

计算机软件工具应用与开发系列

Visual Basic 6.0 中文版 应用与开发

鸿志创作组 编著



科学出版社

计算机软件工具应用与开发系列

Visual Basic 6.0

中文版应用与开发

鸿志创作组 编著

科学出版社

1999

内 容 简 介

本书首先从 Visual Basic 6.0 的新特征开始,随后对 Visual Basic 的基础知识、编程方法和相关技术进行了详细的介绍。

本书介绍了如何使用 Visual Basic 开发应用程序,内容包括 Visual Basic 集成开发环境、Visual Basic 编程基础、Visual Basic 工程管理、Visual Basic 用户界面、Visual Basic 标准控件和 Visual Basic 应用程序的查错和调试。另外,本书还介绍了一些高级的 Visual Basic 编程技术,内容包括 Visual Basic 高级编程基础、Visual Basic 的对象编程、创建 ActiveX 控件和 Visual Basic 的数据库编程。

本书适用于学习和使用 Visual Basic 进行程序设计和工程开发的广大计算机用户,以及大专院校相关专业的师生。

图书在版编目 (CIP) 数据

Visual Basic 6.0 中文版应用与开发/鸿志创作组编著. -北京: 科学出版社, 1999.1

(计算机软件工具应用与开发系列)

ISBN 7-03-007194-8

I. V… II. 鸿… III. Basic 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (98) 第 38374 号

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

北京双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1999 年 1 月第 一 版 开本: 787×1092 1/16

1999 年 1 月第一次印刷 印张: 26 1/2

印数: 1—5 000 字数: 598 000

定价: 35.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

前 言

欢迎使用 Microsoft Visual Basic, 它提供了开发 Microsoft Windows 应用程序的最迅速、最简捷的方法。不论是 Microsoft Windows 应用程序的资深专业开发人员还是初学者, Visual Basic 都为他们提供了整套工具, 以方便开发应用程序。

何谓 Visual Basic? “Visual” 指的是开发图形用户界面 (GUI) 的方法。不需编写大量代码去描述界面元素的外观和位置, 而只要把预先建立的对象拖放到屏幕上的一点即可。如果已使用过诸如 Paint 之类的绘图程序, 则实际上已掌握了创建用户界面的必要技巧。

“Basic” 指的是 BASIC (Beginners All - Purpose Symbolit Instruction Code) 语言, 一种在计算技术发展历史上应用得最为广泛的语言。Visual Basic 在原有 BASIC 语言的基础上进一步发展, 至今包含了数百条语句、函数及关键词, 其中很多和 Windows GUI 有直接关系。专业人员可以用 Visual Basic 实现其他任何 Windows 编程语言的功能, 而初学者只要掌握几个关键词就可以建立实用的应用程序。

Visual Basic 不仅是 Visual Basic 编程语言。Visual Basic 编程系统和 VBA 都使用这一语言。Visual Basic 编程系统和 VBA 是 Visual Basic 语言的子集。这样, 在学习 Visual Basic 中得到的经验可应用到所有这些领域中。

无论按照哪种标准, Microsoft Visual Basic 都是计算机历史上最为成功的 (同时也是最流行的) 编程语言。该软件的最新版本 - Visual Basic 6.0, 虽然不如 Visual Basic 5.0 相对于以前版本那样震撼人心, 但如果您计划将其用于数据库应用程序的开发, 或是用于构建与 Web 有关的应用软件, 那么它是必不可少的工具。如果您涉及的是更为平凡的事务, 则 VB6 也是您完成的上佳选择。

本书的读者

本书是为所有对 Visual Basic 有兴趣的读者编写的。本书中不但有足够的基本信息帮助初学者起步, 还包括了许多高级功能的介绍使本书对于 Visual Basic 的使用者和开发者也有实际的价值。

