

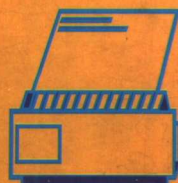
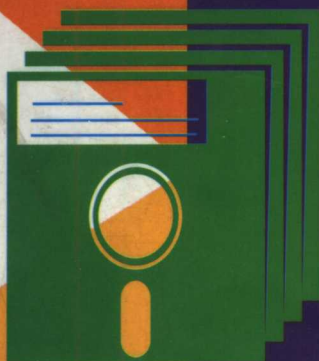
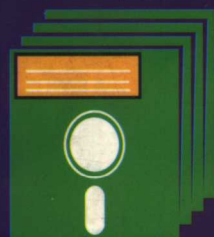
VENTANA

北京科海培训中心

Visual Basic 4.0

[美] Richard Mansfield 著
廖卫东 赵军 译

编程手册



机械工业出版社

VENTANA

北京科海培训中心

Visual Basic 4.0 编程手册

[美] Richard Manfield 著

廖卫东 赵 军 译

机械工业出版社

北京市版权局著作权合同登记号:01-95-212号

内 容 提 要

这是一本结构清晰、实用性强的 Visual Basic 4.0 技术手册。

全书分二篇,第一篇介绍 Visual Basic 4.0 新增功能,使用技巧、调试策略和图形浏览器实用程序的构造过程。

第二篇按字母顺序逐一介绍了 VB4 中 400 多个命令、函数的特性及其功能,以及实用工具的应用技巧,针对这些命令、函数提供大量的实例和图示。

本书旨在使你:掌握 VB4 新增功能,自行创建 OLE 对象、创建定制的 Windows 95 程序,学会使用 Windows API 中的 600 多个命令,掌握 VB 核心技术,学会用“JET EN”功能创建高效、功能强大的数据库管理系统。

本书既可作为 VB 初学者的实用教程,也是编程人员必备的随机手册。

图书在版编目(CIP)数据

Visual Basic 4.0 编程手册/(美)曼斯菲尔德(Mansfield,R.)著;廖卫东,赵军译. — 北京:机械工业出版社,1996.7

书名原文:The Visual Guide to Visual Basic 4.0 for Windows

ISBN 7-111-05226-9

I. V... I. ①曼... ②廖... ③赵... II. Basic 语言-程序设计-手册 IV. TP312BA-62

中国版本图书馆 CIP 数据核字(96)第 11598 号

出版人:马九荣 (北京市百万庄南街 1 号 邮政编码 100037)

责任编辑:科培 责任校对:成昊

门头沟胶印厂印刷·新华书店北京发行所发行

1996 年 6 月第 1 版·1996 年 6 月第 1 次印刷

787mm×1092mm 1/16·68.625 印张·1669 千字

0 001—5 000 册

定价:94.00 元

引言

在编写本书时,我首先考虑的是使其结构清晰,并具有良好的可读性。许多涉及计算机语言和程序设计方面的书都由于过份关注细节问题而导致整本书的结构混乱。本书的第二个主要目标是实用性,本书提供了大量实例。这些实例中的绝大部分不但演示了 Visual Basic 语言的某个命令或特性,而且还实现了某些在实际编程中真正需要的功能。

Visual Basic 为编制 Windows 应用程序提供了一种全新的、绝对简单的方式。既然 Windows 95 将成为个人计算机上的标准操作系统,那么 Visual Basic 就必须对在 Windows 95 (而不是 Windows 3.1)环境中进行程序设计所涉及的新结构和新技术作出响应。其中涉及的变化,包括 32 位取代了原先的 16 位,面向对象的编程方法取代了原先面向过程的编程,以及正在不断普及和应用的对象链接与嵌入技术(OLE)等。为扩充其自身以便包括这些新特性,Visual Basic 版本 4 采用了一些与 VB3 及其更早的版本截然不同的方式。它并不是一种新的语言,但却包括了一些非常新的方法及命令。鉴于在 VB 版本 4 中对 Visual Basic 作了较大的改动,在这个引言中我们将针对这些改动列一个清单,同时告诉你在本书什么地方能够找到有关新的命令和新的编程风格的更详细的介绍及实例。

本书的开头及结尾部分都是一些编程指南。其中第一篇介绍 VB4 中新增的内容。它涵盖了 Visual Basic 4.0 的所有主要特性——所有的菜单选项、编程及调试策略,以及一个图形浏览器实用程序的逐步构造过程。而在本书的最后,附录 A 解释了如何使用 Windows API(应用程序编程接口)以给 Visual Basic 增加新的功能。

第二篇是本书篇幅最大的部分,即“Visual Basic 指南”部分。这部分内容的组织有些类似于百科全书,其中对 Visual Basic 中的每个成分均作了详细阐述。在这里我们依字母顺序阐述了各命令的特性及其功能,以及 VB 的其他语言特性。可以说,这部分是一个可用于日常查阅的 VB 汇总信息表,它可以帮助你澄清有关 Visual Basic 的某些事项,并且激发编程人员的灵感。

