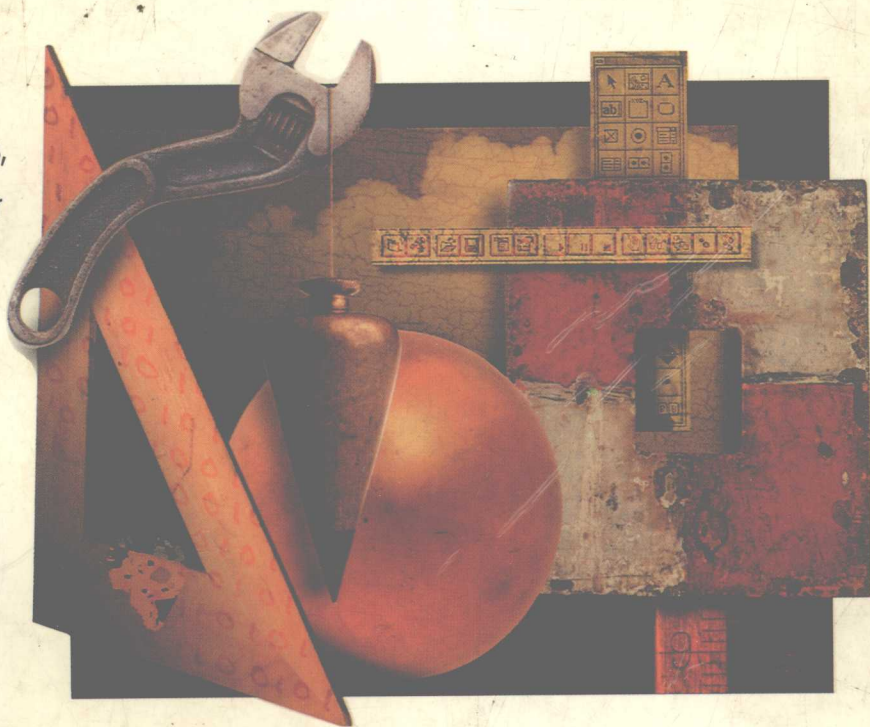


Visual Basic 4

大全

Secrets, Shortcuts,
and Solutions for
Programming
Windows®
Without Using C



[美] B. 麦金尼 著

Microsoft Press

科学出版社
龍門書局

Visual Basic 4 大全

[美] B·麦金尼

著

李 华 周 枫

译

万 博

校



科学出版社

龍門書局

1997

内 容 简 介

本书介绍 Visual Basic 4 的编程和使用方法,其特点是:针对程序设计者的实际需求介绍 Visual Basic 的特色和包括 OLE 在内的高级用法,内容循序渐进,示例丰富,使得 Visual Basic 的开发者能简单快捷地开发出 Windows 3.x, Windows 95 和 Windows NT 应用程序。本书所描述的内容适合于从初学者到专家,是学习 Visual Basic for Windows 95 的一本好书。

需要本书的读者可直接与北京海淀 8721 信箱书刊部联系,电话: 62562329, 邮编:100080。

版 权 声 明

本书英文版名为《Hardcore Visual Basic 4》,由 Microsoft Press 出版,版权归 Microsoft Press 所有。本书中文版由 Microsoft Press 授权出版。未经出版者书面许可,本书的任何部分都不得以任何形式或任何手段复制或传播。

图书在版编目(CIP)数据

Visual Basic 4 大全/[美]麦金尼(Mckinney,B)著;李华、周枫译,

—北京:科学出版社,1996

书名原文:Hardcore Visual Basic 4

ISBN7-03-005482-2

I. V… II. ①麦… ②李… ③周… III. BASIC 语言—手册 IV. TP312BA

中国版本图书馆 CIP 数据核字(96)第 16100 号

科学出版社 出版
龙 门 书 局

北京东黄城根北街 16 号

邮政编码:100717

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

*

1997年2月第 一 版	开本:787×1092 1/16
1997年2月第一次印刷	印张:29 3/8
印数:1—5000册	字数:68300

定价:40.00 元

前 言

在一个工作日的深夜,电话铃响了。BruceM 的名字和他的最新的 Visual Basic 内部用户的 e-mail 别名显示在我的电话的 LCD 显示屏上。随之而来的是 45 分钟的关于最新的 Visual Basic 4 版的某个不足之处的讨论。我竭力抑制住自己想不予理睬的冲动。Bruce 在研读他的书时总是一头钻进那些代码中并找出一大堆必须改正的错误,而且改得越快越好。我向后一靠,提起话筒:“你好,Bruce,让我猜猜……”。在 Microsoft,Bruce Mckinney 是 Visual Basic 高级用户的一个主要的倡导者,并且实在是最顽固的一个。每一个产品都需要有一个内部的批评家,而 Visual Basic 幸运地拥有 Bruce 这样富有献身精神而又令人厌烦的人。

作者在 Microsoft 的职业生涯好比是一条与一些引人注目的人(即使不算古怪,至少是有点独特)的关系织成的形状很不规则的挂毯。一个曾做过 Microsoft FORTRAN 工作,后来做 Visual Basic 工作的人,在他的工作中不时表露出的原来的工作对他们的影响则成了这条挂毯中经常重复出现的花纹图案。Bruce Mckinney 和作者第一次相遇时,他刚刚成为 FORTRAN 运行时(run-time)开发者,而我是支持部门的领导者。他对可用的开发工具是由什么构成的这一问题的强烈见解很快就显现出来,并且推动了许多长时间以来就提出要求的运行时特性的实现。Bruce 编的代码都是能派上用场的,而当他使用的工具不能满足他的要求时,他就修改它们。