本书概况

本书共包括 16 章, 外加一个附录。本书的第一章介绍了 Visual Basic 的新特征。然后从第 2 章到第 11 章介绍了如何使用 Visual Basic 开发应用程序, 其中包括 Visual Basic 6.0 的入门知识和 Visual Basic 语言介绍, Visual Basic 的窗体和控件, 管理 Visual Basic 工程, 创建用户界面, 使用 Visual Basic 6.0 的标准控件, Visual Basic 的文本和图形功能, 响应用户输入事件以及调试代码和处理错误等内容。然后, 从第 12 章到第 16 章介绍了一些 Visual Basic 编程的高级功能, 包括 Visual Basic 的对象编程、使用和创建 ActiveX 控件、访问数据以及 Visual Basic 的数据库编程等内容。

在本书的附录中, 我们列出了 Visual Basic 的常用词汇表。

致 谢

很高兴本书能够成书并出版，编者陈苏桥、周祖胜、任伟、郝水侠、张运涛、胡宏、刘吉强、王文明、王志峰、李梁、邓斌、杨新苗、孟平、廖文斌、黄章任、廖文彬、邓深等感谢众人的帮助和支持，特别要感谢王伟、邢宏宇、王宇、王映洲等几位同仁的特别支持。

Visual Basic 的内容博大精深，由于时间仓促，我们的水平以及篇幅有限，书中一定有许多不足之处，敬请广大读者不吝指正。

编 者

1998 年 11 月

目 录

前 言

1 Visual Basic 6.0 的新特征	(1)
1.1 数据访问新特征	(1)
1.2 Internet 新特征	(2)
1.2.1 IIS 应用程序	(2)
1.2.2 DHTML 应用程序	(4)
1.2.3 Web 发布向导	(6)
1.3 控件新特征	(6)
1.3.1 新控件	(6)
1.3.2 增强的控件	(7)
1.3.3 控件新特征	(8)
1.4 部件创建新特征	(8)
1.5 语言新特征	(9)
1.5.1 公有方法中的用户定义数据类型	(9)
1.5.2 函数返回值可以为数组	(9)
1.5.3 分配大小可调数组	(10)
1.5.4 文件系统对象(FSO)	(10)
1.5.5 CallByName 函数	(10)
1.5.6 增强的 CreateObject 函数	(10)
1.5.7 增强的 StrConv 函数	(11)
1.5.8 词典对象	(11)
1.5.9 新字符串函数	(11)
1.6 向导新特征	(11)
1.6.1 增强的安装向导(打包和展开向导)	(11)
1.6.2 数据对象向导	(13)
1.6.3 增强的数据窗体向导	(13)
1.6.4 增强的应用程序向导	(14)
1.6.5 外接程序设计器(Add-In Designer)	(14)
1.6.6 增强的类生成器实用工具	(16)
1.6.7 工具栏向导	(16)
1.7 升级使用了 Windows 通用控件的应用程序	(17)
1.7.1 为什么要升级	(17)
1.7.2 如何升级	(18)
1.8 Visual Basic 6.0 的版本	(18)
1.8.1 Visual Basic 学习版	(18)
1.8.2 Visual Basic 专业版	(18)
1.8.3 Visual Basic 企业版	(18)

2 Visual Basic 6.0 入门	(19)
2.1 Visual Basic 概念	(19)
2.1.1 Windows 的工作方式:窗口、事件和消息	(19)
2.1.2 事件驱动模型	(20)
2.1.3 交互式开发	(20)
2.2 Visual Basic 6.0 集成开发环境	(21)
2.2.1 启动 Visual Basic 6.0 集成开发环境	(21)
2.2.2 菜单栏	(22)
2.2.3 上下文菜单	(23)
2.2.4 工具栏	(23)
2.2.5 工具箱	(24)
2.2.6 工程资源管理器	(25)
2.2.7 属性窗口	(25)
2.2.8 窗体设计器	(26)
2.2.9 窗体布局	(26)
2.2.10 对象浏览器	(26)
2.2.11 代码编辑器窗口	(26)
2.2.12 立即、本地和监视窗口	(28)
2.3 定制 IDE 环境	(28)
2.3.1 “编辑器”选项卡	(28)
2.3.2 “编辑器格式”选项卡	(29)
2.3.3 “通用”选项卡	(30)
2.3.4 “可连接的”选项卡	(31)
2.3.5 “环境”选项卡	(31)
2.3.6 “高级”选项卡	(32)
2.4 Visual Basic 开发应用程序的基本特点	(33)
2.4.1 可视开发和事件驱动编程	(33)
2.4.2 几个常用属性	(33)
2.4.3 几个常用方法	(34)
2.4.4 几个常见事件	(35)
2.5 创建第一个 Visual Basic 程序	(39)
2.5.1 创建工程	(39)
2.5.2 调整大小、移动和锁定控件	(40)
2.5.3 设置属性	(41)
2.5.4 编写代码	(42)
2.5.5 运行应用程序	(43)
2.5.6 保存工程	(43)
3 使用 Visual Basic 创建应用程序	(45)
3.1 Visual Basic 应用程序的结构	(45)
3.2 代码编写机制	(46)
3.2.1 代码模块	(47)
3.2.2 代码编辑器	(47)
3.2.3 自动完成编码	(48)