在第二篇中的每个条目均包含如下 7 类信息:

描述:综述当前条目的实用功能以及它在 Visual Basic 总体模式中所处位置。

用于:与当前条目相关的其他 Visual Basic 成分。

变量:每个条目所需的语法格式,以及所有例外情形、可选和变化形式。这部分内容主要介绍如何编写在程序中引用当前条目并调用其功能的程序行。

用途:对当前条目所起的作用做一个现实合理的说明。VB 中的大多数元素都有一到多个用途,然而在有些条目下,你会看到在“用途”一栏中只有一个注释“无”(参见“Let”)。这是因为 VB 确实包括一些来自 Basic 早期版本的条目。

警告:需引起注意的事项,如为什么会出现异常的运行结果,以及规则的例外情形、冲突及解决措施等。

实例:许多计算机著作中的很多例子都不太现实,它们只提供有关的语句格式及标点,而不具任何真正的含义。本书将尽量避免这种情形,提供一些具备一定功能并可

展示命令或特性的例子,这些例子反应了命令或特性的真正用途和行为。而且这些例子已经过了测试。

参见:介绍相关条目及可选情形,或推荐更进一步的读物。

本书的读者对象

本书面向所有对编写 Visual Basic 程序感兴趣的人士,而不管他们在程序设计方面有过多少经验。正如字典既适用于小学生也适用于专家学者一样,相信本书也可帮助所有的人,无论他是一位初学者,还是一名经验丰富的高级程序员。

对于熟悉 VB 的人来说,本书对 For... Next 用法的介绍无疑太繁琐,虽然它对初学者可能很有帮助。然而,鉴于 VB 版本的不断升级和功能的扩充,即使是 VB 老手也可能会对书中下列方面感兴趣:新的属性过程、对象、集合、类模块、Line 在窗体设计中的应用以及 Link 命令的具体应用。同时,对于我们中的大多数人来讲,即使许多年来我们一直在用 Visual Basic 编程,有时也仍然需要一本结构清晰且能及时反映 VB 最新变化的读物(诸如反映 ClipControls, AutoRedraw 及 Paint 之间关系的细节)。我希望这本书能够成为编写 Visual Basic 程序时必备的读物。

Visual Basic 版本 4 中的新特性

下面列举 VB4 中的新增特性。本书其他地方出现的词汇或概念将以黑体形式给出。

1. 在窗体之间传递数据。VB3 及其早期版本可通过定义公共变量来进行窗体之间的通信。而在 VB4 中,你可以定义公共子例程或函数,然后从它所在窗体外部向其传递数据。
2. VB4 可以创建 VB 对象(Object),像 Excel 之类的外部应用程序可通过 OLE 机制直接调用这些对象。这些对象是用新的 VB4 类模块(Class Module)创建的,它们可以有自己的方法和属性。同时,VB4 还可创建为 VB 程序本身所用的对象。最后,类模块还支持用户运用面向对象编程中的各要素,如封装。此外,它还有一个新的嵌入式“对象浏览器”(Object Browser)。
3. 在传统的两类过程(子例程及函数)中又增加了 3 个新过程,分别为 Property Get, Property Let 或 Property Set。它们用于给根据某个类模块而创建的对象增加属性。
4. 目前的 VB 编辑器是可扩充的——编程人员可以编写或购买项目管理器或其他辅助编程工具,然后用 Add-Ins 功能把它们加入到程序开发环境。
5. 目前的网格控件是“数据联编”(data-bound)的,这意味着它可以连接到某一数据控件,并用于充当进入数据库的入口。目前的 OLE、列表框和组合框控件也是数据联编的。此外,在 VB4 中控件可以联编到另一个不属于同一个窗体的数据控件,而且你还可以在位于不同窗体的各控件之间共享记录集合。数据库语言已经更加丰富,而且增加了许多新技术。例如,当用户希望滚屏到文件的开始或结尾处时,你可通过 BOFAction 和 EOFAction 命令来告诉数据控件如何满足用户要求。参见“Data”。
6. 为了与 Windows 95 程序中富有吸引力的三维外观保持一致,VB4 为窗体新增了

Appearance 属性。作为一种改善文本框和其他控件的外观的方法,该属性不仅增强了大多数控件的外观,而且减少了我们对 SSPanel 和其他 Sheridan 3D 控件的依赖程度。

