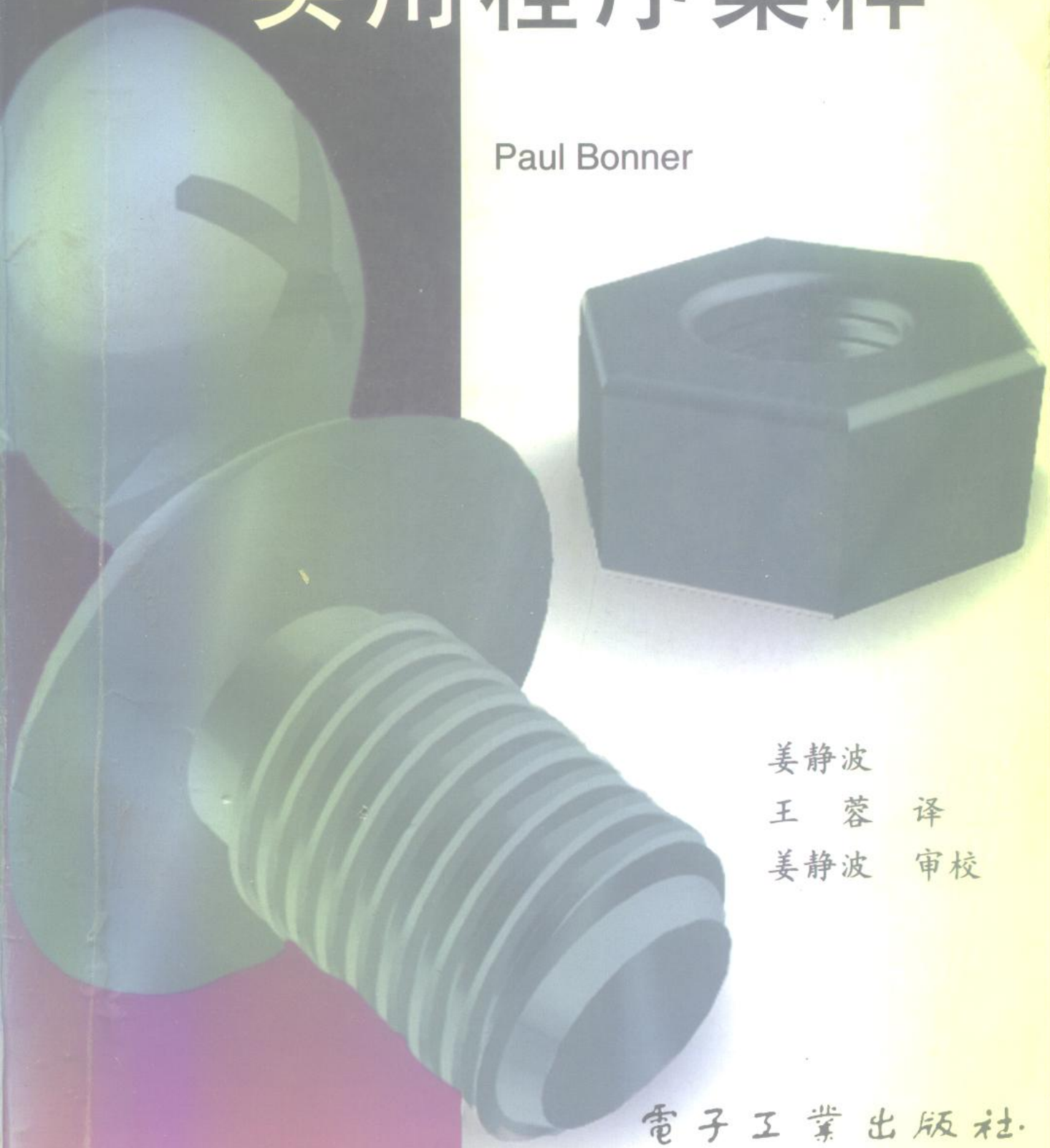
Visual Basic
实用程序集粹

Paul Bonner



姜静波
王　蓉　译
姜静波　审校

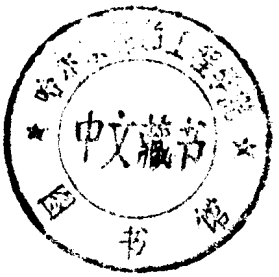
電子工業出版社

Visual Basic Utilities

Paul Bonner

Visual Basic 实用程序集粹

姜静波　王蓉　译
姜静波　校



電子工業出版社

（京）新登字055号

内 容 提 要

本书是美国著名Visual Basic专家的一部力作，它通过作者精选的八个一流Visual Basic 3.0实用程序详细介绍了Visual Basic的编程技术。阅读本书，您不但能够学会使用Visual Basic解决实际问题的全部过程，而且可以掌握充分发挥Visual Basic 能力的各种技术和技巧以及如何通过Windows API和DLL来扩充Visual Basic的能力。本书给出的例子非常实用，它们既可以不加任何改动地作为Windows实用程序使用，也可以裁剪之后用在读者的项目中。