3.3 编码基础	(49)
3.3.1 将单行语句分成多行	(49)
3.3.2 将多个语句合并到同一行上	(49)
3.3.3 在代码中添加注释	(49)
3.3.4 理解数制系统	(50)
3.3.5 Visual Basic 的命名约定	(50)
3.4 编写代码之前的考虑	(51)
3.5 使用 Visual Basic 创建 VCR 应用程序	(52)
3.5.1 创建 VCR 应用程序的主窗体	(52)
3.5.2 创建 VCR 应用程序的定时录像窗体	(60)
3.5.3 创建 VCR 应用程序的标准模块	(63)
3.5.4 创建 VCR 应用程序的类模块	(66)
3.5.5 演示 VCR 应用程序	(67)
4 Visual Basic 语言	(69)
4.1 变量	(69)
4.1.1 声明变量	(69)
4.1.2 变量的作用范围	(71)
4.1.3 使用同名的变量	(72)
4.1.4 静态变量	(74)
4.2 常数	(75)
4.3 数据类型	(76)
4.3.1 Numeric 数据类型	(77)
4.3.2 Byte 数据类型	(77)
4.3.3 String 数据类型	(78)
4.3.4 Boolean 数据类型	(78)
4.3.5 Date 数据类型	(79)
4.3.6 Object 数据类型	(79)
4.3.7 Variant 数据类型	(79)
4.3.8 转换数据类型	(80)
4.4 数组	(81)
4.4.1 声明固定大小的数组	(81)
4.4.2 声明包含其他数组的数组	(82)
4.4.3 多维数组	(82)
4.4.4 动态数组	(83)
4.5 过程	(84)
4.5.1 子(Sub)过程	(84)
4.5.2 函数(Function)过程	(86)
4.5.3 调用过程	(86)
4.5.4 向过程传递参数	(88)
4.6 Visual Basic 的控制结构	(90)
4.6.1 判断结构	(90)
4.6.2 循环结构	(92)
4.6.3 嵌套控制结构	(95)

4.6.4	退出控制结构	(95)
4.6.5	退出子过程或函数过程	(96)
5	Visual Basic 的窗体和控件	(98)
5.1	属性、方法和事件概述	(98)
5.2	设计窗体	(100)
5.2.1	设置窗体属性	(100)
5.2.2	窗体事件和方法	(100)
5.3	创建 Controls 应用程序主窗体	(101)
5.4	单击按钮执行操作	(104)
5.4.1	使用命令按钮	(104)
5.4.2	创建测试命令按钮的窗体	(104)
5.5	文本控件	(106)
5.5.1	用 Label 显示文本	(106)
5.5.2	使用 TextBox	(109)
5.6	用于选择的控件	(112)
5.6.1	用 CheckBox 选择单个选项	(112)
5.6.2	用 OptionButton 进行选项分组	(114)
5.6.3	使用 ListBox 和 ComboBox	(117)
5.6.4	用 ScrollBar 作为输入设备	(118)
5.7	显示图片和图形的控件	(118)
5.7.1	使用 PictureBox 控件	(118)
5.7.2	轻图形控件	(119)
5.7.3	Images 应用程序	(120)
5.8	附加控件	(122)
5.8.1	数据访问控件	(123)
5.8.2	文件系统控件	(123)
5.8.3	其他控件	(123)
5.9	焦点概述	(123)
5.10	设置 Tab 键顺序	(124)
5.11	菜单基础	(125)
5.11.1	弹出式菜单	(125)
5.11.2	菜单编辑器	(126)
5.12	用对话框提示用户	(127)
6	管理 Visual Basic 工程	(128)
6.1	使用工程	(128)
6.2	Visual Basic 工程的结构	(129)
6.3	创建、打开和保存工程	(130)
6.4	添加、删除和保存文件	(131)
6.5	在工程中添加控件	(132)
6.6	制作和运行可执行文件	(134)
6.7	设置工程选项	(135)
6.8	使用外接程序和向导	(136)
6.8.1	使用外接程序管理器	(136)