7. 窗体、图片框和 Printer 都新增了一个名为 PaintPicture 的方法。该方法与 API 的 BitBlt 调用的功能相同,但与为达到特殊的视觉效果而调用 API 相比,一些 VB 编程人员会觉得这个方法更为方便。
8. 当需要进行“即时编辑”(in-place editing)时,新增的 NegotiateMenus 和 NegotiatePositions 属性使你能够控制 OLE 对象的行为方式:在 Windows 95 操作系统下,当用户双击某一链接或嵌入的对象时,便可触发这种 OLE 现象。
9. 对控件的操纵。使用箭头键不仅可以改变各控件的位置,而且可以逐个像素地调节控件的大小。然后,当觉得窗体的外观完全合乎你的要求时,可单击“Lock Controls”按钮。这样,它们将被冻结在相应位置,从而不能移动。这些技术将在第一篇中阐述。
10. 对于 Font, Error, Common Dialog 和 Printer 对象新增了一些简化的、增强的操纵技术。
11. 多文档界面窗体已经有了一些附加性能。MDI 窗体新增加了 Picture 和 BackColor 属性,而且可以隐藏子窗体。
12. 目前的 VB4 包括两种版本:用于创建 Windows 3.1 应用程序的 16 位版本和用于创建 Windows 95 和 NT 应用程序的 32 位版本。向 32 位应用程序的转变并不会对 VB 编程人员造成太大的冲击。主要的变化出现在调用 API 之时——过去名叫“USER”的库现在已改名为“USER32”,过去的整型参数现已改变为长整型等等。这些不同将在附录 A 的例子中予以介绍。此外,VB 现在还包含了 Visual Basic for Application (VBA)——即用于替换所有 Microsoft 应用程序中原先的宏语言的一种新语言(Word 仍未进行这种转换)。
13. VB4 可以给已完工的 .EXE 代码插入自动递增的版本号、版本信息、注释、商标、描述等。你只需从 VB 的 File 菜单中选择“Make EXE File”,之后再选择“Options”菜单项,即可完成此项功能。这些将于第一篇中介绍。
14. 条件编译功能。这项 VB4 新增功能利用 #If...Then...#EndIf 来跳过并忽略位于 #If 和 #EndIf 之间的任何程序。如果条件为 False,则忽略相应的程序段。这种结构可用于创建同一个应用程序的多个版本。函数和子例程均可以使用条件编译,你还可以把多个过程包括到“条件区”中。
15. 一个新的编辑器。在 VB4 中,有两种方法可用来录入并修改程序。首先是一个 Visual Basic 专有的编程环境——从中你每次只能操作(实际上是看到)一个子例程或函数——但可用 PgUp 和 PgDn 键在各子例程或函数之间切换。这是一个从 Visual Basic 的第一版本开始一直使用的编辑器。与那些常见而又古老的程序编辑器相比,许多人都认为 VB 的这种“封装”方法是一种很鼓舞人心的改进,因为在老式程序编辑器中,全部程序都是连续的(就如同打印在一个巨大的滚筒纸上一样),程序员通过前后滚屏来定位各个过程。另一方面,VB4 也支持这种“连续”方式——你可以选用一种最适于自己的方式。此外,连续方式还可在各子例程或函数之间给出一个信息行,从而使你可以确定在何处是一个子例程的结束,下一子例程的开始又

在何处。为在这两个子例程之间切换,可单击 Tools 菜单,然后选择 Options 和 Editor,并改变“Full-Module view”选项。请注意,过程之间的分隔行也是可选的。

编辑器还有另一种很好的新特性:续行符。你可以在一行结束处加一个空格,后接一个下划线字符“-”,以分隔一个程序行。这时 VB 仍然将其视作一个连续行,但 VB4 还可用 ENTER 键分隔过长的行,以便清楚地看到完整的行,而不是在屏幕右端被切断。

以前:

```
Declare Function BitBlt Lib gdi32 Alias BitBlt (ByVal hDestDC As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal XSrc As Long, ByVal YSrc As Long, ByVal dwRop As Long) As Long
```

目前:

```
Declare Function BitBlt Lib gdi32 (ByVal hDestDC As Long, _  
ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, _  
ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal XSrc As Long, _  
ByVal YSrc As Long, ByVal dwRop As Long) As Long
```

为了使 VB 把它当作一连续行,必须在下划线字符之后紧接一个空格符(按空格键)。实际上,这并不是一个行连续字符,而是一对续行符。此外,这种技术还无法用于分隔一个文本“字符串”,如:Print “This line should be snapped.”。

16. 新增的 Collection(集合)对象可用于替代传统的数组,但它要比数组灵活。
17. 新增 Boolean(布尔)变量类型(只有两种状态:True/False)和 Byte(字节)数据类型(保存 0~255 之间的数值,可用于每次从磁盘文件中取一个字节。参见“Open”部分)。
18. 在 VB4 中,你可以为对象和一些方法命名参数。如有必要,你甚至可以为你所写的任何子例程或函数命名参数。这有两个优点:(1)它更具描述性,从而可使你的程序更易读;(2)可以重新组织参数的顺序。有关这方面的细节,请参例子例程、函数或命名参数部分。
19. 在 VB4 中,过程(子例程或函数)的参数可以是可选的。这使你的过程更通用,因为它们可以操纵不同数目的参数(参见“Sub”,“Function”或“IsQueries”部分)。另一方面,下面介绍的 ParamArray 又给参数传递带来了更多的灵活性。
20. 使用 VB4 中新增的 ParamArray 命令,你可以给子例程或函数传递任意数目的参数,这些参数可以是任意的变量类型,这增加了参数传递的灵活性。参见“Sub”或“Function”两节。
21. 从 VB3 开始,你便可以创建一个由 Variant 类型数据构成的数组,而在 VB4 中,你甚至可以在一到多个普通数组赋给 Variant 数组。
22. 在 VB4 中,许多常量定义是内置的。你既不必装入一个单独的预定义文件,也不必自行定义这些常量,它们随时都准备供你调用。为查阅这些常量,请按 F2 键并双击“Constants”。
23. 新增图片对象(同样请参阅新增的 ImageList 控件下的各项。这些新技术都是把随后需要显示的图形作为图形文件保存,而不是用非可视图片或图形控件来填充窗