FORTRAN 和 Visual Basic 之间一个明显的联系是二者结构的设计原则都是避免直接使用指针。这就导致了与以 C 语言为中心设计的操作系统打交道时受到很大的限制,以及其他的一些问题。于是,两种语言都吸引了那些把这些限制看成挑战的人们——看起来是一个路障,但可能成为一条捷径。尽管所有的文档都说不行,但读者还是可能会惊喜地发现,确实可以找到办法来实现一些被禁止的功能。读者会因为无视既有规则而完成一件事从而获得满足。读者手中的这本书出自一位不能忍受任何障碍的反传统者,他的与众不同的软件设计思想是不受任何限制的。本书中的代码、技巧和提示是对 Visual Basic 4 提供的新特性进行长时间研究的结果。

作为一个 Visual Basic 的开发主管,作者应该提出一些不那么明显的问题:“为什么使用 Visual Basic 来实现什么东西都那么困难?为什么不能想做什么就做什么?”大多数关于 Visual Basic 的用户查询电话并不是集中在因没有指针或其他 Basic 语言本身的限制而无法创建递归的数据结构,而是集中在与无法用一种透明的方式来调用所有的操作系统功能相关的问题上。Visual Basic 是特别设计的,以便把创建一个 Windows 应用程序所需的底层细节抽象出来。这就意味着将特定的功能封装到高层的构件中。因为封装的目的在于更好地组织系统,而成功的组织有时需要简化一些东西,我们必须经常就一种功能是否有趣并且足够重要而必须给予支持作出决定——否则,我们实现的将是一个含糊不清的结果。Visual Basic 作为一个 Windows 开发工具所获得的巨大成功表明简化所带来的好处超过局限性带来的坏处。

高水平的用户往往在使用 Visual Basic 内置的功能的同时使用大多数的 Windows API

函数。这就引出了一种两个层次的方法：一个层次是 Visual Basic 的核心功能，以另一层次的直接调用操作系统接口为补充。非常大、高水平的应用程序的成功开发也只用到了这种两层次的方法。但是，这种方法已经不那么行得通了。因为它无法提供函数指针以便将 Visual Basic 过程登记为回调函数，它没有提供对 Windows 一般消息的处理机制，它无法为 API 调用提供函数指针，它没有对直接使用指针的支持，这样在处理带有指针的 C 结构时就受到一定限制。

从 Visual Basic 设计的历史来看，可以得出一个结论，如果要实现一个关键的功能，则它应该是系统内置的而不是外挂的功能，应该将它封装起来以便所有的 Visual Basic 用户都可以使用——这比揭示那些只能由数量有限的专家使用的，没有经过加工处理的操作系统的某些基础特性有用得多。对于如何达到如此困难的第三层次，最常听到的建议是结合使用 C/C++ 和 Visual Basic，而不是将 Visual Basic 变成一种有局限性的“C”。本书提供了另一种很有用的方法：一个带有足够的使用信息的功能调用的核心集合，可以让高水平的用户实现他们想要的任何功能。

Visual Basic 4 将更方便、更简单地创建基于 Windows 程序的思想带入了新操作系统的领域——自动的 OLE。Visual Basic 开始是作为一个快速应用程序开发 (Rapid Application Development, RAD) 的集成环境，它的优点是快速的用户界面构建，它将成为构造一个功能齐全的数据库最快的一种方法。现在看来，它还将成为创建自动的 OLE 服务器与客户的最重要的工具。为此，我们将 Visual Basic 整个重新设计了，Visual Basic 和它的各组成部分之间的通信也都使用自动化的 OLE，其中最引人注目是面向应用程序的 Visual Basic (VBA) 引擎，Jet 和数据存取对象 (Data Access Objects) 引擎，以及 OLE 用户控件。在开发 Visual Basic 4 的过程中，Microsoft 完成了两个 32 位的主流操作系统，Windows 95 和 Windows NT 3.51。Visual Basic 通过充分使用这种技术来提高对这两个新操作系统的 OLE 和自动 OLE 支持。使用 C++ 自动 OLE 的开发者们现在转向 Visual Basic，将它作为一个对象原型或客户测试引擎的开发工具。

支持自动化的 OLE 导致一个有趣的结果：Visual Basic 语言作出了一个重大的、堪称典范的面向对象支持的转移。这是一些新的需要所共同造成的：Visual Basic 需要变得更为面向对象，它也需要更彻底地支持自动 OLE。结果是 Visual Basic 4 类所定义的对象总是自动 OLE 对象，而不像 C++ 那样有一般类对象和 OLE 对象之分。只要改变一个属性，一个内部类模型就可以提供公用的 OLE 组件。Bruce 在本书中通过实际使用 Visual Basic 对象，对这一特性潜在的可能性和意义作了分析。他还给出了一些便于产生对象代码的工具，而且就对对象编程风格和潜在的陷阱提供了宝贵的评述。

本书既是高级用户手册，也是开发工具使用手册，还是作者个人在用 Visual Basic 编程时所遇到的挫折和获得的快乐的经验汇集。因为有多年的使用许多种不同的软件开发工具的经验，Bruce 所能提出的有价值的观点几乎是无穷无尽的。这正是我们所需要的。所有用来传递知识的信息都是通过“观点”这个棱镜折射出来的。而对于技术上的信息，我发现鲜明的观点总是更可取的，它使读者不用去仔细研读代码，并被迫去迅速增强自己的洞察力。我相信高水平的 Visual Basic 开发者们会发现这本书不仅有用而且有趣，且是具有挑战性的。