6.8.2 使用向导	(137)
6.9 使用应用程序向导	(137)
6.9.1 应用程序向导介绍	(138)
6.9.2 应用程序向导——界面类型	(139)
6.9.3 应用程序向导——菜单	(139)
6.9.4 自定义工具栏	(140)
6.9.5 应用程序向导——资源	(141)
6.9.6 应用程序向导——Internet 连接性	(142)
6.9.7 应用程序向导——标准窗体	(142)
6.9.8 应用程序向导——数据访问窗体	(143)
6.9.9 应用程序向导——完成	(144)
6.10 使用数据窗体向导	(145)
7 创建用户界面	(147)
7.1 界面设计基础	(147)
7.1.1 图形的设计原则	(148)
7.1.2 Windows 界面准则	(150)
7.1.3 用户辅助模式	(150)
7.2 界面样式	(152)
7.3 多文档界面(MDI)应用程序	(154)
7.3.1 设计时使用 MDI 子窗体	(155)
7.3.2 MDI 窗体运行时的特征	(155)
7.4 创建 MDI NotePad 应用程序的界面	(156)
7.5 使用 MDI 窗体及其子窗体	(157)
7.5.1 指定活动子窗体或控件	(158)
7.5.2 加载 MDI 窗体及其子窗体	(158)
7.5.3 设置子窗体	(159)
7.5.4 卸载 MDI 窗体	(159)
7.6 使用菜单	(160)
7.6.1 用菜单编辑器创建菜单	(160)
7.6.2 菜单标题及其命名规则	(163)
7.6.3 创建子菜单	(163)
7.6.4 创建菜单控件数组	(164)
7.6.5 运行时创建和修改菜单	(165)
7.6.6 编写菜单控件代码	(166)
7.6.7 显示弹出式菜单	(167)
7.7 工具栏	(168)
7.7.1 创建工具栏	(168)
7.7.2 协调菜单和工具栏	(170)
7.8 对话框	(170)
7.8.1 模式与无模式对话框	(171)
7.8.2 预定义对话框	(171)
7.9 自定义窗体对话框	(172)
7.9.1 添加标题	(173)

7.9.2	设置标准对话框的属性	(173)
7.9.3	添加命令按钮	(174)
7.9.4	使对话框中的控件无效	(175)
7.9.5	显示自定义对话框	(175)
7.9.6	创建 MDI NotePad 应用程序的“查找”窗体对话框	(176)
8	使用 Visual Basic 的标准控件	(178)
8.1	Visual Basic 控件简介	(178)
8.1.1	内部控件	(179)
8.1.2	标准 ActiveX 控件	(180)
8.1.3	控件命名约定	(180)
8.1.4	使用控件值	(181)
8.2	使用控件数组	(182)
8.2.1	设计时创建控件数组	(182)
8.2.2	Calculator 示例应用程序	(183)
8.2.3	运行时添加控件数组	(187)
8.3	文件系统控件	(189)
8.3.1	检查文件系统	(189)
8.3.2	驱动器列表框	(190)
8.3.3	目录列表框	(190)
8.3.4	文件列表框	(191)
8.3.5	组合文件系统控件	(192)
8.3.6	文件搜索器应用程序	(192)
8.4	Timer 控件	(198)
8.4.1	示例应用程序: Alarm.vbp	(199)
8.4.2	响应 Timer 事件	(199)
8.5	Frame 控件	(201)
8.6	OLE 容器控件	(201)
8.7	CommonDialog 控件	(202)
8.7.1	显示“打开”和“另存为”对话框	(203)
8.7.2	使用“颜色”对话框	(204)
8.7.3	使用“字体”对话框	(204)
8.7.4	使用“打印”对话框	(205)
8.7.5	使用 ShowHelp 方法显示帮助文件	(206)
8.8	滚动条控件	(207)
8.8.1	滚动条控件是如何工作的	(207)
8.8.2	创建可滚动的图形窗口	(208)
9	Visual Basic 的文本和图形功能	(211)
9.1	使用字体	(211)
9.2	显示文本	(213)
9.2.1	在单行上显示不同的项目	(213)
9.2.2	在指定位置显示 Print 输出	(214)
9.2.3	TextHeight 和 TextWidth 方法	(214)
9.3	格式化数字、日期和时间	(215)