本书可以使Visual Basic的初学者变成老手，使老手变成真正的Visual Basic大师。

Visual Basic 实用程序集粹

Paul Bonner 著
姜静波 王蓉 译
姜静波 校
责任编辑 李莹

*

电子工业出版社出版（北京市万寿路）
电子工业出版社发行 各地新华书店经销
北京天竺华颖印刷厂印刷

*

开本：787× 1092 毫米 1/16 印张：31 字数：750 千字
1994年 10 月第 1 版 1994年 10 月第 1 次印刷
印数：5000 册 定价：48.00 元
ISBN 7-5053-2592-2/TP・780

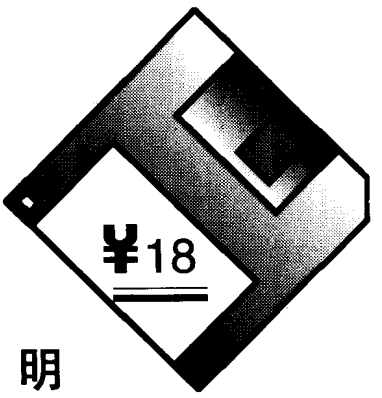
出 版 说 明

计算机科学技术日新月异，为了引进国外最新计算机技术，提高我国计算机应用与开发的水平，中国电子工业出版社与美国万国图文有限公司合资兴办的北京美迪亚电子信息有限公司取得了美国Ziff-Davis Press的独家版权代理。Ziff-Davis Press授权本公司通过电子工业出版社等出版机构全权负责在中国大陆出版该公司的中文版和英文版图书。现在与广大读者见面的是最近推出的第一批图书。今后我们还将陆续推出Ziff-Davis Press的最新计算机图书和软件，为广大读者提供更好的服务，传递更多的信息。

美国Ziff-Davis Press是全美最大的计算机出版商之一，它出版的书籍、杂志和光盘，主办的展览和会议，提供的咨询和网络服务，形成了整个行业潮流的主导。我们优选翻译出版的图书是Ziff-Davis Press的最新计算机图书，并采用了该公司提供的电子排版文件，从而提高了质量并大大缩短了图书的出版时间，从根本上改变了以往翻译版图书要落后原版书较长的“时差”现象，这在电子技术日新月异的时代具有深远的意义。

北京美迪亚电子信息有限公司

1994年9月



读 者 购 说 明

我公司为支持和方便读者阅读本书，决定以优惠价提供该书配套程序软盘，每张软盘18元（5英寸高密软盘，邮购加10%邮费）。

凡购买者请将现金寄回我公司，我公司在收款后尽快将软盘和发票一同寄出。为避免差错，请将收件人姓名、地址和邮编填写清楚。

供盘单位（通讯地址）：北京海淀区车道沟一号滨河大厦九层北京美迪亚电子信息有限公司

邮　　编：100081
联 系 人：李凤萍
联系电话：8425134、8414477-3699

献给我的母亲、纪念我的父亲，他们使我懂得了工作、独立和爱情的价值。

致　谢

如果没有无数同仁的帮助，本书就不会问世。首先，我要感谢很多通过计算机服务网络CompuServe共同切磋MSBASIC的同行，他们使我在多年从事Visual Basic教育工作时受益匪浅。其中，Jonathan Zuck、Jim Ferguson、Costas Kistos、Nelson Ford、Keith Funk、Keith Pleas和Jeff Simms特别值得一提，他们随时让我分享了他们的知识和宝贵经验。

感谢Neal B. Scott和Gary Wisniewski，本书中的部分实用程序取材于他们建立并且发表在公共场所的源代码。本书第3章使用了Neal提供的ZIP文件读取程序，没有它就不可能有VB ZIP这个外壳程序。第4章使用了Gary设计的File Manager控制，通过这一控制用户不但可以对File Manager实施“拖－放”操作，而且还可以对其进行配置。另外，Gary还应我的请求两次修改了这个程序。

感谢所有为本书的问世付出了辛勤劳动的人们。首先是Rob Hummel，是他第一个提出了出版这套“实用程序”丛书的设想并使之成为现实，Rob 清晰的思维和非凡的判断能力指导了本书的整个写作过程。然后需要真诚感谢的是Dan Appleman，作为本书的技术编辑，他的漂亮工作和广博的Visual Basic学识帮助我解决了数不胜数的问题。最后，还要感谢Kandy Arnold，感谢她高质量的编辑工作；感谢我的出版人Cindy Hudson，感谢她对本书的信任和忍耐；感谢KimHaglund以及Ziff-Davis Press公司中为本书的设计、印刷和销售付出劳动的每个人。