John Norwood

Visual Basic 生产组程序管理员

鸣 谢

我感觉就像得了奥斯卡金像奖一样。当然,没有人会给我发奖,但我还是得到了一个机会来提起几个名字,对他们表示一下诚挚的感谢,并追忆一下过去的时光。

你好,妈妈;你好,爸爸。Chuck 和 Kandace,你们好。感谢 Rhonda, Sarah, Sam 和 Peggy,当我在写这本书的时候,或者说,我正在学写书的时候,他们隐忍迁就我。

我有一些写作上的良师。感谢 Linda Jaech 和 Stewart Konzen,在我还几乎没有经验(至少在写作方面)时雇用了我。感谢 Bill Johnston 帮助我成为一个记者,虽然不是一个很好的记者。还要感谢 Strunk 和 White。

至于编程上的导师,我要提到 Cleon white,他是我所遇到的最好的老师。直到我在大学低年级时决定要成为一个著名作家而不是一个工程师之前,White 先生一直在尽量往我愚钝的脑袋里填塞尽可能多的数学知识。对不起,White 先生,每天我都这样感觉。

接着是 Rich Gillman,他相信我是个够格的程序员,当时几乎没有人这么认为。还有 Gene Apperson,他对 Joe Hacker 说:“如果你的代码不能正常工作,那么它再快又有什么用呢?”这本书将会证明这句话是不对的。感谢 Jose Oglesby,在我想放弃 FORTRAN 时鼓励了我,使我坚持了下来。感谢我所喜爱的两位编程者 Steve Maguire 和 Michael Abrash。

我想把这本书的 Basic 部分(还有所有其他部分)献给一份 e-mail 别名表上的所有那些人,他们要求我不要提起他们的名字。Microsoft 持有这份关于 Visual Basic 问题讨论的内部别名表。这份别名表上的人大多是 Microsoft 的雇员,他们中的许多人在制作 Microsoft 的内部工具。有的是为其他公司工作,但因为协同工作或其他关系而列入 Microsoft 的 e-mail 表中。在这份别名表中,每个人都提出问题,不过答案也是由这些人们给出的。其中一些人是编写 Visual Basic 手册或部分章节的专家。其他人则是像读者掌握了本书精髓后那样成为专家的,因为他们不会接受“不可能”这样的回答。

下面还要提起一些人给我的帮助:Adam Overton; Antonio Maraglino; Benjamin Bourderon(分配器条), Bob Guy(减少模块数), Bruce Ramsey; Bryce Ferguson(分配器的帮助); Chris St, Valentine; Curt Carpenter; David McCauley; Dennis Quinn; Ed Staffin(Basic 的领袖人物,《Message Blaster》一书的作者); Eric Ledoux; Francis Gan; Fred Freeland; Gary Lemmon; Geoff Bauman; Hai Nguyen; Hubert Daubmeier; Jeff Webb; Jim Cash(提出关于模块的建议及其他); Jim Collins(检查出许多关键性错误); Joe Kennebec; John Schwabacher; Josh Kaplan; Karl Garard; Keith Pleas; Kenneth Lassen(出了许多主意和一些好的意见); Lewis Chapman(纠正了作者非随机“洗牌”的错误); Mark Buettemeier; Nick Baker; Patrick Dengler; Randall Kern; Rhy Mednick; Rob Purser; Robert Scheffler; Roman Lutz(屏幕截取); Ron West; Rutherford Le Blang; Scott Butler; Scott Ferguson(帮助我回溯到 Visual Basic 以前的历史); Sonja Al-kass; Stephen Weatherford(指出了 Notify 中一个关键的错误); Steve Anonsen; Steve Glenner; Stere Hayr; Steven Mitchell; Ted Keith; Tim Brown; Troy Strain; Vernon Lee; Wayne Radinsky(在划分子类上的无偿帮助); 还

有 Wayne Torman。

在 Visual Basic 小组中,一些人容忍了作者的“否定的态度”,并帮助作者理解了 Basic 的面向对象编程的能力和限制。特别感谢面向应用程序的 Visual Basic(VBA)小组的 Craig Symonds 和 Visual Basic 小组中的 John Norwood 和 Joe Robison。Matt Curland 是编制图形和 OLE 向导程序的奇人,提供了许多本书中的核心技术中的最核心部分。还要感谢 Arman Gharib; Betty Chin(计算机设置方面的专家); Brent Aliverti(在工具栏按钮,位图和图标等方面给予帮助); Brian Lewis(P 代码和内存分配方面的帮助); Chris Dias; Dirk Hawthorth, Doug Franklin(类型库专家); Glenn Hackney(对象和 OLE 服务器方面的领袖人物); 作者的老友 John Burke(错误处理); Lauren Feaux; Mark Roberts(集合); Michael Bond(以参数传递的 UDT(用户自定义类型)类型); Peter Golde; DoEvent 和初始化次序); Philippe Nicolle(图标); Prashant Ramanathan(设置); Steven Lees (LastDllError 函数); Tim Patterson(字节类型处理和 VarPtr 函数;还有 Todd Apley)。

Windows 95 小组的 Francis Hogle 帮助作者理解了 shell 编程的精髓。作者很遗憾没有用上它,而是使用了 Visual Basic 中的方法。作者也从 Nancy Winnick Cluts 写的《Programming the Windows 95 User Interface》(Microsoft 1995 出版)一书中学到了许多东西。作者很幸运能在这本书出版之前就借用了书中的程序。