9.4 使用剪切板对象	(217)
9.4.1 使用剪切板处理文本	(217)
9.4.2 在剪切板中使用多种格式	(218)
9.4.3 检查剪切板中的数据格式	(219)
9.5 使用坐标系统	(220)
9.5.1 改变对象的坐标系统	(221)
9.5.2 转换刻度	(223)
9.6 使用图形控件	(223)
9.6.1 添加图片	(224)
9.6.2 窗体和控件的图形属性	(225)
9.6.3 持久的图形	(226)
9.6.4 裁剪区域	(227)
9.6.5 动态移动控件	(227)
9.6.6 动态改变控件大小	(228)
9.6.7 创建简单动画	(229)
9.7 使用图形方法	(229)
9.7.1 图形方法画图的原理	(230)
9.7.2 画点	(230)
9.7.3 画直线	(231)
9.7.4 绘制方框	(232)
9.7.5 画圆	(233)
9.7.6 画椭圆	(234)
9.7.7 在任意位置绘制图形	(235)
9.7.8 指定线宽	(235)
9.7.9 指定实线或虚线	(235)
9.7.10 窗体加载时创建图形	(236)
9.8 使用颜色	(237)
9.8.1 使用 RGB 函数	(237)
9.8.2 使用颜色属性	(237)
9.9 使用图片对象	(238)
10 响应用户输入事件	(240)
10.1 响应鼠标事件	(240)
10.1.1MouseDown 事件	(241)
10.1.2 MouseMove 事件	(242)
10.1.3 MouseUp 事件	(243)
10.2 检测鼠标按钮(button 参数)	(244)
10.2.1MouseDown 和 MouseUp 事件中的 button 参数	(245)
10.2.2 MouseMove 事件中的 button 参数	(246)
10.3 检测 Shift, Ctrl 和 Alt 键的状态	(246)
10.4 响应键盘事件	(248)
10.4.1 低级键盘处理程序	(248)
10.4.2 KeyPress 事件	(249)
10.4.3 KeyUp 和 KeyDown 事件	(249)

10.4.4	窗体级键盘处理程序	(251)
10.5	拖放	(252)
10.5.1	启动自动播放模式	(253)
10.5.2	改变拖动图标	(253)
10.5.3	放下对象时的响应	(253)
10.5.4	启动拖动控制	(254)
10.5.5	改变控件的位置	(255)
10.6	OLE 拖放	(256)
10.6.1	自动拖放和手工拖放	(257)
10.6.2	启动自动 OLE 拖放	(257)
10.6.3	DataObject 对象	(258)
10.6.4	OLE 拖放工作原理	(259)
10.6.5	启动手工 OLE 拖动	(260)
10.6.6	将 OLE 拖放源放到 OLE 拖放目标上	(261)
10.6.7	从 Windows 资源管理器中拖动文件	(262)
11	调试代码和处理错误	(264)
11.1	如何处理错误	(264)
11.2	设计错误处理程序	(266)
11.2.1	设置错误捕获	(267)
11.2.2	编写错误处理例程	(267)
11.2.3	退出错误处理例程	(267)
11.3	调试方法	(269)
11.3.1	编译错误	(270)
11.3.2	运行时错误	(270)
11.3.3	逻辑错误	(270)
11.3.4	调试工具怎样提供帮助	(270)
11.3.5	避免错误	(271)
11.4	设计时、运行时和中断方式	(271)
11.4.1	标识当前模式	(271)
11.4.2	用工具栏改变模式	(272)
11.5	使用调试窗口	(272)
11.6	使用中断模式	(273)
11.6.1	在有问题的语句处进入中断模式	(274)
11.6.2	更正运行时错误并继续	(274)
11.6.3	使用监视表达式监视数据	(275)
11.6.4	使用断点中断程序的执行	(277)
11.6.5	使用 stop 语句	(278)
11.7	运行应用程序的特定部分	(279)
12	Visual Basic 的对象编程	(281)
12.1	Visual Basic 对象	(281)
12.1.1	什么是对象	(281)
12.1.2	如何使用对象	(282)
12.1.3	对象之间的关系	(283)