最后，仍然要感谢Betsy，她陪伴我又完成了一本书。

引　言

本书可以说是我与Visual Basic之间的爱情故事。同其它的爱情故事一样，它经历了许多的困苦和磨难，但也带给我了很多的喜悦。

Visual Basic在做象数据输入屏幕的构造、应用程序原型的建立以及简单数据库程序的编制这些事情时，非常容易。因而，它难免会习惯性地被人认为只是一个只能做做这种简单工作的工具。但对于那些富于进取精神的人们，对于那些勇于钻研、积极探索Visual Basic奥秘的人们，对于那些希望突破Visual Basic局限的人们，事实将会证明Visual Basic绝不仅仅只是一个只能用来构造这种简单应用程序的简单工具。在这些人的手中，Visual Basic将是一个可以用来构造系统级应用程序、使其它的应用程序自动运行或者对整个Windows系统的操作进行裁制的理想工具。事实上，除了偶尔需要借助于特制控制或动态连接库设施之外，作者认为几乎对于所有的Windows编程任务Visual Basic都是一个速度最快、使用最易和效率最高的工具。

本书精选了八个Visual Basic实用程序，每个程序都有两个用途。第一，每个程序本身都是一个很有用的实用程序。因此，即使大家宁愿不读本书的一个字，宁愿不看本书的一行程序，这八个实用程序仍然很有价值，它们可以大大丰富大家的Windows环境。

然而，我更希望的是本书读者能够体会出作者精选这些实用程序的另一个苦心所在：向大家介绍一些非常有用但至今尚未公开的Visual Basic编程技术，以及利用这些技术构造Windows应用程序的整个过程。这也是本书要完成的第二个任务。

本书的每一章都是按照这一思路组织的，都是引导读者先从描述实用程序目标的问题规划阶段开始，然后逐步过渡到实用程序用户界面和数据结构的设计阶段，最后以循序渐进的方式完成整个实用程序的编码。在每一章中，我们都要向大家交代清楚实用程序的实现技术困难以及相应的解决办法。

读者对象

本书给出的所有实用程序都经过了精心的设计，目的是帮助读者突破 Visual Basic本身的局限。

虽然本书也可以作为标准Visual Basic编程入门教材的辅助学习材料，但本书的读者并不是Visual Basic的初学者，而是那些已经熟悉了Visual Basic的基本功能、也许正苦于不知如何突破Visual Basic本身局限的读者。本书就是要向这些读者介绍扩充Visual Basic功能和克服Visual Basic局限的方法，目的是使这些读者能够充分地开发出Visual Basic这个优秀编程工具的全部潜能。

为了实现这一目标，书中很多的实用程序大量地使用了Windows应用程序编程接口(API)函数。Windows API是Visual Basic程序员不可多得的珍贵宝藏，但它也可能会使用户望

而却步。同Visual Basic提供的功能相比，API 函数不但使用起来很不直观，而且它对用户的水平也要求得太高，这是我只用了几次API 函数之后就得出的一个结论。在每次使用API函数时，我的手心里都捏着一把汗，都得随时准备去应付各种可能出现的问题。为此，本书要达到的目标之一就是要向读者交代清楚当设计一个Visual Basic应用程序时，应该如何确定实现所需要的API 函数，确定这些函数之后又该怎样把它们集成到应用程序中以及怎样对所实现的应用程序进行测试以验证是否完成了我们的设计意图。

由于有些Windows功能单靠Visual Basic无法实现，所以书中的有些实用程序也使用了特制控制或动态连接库(DLL)来访问相应的Windows 函数。对于 Visual Basic程序员来说，特制控制和DLL是两个非常有用的设施。尽管如此，本书还是尽量地避免这些设施的使用。在多数的情况下，书中使用这些设施的目的都是证明只用Visual Basic和API调用也可以达到完全相同的效果。

实用程序简介

本书给出的实用程序在设计上颇具匠心，这些程序可以满足各种需要。

第1章给出的实用程序是一个VB源代码库管理程序(VB Code Librarian)，它是一个为简化Visual Basic应用程序增加标准例程之过程而设计的程序。这个程序提供了一个可以存储Visual Basic应用程序开发常用标准例程的源代码库，例如屏幕格式居中显示例程、.INI文件访问例程和标准消息框创建例程等。通过这个实用程序的介绍，我们可以学会顺序文件高速访问例程的使用，掌握利用列表框的Sorted属性对其进行索引的技术和Windows剪贴板的用法。