对 Microsoft 出版社,作者要感谢编辑 Mary Renaud。她使我省了许多事,虽然这一切并不像作者所害怕的那样麻烦。她常常加班,甚至比作者还晚下班。感谢 Dail Magee Jr., 作者的技术编辑,还有 Eric Stroo——作者大多数项目的上司和其余项目中的责任编辑。感谢 Jim Fuchs,他做了 setup 程序,并常常和作者谈论一些和 Basic 很少有什么关系的技术问题。感谢 Peggy McCauley 的 stir-fry 和其他的主意。感谢 Marc Young 的猪尾巴。还要感谢下列人员:艺术家 David Holter 和 Michael Victor,校对员 Jennifer Harris, Sally Anderson, Sally Anderson, Lisa Theobald 和 Stephanie Marr; 排版和制作专家 Barb Runyan, Jim Kramer, David Mighell, 特别是 VersaTech 协会的 Sybil Ihrig 和“Aardvark 字处理”的 Tara Fort。编辑 Janna Hecker Clark 和编制索引的 Lynn Armstrong。

还要感谢所有那些住在假想的 Swinen 和 Cathistan 的最终用户,虽然这谢意无法传达到他们那里。

目 录

引论	1
0.1 没有限制的 Visual Basic	1
0.2 本书讲述范围	2
0.3 附带的 CD-ROM	5
0.4 不必太紧张	8
第一章 Basic 的灵魂	9
1.1 语言的净化	9
1.2 数据类型	14
1.3 命名约定	18
1.4 高效代码	23
1.5 自己断言	28
第二章 Basic 的第二层次的启蒙	35
2.1 编写可移植的代码	35
2.2 从 Basic 调用 Windows API	38
2.3 API 合约	44
2.4 处理字符串	56
2.5 处理指针	67
2.6 忘掉刚才所学的东西	71
第三章 老式的 Basic 工具	76
3.1 代码的回顾	76
3.2 Win16 的 Win32 API 版本	84
3.3 关于位操作	92
3.4 当已有方法都失效时	100
3.5 排序、随机化和搜索	104
第四章 Basic 的对象方法	114
4.1 面向对象编程, Basic 风格	114
4.2 设计类	117
4.3 表单类	130
4.4 群对象	138
第五章 控制 Windows	151
5.1 WinWatch 程序	151
5.2 Windows 概要	154
5.3 关于程序	172
5.4 收发消息	184

第六章 画图	196
6.1 从 Windows 方法到 Basic 方法,再从 Basic 方法到 Windows 方法	196
6.2 绘图的两种方法	203
6.3 Basic Windows 画图	211
6.4 关于位块传输(Blitting)	214
6.5 动画图片	225
6.6 一起参与	233
第七章 数据的处理	240
7.1 处理数据的三种方法	240
7.2 数据初始化问题	241
7.3 使用自己的资源	246
7.4 使用其他人的资源	253
7.5 使用资源的 Windows 方法.....	257
第八章 不要重复编写代码	272
8.1 Edward 和 CEditor 类.....	272
8.2 编辑器控件	277
8.3 扩展 CEditor 类	282
8.4 通用对话框扩展	289
8.5 Find 和 Replace 扩展	297
8.6 CKeyState 类	307
第九章 炒菜	311
9.1 你一直想知道的每一件事	311
9.2 CSortedListBox 类	323
9.3 其他人的菜单	329
9.4 栅格框	339
9.5 过滤多态方法	342
9.6 其他人的程序	349
第十章 OLE 准则	355
10.1 Visual Basic 对 OLE 准则的扩展	356
10.2 质数服务器.....	358
10.3 重新设计 Visual Basic	373
10.4 Add-In 挑战	397
第十一章 新的前沿	401
11.1 转向 Chicago	402
11.2 新的样子和感觉.....	406
11.3 通用控件.....	435
11.4 对期盼已久的 32 位的探索	445

引 论

如果用户正在以 10 个你喜爱的基于 Windows 的应用程序为样板学习编程,你可能有这样的想法:

“我想要用这一部分。它编得不错,但我想作一点改动。如果我从这个程序里取一点,从那个程序里取一点,再加上一点自己的东西将会怎样呢?”当用户想在 Visual Basic 中做这些事情时,有时可以很快达到目的。

Microsoft Visual Basic 使得用户的百分之九十的应用程序容易写了,但当用户去对付那剩下的百分之十时,它好像又总是跟你闹别扭。对于像修改控制菜单这样看起来十分简单的事,实际上却无法办到。这种语言对排序、查找、或语句分析所提供的支持很少。一个简单的思路需要几个小时的研究和试验,而获得的成果很少。

让我们正视这个现实:Visual Basic 是一种有限制的语言。最终用户会想去做一些 Visual Basic 做不了的事,或者用户会想更快地做一些 Visual Basic 做得很慢的事,你可以理解为什么这样简单的语言不得不在复杂性和功能之间作出一些折衷——但你不愿接受它。必须找个解决办法。

0.1 没有限制的 Visual Basic

本书是给那些喜欢 Basic 但不喜欢限制的人看的。如果用户愿意为了更好的性能和更强的功能而花点力气的话,会觉得这本书很有趣味。

0.1.1 超越 Windows API 的限制

在 Microsoft Windows 以前的 MS-DOS 时代,在 MS-DOS 下开始编程是很简单的,当用户使用 Basic 时更是如此。在用户还不真正理解在做什么的时候,用户已经可以获得很大的进展。但如果用户想不受限制地编程,就不得不去学习关于低端内存、端口和 MS-DOS 及 BIOS 中断的知识。Visual Basic 也是这样,用户必须付出更多的努力,才能不受限制。