体)。下面介绍一个关于如何使用图片对象的例子。在程序运行期间,每当你单击命令按钮时,VB 就将循环显示 4 幅. BMP 图片。

```
Dim Picobj(0 To 3) As Picture
Private Sub Command1_Click()
Static c As Integer
Picture=Picobj(c)
c=c+1
If c=4 Then c=0
End Sub

Private Sub Form_Load()
Pics=Array("tempx","hub5","tempp","t1")
For I=0 To 3
    n$="C:\\" & Pics(I)& ".BMP"
    Set Picobj(I)=LoadPicture(n$)
Next I
End Sub
```

如果想自己运行这个程序,可以用硬盘上的. BMP 文件名替换程序中的“tempx”, “hub5”, “tempp”和“t1”,然后在路径中加上“C:\”之类的内容。注意,上面程序中用到的 VB4 新增的 Array 函数。如果程序仍无法运行,可单击 Tools 菜单并选择 References 菜单项,看看其中的“Standard OLE Types”库是否已被选中。

24. With...End With 结构可更直接地设置多个属性。
25. For Each...Next 结构使程序可以在一个数组或对象集合的所有元素中循环,而无需事先知道它有多少个元素。
26. 用户定义类型现在可包括字符串类型的动态数组。
27. 在 4.0 以前的 VB 版本中,包含有关整个项目(程序)信息的文件带有扩展名 .MAK。除其他信息外,这个“项目文件”包含了项目中每一窗体的路径名和文件名。在 VB 4.0 中,该文件的扩展名为. VBP(代表“Visual Basic Project”)。
28. Windows 95 增加了许多新功能,如标签对话框(称为“属性表”,因为它们一般用于改变选项和参数)、一个滑动条(非连续滚动条),以及一个进度指示栏(一个非连续的状态指示器)等等。这 8 个新控件均被包括在 Windows 95 Controls 项中。此外还有一个新增的 RichTextBox 控件,该控件扩充了传统的 VB 文本框的性能——支持各种形式的格式化、字体变化、斜体、颜色以及其他类似字处理器的文本录入功能。有关这方面的细节,参见“TextBox”。

3 种版本的 VB4

VB4 是按 3 种版本销售的。

标准版(Standard Edition)包含了所有主要的功能,它主要是针对那些不需要重要的数据库编程机制、Windows 95 命令集以及其他类似于 3D 命令按钮和图形控件等的编程人员的。对于那些想尝试一下 VB 或者刚刚接触程序设计的人员而言,选购 VB 的标准版不失为一种好的选择。

专业版(Professional Edition)则包括:一个由 28 个附加控件组成的控件集合;一个报表生成器;一个“hotspot”编辑器,从中你可以在一幅图形内定义各个区(zone),当单击这些区时可触发帮助信息或宏;以及包括 SQL 语言在内的附加数据库支持。对很多编程人员而言,专业版几乎可满足其所有要求。

企业版(Enterprise Edition)是针对那些需要具有工业保障的数据库工具——尤其是需要网络支持的用户。它还包括了用于管理程序员组的工具。

后续内容简介

无论你是刚刚接触程序设计还是已经积累了数十年的编程经验,Visual Basic 都能使你的编程工作取得事半功倍的效果。与以前的任何编程语言不同,VB 不仅提供了内置的已预先编写好的功能体(控件),而且包含了一个非常灵活的编程机制(编辑器),同时所有这一切还都处于一个高度可视化的环境之中。除试图覆盖 VB 的广度之外,本书还将尽量探讨一些具有相当深度的内容。其中提供深层次探讨的目标是针对目前的程序设计中最流行的可视元素(参见“Line”),因为它们在许多关于程序设计的著作中仍未引起足够的重视。和其他方面一样,高度可视化的程序可提供一个更高效、并且最终更吸引人的用户界面。正如 Windows 优于 DOS 的主要原因在于它是一个图形化而非文本的环境,Visual Basic 从根本上说是可视的,并且可产生在图形方面更具吸引力的效果。