12.2 创建对象	(285)
12.2.1 使用对象变量	(285)
12.2.2 特定对象类型和一般对象类型	(285)
12.2.3 使用 New 关键字	(286)
12.2.4 释放对对象的引用	(287)
12.2.5 传递对象给过程	(287)
12.3 查找对象信息	(288)
12.4 创建自己的类	(290)
12.4.1 类的作用	(290)
12.4.2 定制窗体类	(291)
12.4.3 定义类模块	(292)
12.5 在类中添加属性和方法	(294)
12.5.1 向类中添加属性	(294)
12.5.2 使用属性过程	(296)
12.5.3 向类中添加方法	(297)
12.5.4 使属性和方法成为缺省的	(298)
12.5.5 友元属性和方法	(298)
12.6 在类中添加事件	(299)
12.6.1 事件的声明与引发	(299)
12.6.2 处理事件的对象	(301)
12.6.3 向窗体添加事件	(303)
12.7 创建自己的集合类	(304)
12.8 多态	(307)
12.8.1 Visual Basic 是如何提供多态的	(308)
12.8.2 创建和实现接口	(308)
12.8.3 实现属性	(311)
13 Visual Basic 高级编程基础	(312)
13.1 使用多工程	(312)
13.2 应用程序设置值	(314)
13.2.1 创建和保存应用程序设置值	(314)
13.2.2 检索应用程序设置值	(315)
13.2.3 删除应用程序设置值	(315)
13.3 使用条件编译	(316)
13.4 使用资源文件	(317)
13.5 使用模板	(317)
13.6 使用命令行开关	(318)
13.7 将工程编译成本地代码	(319)
13.8 创建自定义(用户定义)数据类型	(321)
13.8.1 声明自定义类型的变量	(321)
13.8.2 包含数组的用户定义类型	(321)
13.8.3 向过程传递用户定义类型	(322)
13.8.4 包含对象的用户定义类型	(322)
13.8.5 嵌套数据结构	(323)

13.9	使用枚举数据类型	(323)
13.10	用集合替代数组	(325)
14	使用 and 创建 ActiveX 控件	(326)
14.1	ActiveX 控件概述	(326)
14.1.1	加载 ActiveX 控件	(327)
14.1.2	ActiveX 控件的图标及功能	(327)
14.1.3	ActiveX 控件类	(329)
14.2	使用 SysInfo 控件	(329)
14.2.1	检测当前操作系统及其版本	(330)
14.2.2	检测屏幕大小和分辨率的变化	(331)
14.2.3	管理硬件配置和即插即用设备	(332)
14.2.4	监视电池电源的状态	(332)
14.3	控件创建基础	(334)
14.3.1	控件创建术语	(335)
14.3.2	UserControl 对象	(336)
14.3.3	建造 ActiveX 控件的方法	(337)
14.3.4	创建控件的步骤	(337)
14.4	创建一个 ActiveX 控件	(338)
14.4.1	创建 ControlDemo 工程	(338)
14.4.2	添加 TestCtlDemo 工程	(339)
14.4.3	在设计时运行 ShapeLabel 控件	(340)
14.4.4	UserControl 对象的存活期	(342)
14.4.5	绘制 ShapeLabel 控件	(344)
14.4.6	保存 ShapeLabel 控件的属性值	(345)
14.4.7	给 ShapeLabel 控件添加一个属性页	(347)
14.4.8	把一个事件添加到 ShapeLabel 控件中	(351)
14.4.9	编译 ControlDemo 部件	(353)
15	访问数据	(354)
15.1	用 Data 控件访问数据库	(354)
15.1.1	Data 控件的功能	(354)
15.1.2	数据绑定控件	(355)
15.1.3	绑定控件的类型	(356)
15.1.4	创建简单的数据库应用程序	(357)
15.1.5	添加、更新和删除记录	(358)
15.2	使用数据绑定控件	(359)
15.2.1	使用 DBListBox 和 DBComboBox 控件	(360)
15.2.2	使用 DBGrid 控件	(361)
15.2.3	使用 MSFlexGrid 控件	(364)
15.3	使用代码操作数据库记录	(367)
15.3.1	定位记录集	(367)
15.3.2	查找记录	(369)
15.3.3	更新记录集	(370)
15.3.4	控制事务	(372)