在 Windows 编程中,付出努力意味着:学习消息、窗口、过程以及 Windows API 的所有其他因素。对于 Basic 编程者来说,遗憾的是,他们不得不学习另一种语言——C 中的大部分东西。与 MS-DOS 时代不同,那时用户几乎可以用 Basic 做任何事,而如果用户确实需要用另外的语言,可以根据需要选一种。许多 Basic 程序员选择汇编语言,而不是 C。

OLE 对象技术使得事情更加复杂。虽然用 C 来写 OLE 代码也可以,但头脑清醒的程序员们都选择 C++。只懂得一点 C 的 Visual Basic 程序员可能会大吃一惊:一点儿 C 是不足以使你能够写出真正强有力的 Visual Basic 4 版本的 OLE 控制和 OLE 服务代码的。虽然 C 和 C++ 之间有很大的差别,但当作者谈到其他语言的时候总是用 C 这个词来指代这两者。

本书试图把对 C(和 C++)的要求限制到最小。用户中的一大部分选择 Visual Basic 是因为他们不懂 C,不喜欢 C,或不喜欢 C 的 Windows SDK 编程模式(或者类似的 C++

MFC 模式)。因此,C 和 C++ 在这里不是必需的。虽然这本书和与本书配套的 CD-ROM (另售)中包含着一些 C 代码,但用户并不需要一个 C 编译器。所有这些用 C 写的工具都做成了动态链接模块或可执行文件了,用户可以直接使用它们。这本书的一个前提是,使用所提供的这些工具,用户可以用 Visual Basic 做任何事。是的,几乎是任何事。但这并不意味着用户可以快速地做所有的事。Visual Basic 运行的是 P 代码,不是用 C 产生的可执行代码,对此我们毫无办法。

0.1.2 超越对象的限制

Visual Basic 终于成了一种面向对象的语言,虽然没有人真正理解这意味着什么。在 Visual Basic 4 β 测试之前没有人用 Basic 写过面向对象的代码。文档编程者不理解它,开发者也不理解它,作者当然也不理解。面向对象的 Basic 现在还是一种冒险。

这本书中的面向对象代码远远超出了用户在 Visual Basic 手册中能看到的的所有东西的范围,但还是没有解决那些限制的办法。作者主要是基于自己对其他面向对象语言的知识,用试验的办法学会如何编制 Visual Basic 的类。在这样学了几乎一年之后,还是经常落入意想不到的陷阱。

正如研究 Visual Basic 1 版本的限制花了编程者们数年一样,还需要一些时间才能使人们开始欣赏面向对象的 Basic 所创造的奇迹。本书为面向对象的先锋们开了个头。

0.1.3 超越 Basic 的限制

如果用户不在乎使用 C++(或 C,或 Pascal,或者甚至 FORTRAN),可以将这两个世界的最好部分结合起来。用户可以用 Visual Basic 做它所擅长的事——快速创建用户界面,使用你所能得到的效率最高的开发和调试环境。用户可以用 C 来做程序中那些无论花多大代价也不能在性能或功能上作出牺牲的关键部分。用户将为调试用 C 写的部分而在编程时间和简易上付出代价,正如用户为用 Basic 写的部分在性能上付出代价一样。但用户将能够编制出功能强大而比那些只限于一种语言的开发者编制出的更快的程序。

遗憾的是,为了学会如何做到这一点用户需要另外一本书。作者计划在本书写完之后写那本书。用户不久将会在 Microsoft 的出版物中看到它。

0.2 本书讲述范围

本书是面向那些用 Visual Basic 4 专业版来开发 Microsoft Windows 3.1, Microsoft Windows NT 和 Microsoft Windows 95 应用程序的高水平程序员的。这是它的使命。实际上,本书对其他的人也是有用的。

Visual Basic 4 最重要的新特征是:它是一种不同的语言——面向应用软件的 Visual Basic (VBA)。VBA 是 1993 年在 Microsoft Excel 和 Microsoft Project 中引入的,1994 年又加到了 Microsoft Access 中,而且最终将会在其他许多 Microsoft 应用软件中出现。Visual Basic 现在只是面向应用软件的 Visual Basic 的客户之一(虽然是非常重要的一个),同样地,Microsoft Excel 和 Project 也是客户。这个语言引擎在技术上已经从集成环境中分离出来了。在对象浏览器中,用户可以看到 Visual Basic 和面向应用软件的 Visual Basic 有不同

的入口。

这个差别对用户意味着什么呢?它意味着,如果用户懂得 Visual Basic,也就懂得面向应用软件的 Visual Basic,而且在比较小的程度上反之亦然。Visual Basic 中的面向应用软件的 Visual Basic 的版本和将要出现在 Microsoft Access for Windows 95 中的是一样的。因此,用户可以将几乎是所有在这本书中出现的 VBA 技术(不是 Visual Basic 技术)运用到 Access 中去,不用作改动。但哪个技术是为哪个技术服务的?谁也说不清。用户必须进行试验。作者并没有在 Access 下测试过这些代码,但大多数应该可以工作。那些内部对象——应用程序、屏幕、打印机,和调试器——是 Visual Basic 特有的,作者不能保证它们在其他面向应用软件的 Visual Basic 环境中能正常工作。而那些 Visual Basic 提供的控件是通用的控件,在任何 VBA 应用软件中都应该能用,即使那个应用软件中没有这种控件。

