子象 Access 数据库中带有空格或其它可打印字符(不包括下划线)的字段名及成员集合的与 Visual Basic 保留词冲突的对象名。一般的方括号([])表示一些任选项,比如一个函数中的自变量。
- 大括号({})表示必须在被括起来的两个由管道符(|)连接的元素中选择一个,比如, Do {While|Until}...不过也有例外,当大括号用于 SQL 命令表示的 Microsoft 查询。
- 省略号(...)表示省略了一条命令中余下的应该出现的代码结构。
- 连续符(→)用在连接由于版面限制而分作两行的一句代码。当你在代码编辑窗口输入该命令时,必须写在一行以保持命令文法的完整性。

数据和对象类型的前缀标识

本书的代码示例中运用了一些两个或三个字母的前缀,其目的是为了区别 Visual Basic 和其它对象 Basic 中的变量、符号常数和对象变量所属的基本数据类型。本书应用的前缀依照了对 Access Basic 的对象和变量命名过程中所提出的标准,该标准最早由 Stan Leszynski 和 Greg Reddick 在 Pinnacle 出版公司出版的 Smart Access 通讯中提出。

本书的所有示范均运用了显式说明,也就是说,在给变量赋值的过程前均对变量进行说明,全程的、常数或用 Dim 命令。本书运用的是显式数据类型说明,属于 Variant 数据类型的变量即用 As Variant 定义,除非缺省数据类型即为 Variant。本书中不包含数据类型的说明符,如 %、@ 和 !,仅有一个例外,在返回值为字符串的函数名后加 \$。惊叹号(!)用来在对象包含在总对象(container)的命令中作分隔符。一些与 C 语言中 Hungarian 符号相关的前缀,在本书的普通或对象变量中仍保持一致。

strStringVar、intIntegerVar、lngLongIntVar、curCurrencyVar 和 varVariantVar 是加了前缀的基本数据类型的变量名的实例。Variant 类型是否属于基本数据类型仍是有争议的。Microsoft 公司及本书应用基本数据类型一词,其目的是为了区分普通变量(包括变量名为 Visual Basic 保留词的 Variant 类型)和对象变量(它们的名称可以是保留词或关键词)。

前缀还用于识别用户定义的对象变量的数据类型,常见的 Visual Basic 对象变量类型有 dbDatabase、tblTable、qdfQueryDef、dsDynaset 和 ssSnapShot。对 Object 类型的变量,其前缀是其应用程序的两个字母,如 xlObject (Excel 5.0)、mpObject (Microsoft Project)、wwObject (Word for Windows)、cdObject (CoreIDRAW! 4.0)。

前缀还用于识别用户用 Type...End Type 命令定义的变量。附录 B“Visual Basic 对象和变量的命名规则”详细介绍了类型识别前缀的来源和应用。

0.6 Visual Basic 和数据库的参考书目

在引言的前部曾提到,本书主要是为那些已经熟悉 Visual Basic 和 Access Basic 的人员而写。如果你首次使用 Visual Basic 高性能的数据库前端处理程序,你就想要得到一些初级