现在让我们转入正题——要么遵循本书前三个教程部分中对 VB 所作的循序渐进的综述,要么跳过其他部分,而专门尝试贯穿本书的各个实例。无论你采用哪种方法,也无论你原来的编程经验如何,Visual Basic 都能够打动你。对于大多数编程人员而言,在过去几年中 VB 已经成为他们用于 Windows 编程的首选语言。

目 录

第一篇 Visual Basic 程序设计指南

第一部分 打开窗口	(1)
第二部分 Visual Basic 的基本构件	(3)
第三部分 菜单	(13)
第四部分 创建 Visual Basic 程序	(32)

第二篇 字母顺序索引

A

Abs 函数	(42)
Activate, Deactivate 事件	(42)
ActiveControl 属性	(44)
ActiveForm 属性	(46)
Add, Remove 方法	(48)
AddItem 方法	(51)
Align 属性	(52)
Alignment 属性	(53)
App 对象	(55)
AppActivate 语句	(56)
Appearance 属性	(57)
Archive 属性	(60)
Arrange	(60)
Array 函数	(60)
Arrays	(62)
Asc 函数	(68)
Atn 函数	(70)
AutoRedraw 属性	(70)
AutoShowChildren 属性	(72)
AutoSize 属性	(72)

B

BackColor 属性	(76)
BackStyle 属性	(77)
Beep 语句	(78)
BorderColor 属性	(80)

BorderStyle 属性	(80)
BorderWidth 属性	(83)

C

Call 语句	(84)
Cancel 属性	(86)
Caption 属性	(87)
CBool&CByte 函数	(89)
CCur 函数	(89)
CDbl 函数	(91)
Change 事件	(92)
ChDir 语句	(95)
ChDrive 语句	(96)
CheckBox 控件	(96)
Checked 属性	(99)
Choose 函数	(101)
Chr \$ 函数	(101)
CInt 函数	(104)
Circle 方法	(105)
ClassModule 对象	(110)
Clear 方法	(120)
Click 事件	(121)
Clipboard 对象	(123)
ClipControls 属性	(125)
CLng 函数	(128)
Close 语句	(128)
Cls 方法	(129)
Collections 对象	(131)
Columns 属性	(134)
ComboBox 控件	(136)
Command Button 控件	(138)
Command, Command \$ 函数	(139)
Common Dialog 控件	(142)
Conditional Compilation	(150)
#Const	(150)
Const 语句	(150)
Control Array	(152)
ControlBox 属性	(159)

Cos 函数	(160)
CreateObject 函数	(160)
CSng 函数	(160)
CurDir \$ 函数	(160)
CurrentX,CurrentY 属性	(162)

D

Data 控件	(165)
DataSource,DataChanged 属性	(192)
Date 函数	(192)
Date 语句	(193)
DateSerial 函数	(193)
DateValue 函数	(194)
Day 函数	(195)
DblClick 事件	(196)
Debug 对象	(196)
Declare 语句	(197)
Default 属性	(202)
DefType 语句	(204)
DefVar 语句	(208)
Dim 语句	(208)
Dir 函数	(210)
Directory List Box 控件	(211)
Do...Loop 语句	(214)
DoEvents 函数和语句	(218)
Drag 方法	(219)
DragDrop 事件	(221)
DragIcon 属性	(223)
DragMode 属性	(224)
DragOver 事件	(225)
DrawMode 属性	(226)
DrawStyle 属性	(228)
DrawWidth 属性	(231)
Drive 属性	(231)
Drive List Box 控件	(233)
DropDown 事件	(237)

E

Enabled 属性	(238)
End 语句	(240)
EndDoc 方法	(242)
Environ \$ 函数	(243)
EOF 函数	(245)

Erase 语句	(246)
Err 函数	(247)
Error 语句	(250)
Error \$,Error 函数	(251)
EXENAME 属性	(252)
Exit 语句	(253)
Exp 函数	(255)

F

FetchVerbs 方法	(256)
File List Box 控件	(256)
FileAttr 函数	(258)
FileCopy 语句	(259)
FileDateTime 函数	(261)
FileName 属性	(262)
FillColor 属性	(264)
FillStyle 属性	(266)
Financial 函数	(269)
Fix 函数	(272)
Focus	(272)
Font 对象	(273)
FontBold 属性	(274)
FontCount 属性	(275)
FontItalic 属性	(276)
FontName 属性	(276)
Fonts 属性	(277)
FontSize 属性	(279)
FontStrikeThru 属性	(279)
FontTransparent 属性	(280)
FontUnderline 属性	(282)
For Each...Next 语句	(283)
For...Next 语句	(285)
ForeColor 属性	(290)
Form 对象	(292)
Format,Format \$ 函数	(299)
Frame 控件	(303)
FreeFile 函数	(306)
Function 语句	(307)

G

Get 语句	(315)
GetAttr,SetAttr 函数	(317)
GetData 方法	(320)

GetFormat 方法	(322)
GetText 方法	(323)
Global 语句	(324)
GoSub...Return 语句	(324)
GotFocus 事件	(326)
GoTo 语句	(328)
Grid 控件	(330)