在发行的时候,Microsoft Excel 和 Project 使用的 VBA 版本和 Visual Basic 和 Access 使用的 VBA 版本是不一样的。为老版本写的通用代码(不是面向电子表格和面向计划的)在 Visual Basic 下应该能够工作得很好,但 Visual Basic 代码可能不会在旧的 VBA 应用程序中正常工作。例如,Microsoft Excel 和 Project 不支持控制。也是在出版的时候,Microsoft Word for Windows 在应用程序中使用 Word Basic 而不是 Visual Basic。这种语言类似于 VBA,但与它不兼容。在这本书的存在期间,各版本的 Microsoft Word, Excel, Project, 还有可能其他 Microsoft 应用程序将会不再使用目前的(或晚一点)的 Visual Basic 版本。如果这样,这本书中的代码也能适用于这些环境。

虽然最初作者是为 Visual Basic 专业版用户编写本书的,但另外两种 Visual Basic 产品的用户也能利用这本书。首先,Visual Basic 标准版是专门为 Windows 95 操作系统设计的 32 位产品。作者的大多数代码是为两种环境而写的,使用条件编译(Visual Basic 的一个新特性)来使代码适用于 16 位和 32 位环境。如果用户使用标准版,则一切都白做了,因为它将用户限制在 32 位模式下。用户将获得更少的 Windows API 文档和很少的控件。作者没有在标准版下测试过,因此用户要自己来解决问题。第二种产品——Visual Basic 企业版是 Visual Basic 一个更昂贵的超集的客户/服务器。作者讨论的关于 Visual Basic 的一切也同样适用于这一新产品,但企业版有许多新增加的特点是作者没有讨论到的。

Visual Basic 4 相对于早期版本包含了一些重要的变化,这本书尽量地利用这些变化。但是,本书中的一些代码是在还没有 Visual Basic 4 β 版时用 Visual Basic 3 开发的(这儿描述的许多 Windows API 技术甚至适用于 Visual Basic 1)。

Visual Basic 4 模块的名称与旧版本中使用的不同,因此不能不加修改地使用与本书配套的 CD(另售)上的代码,但可以把它复制到老的模块里。为什么用户仍想继续使用 Visual Basic 3?因为它是一种最便宜的专以 Windows 3.1 为对象的 Basic。如果你的客户不需要 32 位程序的话,你可能还想保留它的。但是筛选并把本书的代码从 Visual Basic 4 转换到 Visual Basic 3 下将成为一个问题。

用户将需要借助目前的 Windows API 文档来充分利用这本书。那对于大多数 Visual Basic 用户来讲都是不成问题的,因为为 Visual Basic 专业版和企业版制作的 CD 盘包括了 Microsoft Developer Network 和 Visual Basic 初学者工具箱。MSDN/VB 初学者工具箱中有所有用户需要的 16 位和 32 位 Windows API 消息。它是对完整的 Microsoft Developer Network Development Library CD 盘的一个讽刺,它可能确实会说服用户买下全套产品。但

它也包含一些对核心 Basic 程序员很关键的信息,没有初学者工具箱或相当的 Windows API 文档,用户就不能充分利用这本书。MSDN/VB 初学者工具箱不附在 Visual Basic 标准版和专业版里。如果有这些版本或用于应用程序的 Visual Basic,用户需要找到单独的 Windows API 文档。

Microsoft 的 Windows API 文档,不论帮助文件或书中的都是针对 C 语言的。揭示如何将 C 的文档用于 Basic 编程是本书的一个主题,然而用户可能会要求额外的帮助。作者很喜爱 Daniel Appleman 所写的《Visual Basic 程序员 Windows API 指南》(中文版销售电话:(010)62562329,(010)62541992)一书,或许他(或别人)不久会将这本书写一个 Win32 版本。

作者写本书使用了多种不同的操作环境。最初阶段作者将 Microsoft Windows for Workgroups 作为主要环境,偶尔在 Windows NT 下启动(虽然一些低层操作在 Windows NT 的 16 位仿真层中工作的不是很好)。当 β 版的 Visual Basic4 问世时,作者开始在原来的机器上用 Windows NT,在另一台机器上用 Windows for Workgroups。当得到 Windows 95 β 测试版时,作者在第二台机器上改用 Windows 95 操作系统。作者偶尔也得启动 Windows for Workgroup 以确信所有新特性都适用于它。

作者通常花费更多时间在 Windows NT 下开发 32 位模式,之后在 Windows 95 下验证代码是可行的。使用多种环境是件困难的事情,如果作者的代码中有错误(看来是不太可能的),那也只能在这种情况下出现。

还有一点:这本书假定用户具有 Visual Basic 4 的类、集合、特性、可选参数、OLE 服务器等许多特性的知识。版本 4 是一个重大变革,将出现许多新特性(以一种人们从 1 版本开始就盼望已久的特性——续行符——为开端)。作者在假定用户已掌握基础的前提下来讨论这些特点。若用户在 Version 4 发布后不久便买了这本书,会觉得有些摸不着头脑。但用户可以通过阅读产品文档和在线帮助来加快学习进程。因为有些东西是新的并不意味着它是艰深的。

0.2.1 本书不涉及的内容

本书不涉及的内容是什么?确实不太好说。它包含高级的编程技术,但许多高级主题并未提及。它是关于专业编程的,但那是一个广阔的题目,我们不可能概括全部。它是讨论如何从 Visual Basic 4 的新特点中受益的,但许多代码在 Version 1 下也能正常工作。