15.3.5 确认数据库变化	(373)
15.3.6 Update 方法	(375)
15.4 访问文件数据	(375)
15.4.1 访问文件的函数和语句	(376)
15.4.2 顺序文件访问	(376)
15.4.3 随机文件访问	(377)
15.4.4 二进制文件访问	(379)
16 Visual Basic 的数据库编程	(381)
16.1 Visual Basic 数据库体系结构	(381)
16.2 关系数据库的设计	(382)
16.2.1 关系数据库的结构	(382)
16.2.2 设计 Biblio 数据库	(384)
16.3 创建数据库	(386)
16.3.1 创建数据库的步骤	(386)
16.3.2 添加索引和关系	(387)
16.3.3 创建 Biblio.mdb 数据库	(388)
16.4 从外部数据库中附加表	(391)
16.5 修改数据库	(391)
16.5.1 向数据库中添加表	(392)
16.5.2 删除表	(392)
16.5.3 向表中添加字段	(392)
16.5.4 更改或删除字段	(392)
16.5.5 向表中添加索引	(393)
16.5.6 删除索引	(393)
16.6 数据库的维护	(393)
16.6.1 映射数据库	(393)
16.6.2 压缩数据库	(396)
16.6.3 修复数据库	(396)
附录 词汇表	(398)

1

Visual Basic 6.0 的新特征

Visual Basic 6.0 作为 Visual Basic 5.0 的升级版本,并不像 Visual Basic 5.0 那样引入了大量新特征,甚至改进了大量核心函数。Visual Basic 6.0 相对于 Visual Basic 5.0,主要的改进是提供了新的、灵巧的数据库和 Web 开发工具,同时也可以创建功能齐全的 Windows 应用程序。

Visual Basic 6.0 提供了包括具有对象浏览器完备功能的 WebClass Designer 在内的大批值得夸耀的新特征。此外,用户还可以像对待传统 Visual Basic 6.0 窗体那样处理 DHTML Web 页面,只是所有这些 DHTML 标记都挂接着事件驱动程序。对数据库和 Web 开发人员而言,Visual Basic 6.0 是一个重要的升级版本。本章将介绍 Visual Basic 6.0 新引入的特征。

本章介绍的主要内容包括:

- 数据访问新特征
- Internet 新特征
- 控件新特征
- 部件创建新特征
- 语言新特征
- 向导新特征
- Visual Basic 6.0 的版本
- 升级使用了 Windows 通用控件的应用程序

1.1 数据访问新特征

Visual Basic 6.0 为数据访问提供了以下新增特征:

(1) ADO (ActiveX Data Objects):这是一种新的数据访问技术,具有以下特征:

- 有一个简化的对象模型。
- 与其他 Microsoft 和非 Microsoft 技术更加紧密的集成。
- 是一个本地和远程数据访问、远程和断开连接的记录集,用户可访问的数据绑定接口以及层次化的记录集的通用接口。

(2) 数据环境:数据环境设计器为创建 ADO 对象提供了一个交互的设计环境。它可以方便迅速地对数据源进行设置,并支持 Visual Basic 的用户连接(UserConnection)设计器的所有功能,以及包括拖放、分组和聚合在内的其他特征。

(3) 增强的数据绑定:在以前的 Visual Basic 版本中,数据绑定只支持对窗体的控件绑定。在 Visual Basic 6.0 中,可以将任何 ADO/OLE DB 数据源绑定到任何 ADO/OLE DB 数