H

hDC 属性	(334)
Height 属性	(337)
HelpContextID, HelpFile 属性	(342)
Hex \$ 函数	(344)
Hidden 属性	(345)
Hide 方法	(345)
HideSelection 属性	(346)
HorizontalScrollBar 控件	(347)
Hour 函数	(347)
hWnd 属性	(350)

I

Icon 属性	(352)
#If...Then...Else 语句	(355)
If...Then...Else 语句	(357)
IIF 函数	(362)
Image 控件	(362)
Image 属性	(368)
ImageList 控件	(369)
Index 属性	(369)
Initialize&Terminate 事件	(375)
Input # 语句	(378)
Input(\$), InputB 函数	(381)
InputBox(\$) 函数	(383)
InStr 函数	(386)
Int 函数	(392)
Interval 属性	(393)
Is 操作符	(400)
IsQueries 函数	(401)
Item	(403)
ItemData 属性	(403)

K

KeyDown,KeyUp 事件	(404)
KeyPress 事件	(408)

KeyPreview 属性	(410)
Kill 语句	(411)
KillDoc 语句	(412)

L

Label 控件	(412)
LargeChange, SmallChange 属性	(415)
LBound 函数	(419)
LCase \$ 或 LCase 函数	(420)
Left 属性	(422)
Left 函数	(427)
Len 函数	(429)
Let 语句	(433)
Like 操作符	(434)
Line 控件	(435)
Line 方法	(437)
Line Input # 语句	(450)
LinkClose 事件	(453)
LinkError 事件	(456)
LinkExecute 事件	(458)
LinkExecute 方法	(463)
LinkItem 属性	(463)
LinkMode 属性	(467)
LinkNotify 事件	(471)
LinkOpen 事件	(472)
LinkPoke 方法	(474)
LinkRequest 方法	(476)
LinkSend 方法	(477)
LinkTimeout 属性	(481)
LinkTopic 属性	(483)
List 属性	(488)
ListBox 控件	(490)
ListCount 属性	(493)
ListIndex 属性	(495)
LinkView 控件	(497)
Load 事件	(497)
Load 语句	(501)
LoadPicture 函数	(503)
Loc 函数	(508)
Lock 和 Unlock 语句	(509)
LOF 函数	(511)
Log 函数	(513)
LostFocus 事件	(514)

LSet 语句	(517)
LTrim \$ (LTrim)函数	(520)

M

Max 属性	(521)
MaxButton 属性	(524)
MDIChild 属性	(526)
Me 保留字	(526)
Menu 控件	(527)
Mid 函数	(530)
Mid(或 Mid \$)语句	(533)
Min 属性	(538)
MinButton 属性	(541)
Minute 函数	(541)
MkDir 语句	(542)
Module	(547)
Month 函数	(547)
MouseDown,MouseUp 事件	(550)
MouseMove 事件	(554)
MousePointer 属性	(558)
Move 方法	(561)
MsgBox 函数和语句	(566)
MultiLine 属性	(571)
Multiple Document Interface(MDI)	(572)
MultiSelect 属性	(579)

N

Name 属性	(580)
Name 语句	(583)
Named Arguments 参数	(586)
NegotiateMenus 属性	(590)
NegotiatePosition 属性	(591)
NegotiateToolbars 属性	(591)
New 属性	(592)
NewIndex 属性	(592)
NewPage 方法	(592)
Normal	(596)
Nothing	(596)
Now 函数	(596)

O

Object Browser	(598)
Objects 对象	(600)
Oct \$ 函数	(608)

OLE 控件	(609)
OLE Automation 控件	(624)
On Error 语句	(636)
On GoSub 语句,On GoTo 语句	(641)
Open 语句	(644)
OpenDatabase 方法	(654)
Option Base 语句	(654)
Option Button 控件	(656)
Option Compare 语句	(658)
Option Private 语句	(658)
Optional 语句	(659)

P

Page 属性	(659)
Paint 事件	(660)
PaintPicture 方法	(664)
Parent 属性	(666)
PasswordChar 属性	(668)
Path 属性	(670)
PathChange 事件	(672)
Pattern 属性	(674)
PatternChange 事件	(676)
Picture 属性	(677)
Picture Box 控件	(680)
Point 方法	(688)
PopupMenu 函数	(689)
Preserve 属性	(692)
PrevInstance 属性	(692)
Print 方法	(693)
Print # 语句	(698)
Printer 对象	(699)
PrintForm 对象	(704)
Private	(705)
ProgressBar 控件	(705)
Property Let,Get&.Set 过程	(706)
PSet 方法	(719)
Public 语句	(724)
Put 语句	(729)

Q

QBColor 函数	(731)
QueryUnload 事件	(735)