下面是一些在本书中不会谈及的主题:

(1) 数据库编程。数据库编程是个热门话题,有些人肯定会为它写书的(有些人已写过一些数据库的书了)。这不是一本那样的书。在数据库上写一两章是不够的,另外,作者也不大精通于此。

(2) 控件。有许多使用复杂的控件,例如网格的技巧。作者提及了列表框和文本框,但仅作为一些能适用于更复杂控件的规律的示例。文档控件可以直接购得,不论怎么说,有太多的控件,全部提及是不可能的。

(3) MDI 表单和菜单。作者必须承认不喜欢 MDI 界面。作者喜欢程序是单一文档,有许多按钮,没有菜单。作者并非抱有成见,不过作者认为 MDI 和菜单并非是应包括在内的高级问题。如果想要了解,可以阅读产品文档。

(4) 构造帮助和安装工具箱。它们是有意义的问题,而且在某种意义上是高级问题——但它们不涉及编程问题,因而在书中并未提及。

(5) 尽量多的 API。作者渴望花几个月为 MAPI 或 WINSOCK 或多媒体编代码。网络和电话声音技术确实很有趣,作者也希望有时间研究增强图形功能的 API,例如 WinG, OpenGL, 或 WinToon,这些都是写书的有意义的题目——其他的书。

(6) 项目管理,界面设计,小组编程,测试,评价反馈,系统分析,摄取咖啡因,打篮球,还有许多其他促成软件开发成功发展的因素。这本书仅是关于代码的。作者的用户界面不必写了。作者的程序很小并易于管理。作者独自编写、设计这些程序,不需与别人合作。换句话说,作为一名软件开发,作者不同于你们(作者作为一名作家同其他人一样,必须在期限内完成创作)。不过,作者不是抽出时间探讨一些用户尽量回避而又不得不面对的编程问题。成为一名编写核心代码者并不代表软件开发的全部,但仍是有一定价值。

0.3 附带的 CD-ROM

本书提供了许多不同难度的例子。多数编程者会发现一些有用的东西来自于自己的程序,或许要作一些小的修改。代码可被引用只是一个方面。真正目的是给用户提供一个 Visual Basic 解决一切问题的信心。

书中的代码只是极小一部分。书中每一行程序在 CD 盘中都有 10 行以上相应的程序。不过 CD 上的一些例子代码确实不够有趣。用户已知道如何填写列表框,打开文件和对敲击按钮作出反应。但用户也会发现许多核心代码并不十分有趣,所以没写入书中。

另外,配套 CD 盘包含几种可从 Basic 中调用的非 Basic 工具:

(1) Windows API 类型库用更简明、更高效的内容代替了大多数 Windows 声明和常量声明语句。

(2) VBUTIL 动态链接库包含 Basic 遗忘的功能。

作者编写了这些工具,并对它们负责。书中大多数例子都需要这些工具支持。另外作者引用了其他作者的两个工具(经他们同意)。用户可以随便取舍。作者保证工具的正确性,但用户只能从它们的作者那里取得支持。这些工具用在书中,但并非必需。

(3) Message Blaster 是捕捉 Windows 消息的共享软件工具。许多 Basic 程序员对它的 16 位 VBX 版本十分熟悉。它的 32 位 OLE 控件版本还有一些新的编程技巧(作者在第五章讨论 Message Blaster)。作者所提供的内容将为你服务,但不支持你的客户。用户得从 WareWithAll 公司购买完整的版本。

(4) 回调服务器是另一种捕捉 Windows 消息和处理 Windows 回调函数的工具(也在第五章中讨论)。这是免费的,但同用户遇到的许多软件产品一样,是不作任何保证的。作者 Matt Curland 对用户的问题可能会感兴趣,但并不保证解决。

* 购买本书配套 CD-ROM,请写信至 Microsoft Press One Microsoft Way Redmond, WA 98052-6399 USA。

0.3.1 Windows API 类型库

作者希望能告诉用户,Windows API 类型库会省去所有写声明语句的需要。第二章将解释为什么这个有意义的目标可接近,但仍未完全达到。我们将更多地讨论类型库,尤其在

第二和第十章中。基本上,它们提供了一种比声明和常量定义语句更有效的方法。对象浏览器中列出了 Visual Basic 提供的将用户代码与 Basic DLLs 和控件相连接的标准类型库。Windows API 类型库用同样方法将用户代码与 Windows 系统动态链接库相连接。A 表简要地列出了库中的内容。

表 A Windows API 类型库小结

模块	概述
Kernel 和 KernelConst	Kernel 库中的函数和常量
User 和 UserConst	User(窗口管理)库中的函数和常量
GDI 与 GDIConst	GDI(图形设备界面)库中的函数与常量
WinBase 与 WinBaseConst	32 位内核库中的基本函数和常量
Multimedia	多媒体库中的函数和常量
WinMessage	消息常量和 SendMessage 函数别名
Shell	Shell 库中的常量和消息
CommonControl	一般控制函数,例如 ImageList
OleAuto	OLE 类型转换函数
Registry	登录函数和常量
Network	网络函数和常量
ErrorConst	Win32 API 错误常量
CommonConst	控制字符和其他有用常量

最后一个模块,即 CommonConst 需要作一些解释。使用 Visual Basic β 版过程中,当作者开始看到类型库时,便为那些 Basic 不让用户定义的常量定义了作者自己的字符常量。作者的类型库中最有价值的东西是 sCrLf——一个回车和换行组合体。用户将看到它在书中贯穿始终。Visual Basic 提供了 vbCrLf。为什么不用 vbCrLf 来代替作者自己定义的 sCrLf 呢?事实上,问题应该这样来提:为什么系统使用了作者的常量呢?

作者是如此欣赏自己的常量,以致作者觉得所有 Basic 编程者都应该喜欢它们,不管他们是否买了我写的书。因此,作者递交了一份错误报告(Microsoft 的建议也以和错误报告相同的方式递交)要求将类似的常量加入 Visual Basic。在报告末尾,作者提供了实现自己的常量的类型库源代码。用户可在 WIN.ODL 中看到类似代码。类型库常量实现并不困难,但在 β 测试期间由于作者递交建议太晚可能不大有希望被 Visual Basic 4 所采纳。但若用户在对象浏览器的 VBA 部分检查一下 Constants 模块的话,会发现它被采纳了。

加入一个新特征,即使是一个简单的,在像 Visual Basic 这样的大型软件产品中可不是一件简单的事情。在那些常量被加入并归入文档时,作者的书还差得远呢,而且将每个例子中的每一个 sCrLf 改为 vbCrLf 确实很伤脑筋。另外,作者还加入了一些 Visual Basic 没有采用的新常量。因此作者坚持使用自己的常量。用户必须特意地使用作者的常量来加载作者的类型库,而 Visual Basic 的类型库总会自动加载。B 表对此列出了 Visual Basic 支持的和作者的常量,还有一些容易混淆的类似常量。

表 B 一般常量

他们的	我的	概述
vbNullChar	sNullChr	空字符,Chr\$(0)
—	sEmpty	空字符串("")
vbEmpty	—	未初始化的可变类型变量

(续表 B)

他们的	我的	概述
Empty	___	未初始化的可变类型变量(用作空字符串时较慢)
vbNull	___	没有有效数据的可变类型变量
vbNullString	sNullStr	字符串类型的空指针(传给 API 函数)
___	pNull	Long 类型的空指针(传给 API 函数)
___	hNull	空(0)句柄
vbCrLf	sCrLf	回车换行组合符
___	sCrLfCrLf	双重的 sCrLf
vbCr	sCr	回车符
vbLf	sLf	换行符
___	sBell	响铃字符(ASCII 7)
vbTab	sTab	制表符(ASCII 9)
vbVerticalTab	sVerticalTab	垂直制表符(ASCII 11)
vbFormFeed	sFormFeed	换页字符(ASCII 12)
vbBack	sBack	回退符(ASCII 8)
___	cMaxPath	文件路径最大长度
___	cMaxName	文件名最大长度
___	sQuote1	单引号
___	sQuote2	双引号

0.3.2 VBUTIL 库

Basic 有一个灵活而强有力的库,但像多数语言的库一样,它有一些令人烦恼的限制。VBUTIL 库就弥补了一些使作者无法容忍的漏洞。这儿简要列出了在 16 位和 32 位模式下同样适用的通用函数:

GetFullPath	将相关路径转换为完整路径,并将完整路径分成各组成部分
SearchDirs	对目录列表(例如 PATH 环境变量)进行搜索
GetTempDir, GetTempFile	在临时目录建立临时文件
ExistFile	检验文件是否存在
Interrupt	调用系统中断(仅在 16 位模式下适用)
LoByte, HiByte, 与 friends	从字中摘用字节,从双字中摘用字
MakeWord, MakeDWord	将字节扩充成字,将字扩充成双字
LShiftWord 与 friends	向左或右移动位

库中也包括了一些 16 位的仿真,32 位 Kernel 库中一些有用的函数:

GetDiskFreeSpace	得到磁盘驱动器大小和剩余空间
GetDriveType	得到磁盘驱动器类型(软驱、硬盘、网络等等)
GetLogicalDrives	判断哪些驱动器可用
GetTempPath, GetTempDathName	在临时目录中建立临时文件(用 GetTempDir 和 GetTempFile 包装起来)
GetFullPathName	得到文件的完整路径(用 GetFullPath 包装起来)
SearchDath	在一目录路径下寻找文件(用 SearchDirs 包装起来)
FLushFileBuffers	刷新磁盘系统文件句柄的内容
GetLastError, SetLastError	得到或设置错误代码

用户将在第三章得到关于 VBUTIL 功能更详细的信息。

0.4 不必太紧张

这本书在 Visual Basic 的 β 测试周期经过许多不可预料的变化和反复,那时看起来完成几乎遥遥无期。硬盘全毁了、最后期限过了、新操作系统推出了、谎也撒了、文件舍弃了、兴趣丢失了、邮件烧掉了、工具失效了、网络出现故障了、囊中羞涩了、夜晚加班开夜车了、头发掉了或变灰白了、主管们离去了、不知怎地无错误的程序又出错了、原先的第十章也因此被删除了。

尽管如此,作者并没有抱怨。稍后用户将看到一些关于语言局限的埋怨和批评,但不要搞错:Visual Basic4 是一个令人惊奇的产品,在兼容性结构下有许多重大革新。它的缺点来自于想在一次升级中做得太多。Basic 一向很有趣,但这一版本提供许多新鲜玩意儿,摆弄这些玩意儿确实很有意思。没有其他语言能给用户提供在 Visual Basic 中那么快获得的喜悦。作者想用户会十分满意,并会很受启发的。至少作者是这样。