R

Raise 语句	(737)	SetText 方法	(839)
Randomize 语句	(738)	Sgn 函数	(840)
ReadOnly	(741)	Shape 控件	(841)
Redim 语句	(741)	Shape 属性	(842)
Refresh 方法	(745)	Shell 函数	(844)
Rem 语句	(750)	Show 方法	(847)
Remove 方法	(752)	Sin 函数	(850)
RemoveItem 方法	(752)	Slider 控件	(850)
Reset 语句	(755)	Sorted 属性	(851)
Resize 事件	(756)	Space \$ 函数	(852)
Resume 语句	(759)	Spc 函数	(853)
Return 语句	(761)	Sqr 函数	(854)
RGB 函数	(763)	Startup Form 对象	(855)
RichTextBox 控件	(768)	Static 语句	(856)
Right, Right \$ 函数	(768)	StatusBar 控件	(860)
Rmdir 语句	(772)	Stop 语句	(861)
Rnd 函数	(774)	Str(Str \$) 函数	(862)
RSet 语句	(780)	StrComp 函数	(864)
RTrim, RTrim \$ 函数	(782)	Stretch 属性	(866)
S		String(String \$) 函数	(867)
SavePicture 语句	(783)	Style 属性	(869)
Scale 方法	(784)	Sub 语句	(872)
ScaleHeight, ScaleWidth 属性	(788)	Switch 函数	(883)
ScaleLeft, ScaleTop 属性	(794)	System	(884)
ScaleMode 属性	(799)	T	
ScaleWidth	(803)	Tab 函数	(884)
Screen 对象	(803)	TabIndex 属性	(886)
Scroll 事件	(805)	TabStop 属性	(890)
ScrollBar 控件	(806)	TabStrip 控件	(892)
ScrollBar 属性(在文本框、网格 和 MDI 窗体内)	(813)	Tag 属性	(892)
Second 函数	(814)	Tan 函数	(893)
Seek 函数	(816)	Text 属性	(893)
Seek 语句	(817)	Text Box 控件	(896)
Select Case 语句	(818)	TextHeight 方法	(903)
Selected 属性	(824)	TextWidth 方法	(908)
SelLength, SelStart, SelText 属性	(825)	Time(Time \$) 函数	(910)
SendKeys 语句	(829)	Time \$ 语句	(912)
Set 语句	(832)	Timer 控件	(916)
SetAttr	(835)	Timer 事件	(924)
SetData 方法	(835)	Timer 函数	(931)
SetFocus 方法	(836)	TimerSerial 函数	(932)
		TimeValue 函数	(935)

Tile 属性	(937)
ToolBar 控件	(937)
Top 属性	(938)
TopIndex 属性	(941)
TreeView 控件	(942)
TwipsPer PixelX, TwipsPer PixelY 属性	(942)
Type 语句	(942)
TypeName	(947)
TypeOf	(949)

U

UBound 函数	(949)
UCase(UCase \$)函数	(950)
UnLoad 事件	(951)
UnLoad 语句	(954)
UseMnemonic 属性	(959)

V

Val 函数	(959)
Value 属性	(961)
Variables	(963)
Variant	(977)
VarType 函数	(983)
VBA	(983)
Vertical Scroll Bar 控件	(986)
Visible 属性	(986)

W

Weekday 函数	(989)
While...Wend 语句	(997)
Width 属性	(1000)
Width # 语句	(1005)
WindowList 属性	(1006)
WindowState 属性	(1006)
Windows95 控件	(1011)
With...End With 语句	(1025)
WordWrap 属性	(1026)
Write # 语句	(1028)

Y

Year 函数	(1029)
---------------	--------

Z

ZOrder 方法	(1031)
-----------------	--------

第三篇 附录

附录 A Windows 应用编程接口	(1035)
附录 B 消除错误	(1060)
附录 C 运算符	(1072)
附录 D Visual Basic & Windows 的 附加产品	(1080)

第一篇 Visual Basic 程序设计指南

第一部分 打开窗口

多年来,编程人员曾经面对许多种计算机“语言”——各种用于告诉计算机“做什么”的“语言”。其中,Visual Basic 的功能要比其他更古老的语言强大得多,因为 Visual Basic 程序中的许多构件——尤其是最具技巧性的可视化构件——都存在现成的程序模块。Visual Basic 无疑是一个突破。

Visual Basic (VB) 提供了一种功能非常完善的嵌入式智能工具,这些工具使得编写 Windows 环境下的程序变得异常简单。在程序中只需把一个完整的目录列表框“拖曳”(drag and drop)到窗口,所有的目录及其子目录就将自动可视。如果单击框中的某个目录,则该列表框将自动作出响应,切换到该目录并显示其所有的子目录。而在创建这个功能完善的目录列表时,不需要进行任何编程工作。

Visual Basic 用起来通常是非常有趣的,而且它的功能也很强大。目前 VB 拷贝已经卖出了一百多万份,它被广泛应用于编制商业软件、共享软件及其他用途的程序。使用 VB,你可以编写出高效而又精致的软件,并使其在每个细节之处都达到商业应用软件的程度。与此同时,与 C 或其他计算机语言相比,Visual Basic 程序的编写要容易得多。

设计而非编写

对于一个刚刚开始接触程序设计的人来说,用 Visual Basic 编写程序的第一阶段似乎更像是在设计一幅图画,而不是编制一些要求机器遵照执行的晦涩难懂的、半数学化的指令。如果你曾经使用过更原始的语言,那么这一点显而易见:在诸如 Windows 之类的高级环境下编程要比更原始的 DOS 环境下容易。

Visual Basic 包含了相当多的内嵌特性,这使得程序用户界面部分的建立很像是从一个目录中挑选一个合适的构件,而不是自己从零开始一步一步地建立这个构件。

而且,Visual Basic 的工具及自定义控件并不仅仅是一些简单的花样,它们还包括了实际功能。若要了解究竟是什么使 VB 如此特别,最快捷的方式也许就是把它想象成一堆用于组装工业机器人的零部件。你只需在所见到的程序表面选择自己所需要的零部件。

Visual Basic 提供了与计算机交互所必需的所有可视化组件:用于在栏中自动按字母顺序显示表项的列表框、滚动条、大小可调的窗口以及按钮等。

这些工具具有内嵌的功能:一个文本可自动回绕的文本框。该文本框还能对各方向的箭头键(上、下、左、右键)以及 Backspace, Del, Enter, Capslock 和 Shift 等键作出响应。在 C 语言中实现这样一个文本框可能需要数天的编程工作,而在 VB 中把它放到窗口则只需几秒钟的时间。

此外,通过从 Properties Window (属性窗口)中选择各参数配置,还可以定制其中的每种工具。如针对文本框属性的一些选择包括 BackColor, BorderStyle, DragIcon, Enabled,

FontBold, FontSize, ForeColor, Height 和 Width, Index, LinkMode, MousePointer, Name, ScrollBars, TabStop, Text 及 Visible。

若要改变一个文本框的背景颜色,则只需用鼠标单击 BackColor,然后通过调色板进行选择:

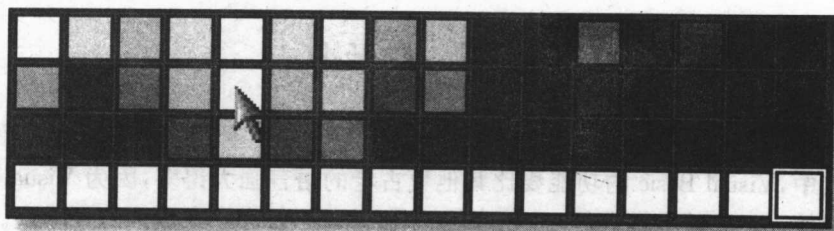


图 1.1 用鼠标改变颜色

此外还可以单击 View Menu 并且选择 Color Palette,以创建自定义颜色。

全面调整各个对象:若希望改变 5 个标签的颜色,可以用鼠标框住这些对象并将其作为一个组选定。这实际上就是在图形化程序中选择一组可视化对象的方法(或者也可以通过按住 Shift 键并单击每个你所希望改变的对象而选择这些标签)。这时 VB 的“属性窗口”(即改变对象的大小及颜色等属性的地方)将自动显示这些选定对象所共同具有的属性。在选择完你希望改变其颜色的所有标签之后,只需单击 Properties Window BackColor 项,就将出现一个调色板。单击所要的颜色,这时所有选定的标签都将相应改变。

若要在给定窗体内或者从一个窗体往另一个窗体拷贝一组选定的对象,或者希望剪切并粘贴这些对象,那么要做的工作与在一个设计良好的字处理器内拷贝并粘贴文本一样简单——你完全可以用 Word 中的 Windows 约定来删除、剪切、拷贝或粘贴这些对象,即先单击其中的某个对象,如果希望操纵一组对象,就用鼠标框住这些对象(请记住,如果不希望用鼠标框住这些对象——这也许是因为它们并未聚集在一块,则可以按住 Shift 键并单击鼠标。这样可以把一个对象增加到一个选定的组中)。

然后,与在字处理器中的情况完全相同:

- Del 键可用于删除对象或组。
- Shift+Del 可用于实现剪切功能(删除并且拷贝到剪贴板,以备随后的粘贴动作所用。这是用于把对象由一个窗口移向另一窗口的方法)。
- Ctrl+Ins 将把选定对象拷贝到剪贴板,以用于随后的粘贴动作。请注意,这时对象的属性(如颜色、宽度以及正文字体等)将同时被拷贝。
- Shift+Ins 用于粘贴对象。

如果你熟悉某种非 Windows 约定,那么可以使用一个类似的纯键盘命令集合。这些命令模仿 Windows 命令的行为,如 Ctrl+C 用于拷贝,Ctrl+V 用于粘贴等。

在设计阶段的绘制过程

Visual Basic 的一个非常重要的突破是改变了常规的程序设计方法。相对于花上大量时间去编写一些用于告诉计算机“如何响应”及“怎样显示”的程序指令,这里你只需简单地“绘