第2章给出的实用程序是一个时钟标题显示程序(Title Bar Clock)，它是一个系统级的实用程序，可以在活跃Windows 应用程序的标题棒上定期地显示出系统的当前时间和日期。为了实现这一功能，这个程序大量地使用了Windows API 函数。这一章以很大的篇幅介绍了有关Windows API函数的使用方法，成功地避免了当用户应用程序需要处理其它应用程序时可能引起的干扰错误。

第3章给出的实用程序是一个PKWARE公司在公共领域发行的文件压缩实用程序的Windows接口程序(VB ZIP Shell)，它为PKZIP和PKUNZIP这两个DOS实用程序提供了Windows接口。通过这个实用程序的介绍，我们可以学到很多非常有用的技术，其中包括如何在Windows环境下读取ZIP文件的内容，如何在隐含窗口中运行DOS 应用程序，以及如何构造具有多文档界面的应用程序。

第4章给出的实用程序是一个Windows打印队列管理程序(VB Q)，它允许用户以立即打印或延迟打印(在指定时间之后打印)两种方式打印Windows的文档文件。这个程序说明了Windows注册数据库(Windows Registration Database)和 SHELL.DLL的用法，介绍了可以指导用户逐步完成困难工作的Wizard 式指南对话框的构造方法，提供了各种同其它应用程序接口的技术。为了实现从File Manager接受“拖－放”消息，这个Visual Basic应用程序还使用了特制的控制。

第5章给出的实用程序是一个Windows字体演示程序(VB Typographer)，它允许用

户以任意的字号或字形显示系统安装的任何字体的所有字符，并且可以打印出指定字体或者系统所有字体的样张。这个程序介绍了很多的技术，利用这些技术可以解决多字体处理、正文折行以及打印机接口问题。

第6章给出的实用程序是一个多剪贴板支持程序(Clips)，它可以存储用户拷贝到Windows剪贴板中的所有正文，用户可以使用多个剪贴板对所截获的内容进行编辑和组合。为了实现对剪贴板操作的监视，这个程序使用了特制DLL。

第7章给出的实用程序是一个桌面菜单程序 (DeskMenu)。当用户在Windows的桌面上按动一下鼠标器的右按钮时，这个程序会弹出一个以Program Manager组中各个程序组及其各个程序项为菜单项的多级层次菜单。这个程序说明了Program Manager动态数据交换接口的用法，介绍了弹出式菜单的动态跟踪方法和菜单控制的动态加载技术。为了实现对桌面窗口上鼠标器事件的监视，这个程序使用了特制DLL。

第8章也就是本书的最后一章给出的实用程序是一个VB程序构造辅助程序(VB MAKer)，它利用Visual Basic中的Access数据库支持部分实现了对VB 特制控制数据库的维护，这个数据库中的内容可以辅助用户生成新的Visual Basic应用程序。这个程序介绍了各种与Visual Basic中数据控制设施有关的技术，提供了对具有数据敏感特性的列表框控制的模拟方法，举例说明了DOS路径的搜索方法。

借助于适当的工具和适当的方法，Visual Basic几乎可以做任何事情。在随后的章节中，我们谨向大家介绍完成这八个实用程序所必需的一些工具和方法以及我们需要这些工具和方法的原因。每个实用程序都由“我想知道这样是否可能”这个问题引出，我希望这种风格会增加大家的阅读兴趣。

目　　录

引言 I
读者对象 I
实用程序简介 II
第1章　VB源代码库管理程序 ..... (1)
VB Code Librarian的使用 ..... (2)
库例程的访问 ..... (2)
库例程的创建 ..... (3)
热键访问 ..... (6)
操作自动化问题 ..... (7)
VB Code Librarian的内部机制 ..... (7)
屏幕格式设计 ..... (7)
事件驱动的代码 ..... (9)
初始化 ..... (9)
GetFiles例程 ..... (10)
选择库例程 ..... (11)
修改库例程 ..... (14)
CopyOldRec例程 ..... (15)
删除库例程 ..... (16)
存储新的库例程 ..... (18)
SaveNewRoutine函数 ..... (18)
拷贝库例程 ..... (19)
小结 ..... (19)
第2章　时钟标题显示程序 ..... (42)
Title Bar Clock的使用 ..... (44)
Title Bar Clock的内部机制 ..... (45)
三个基本的Windows API调用 ..... (46)
时钟显示的美学问题 ..... (48)
父窗口与子窗口 ..... (48)
异常情况的处理 ..... (50)
异常处理的办法 ..... (52)
Visual Basic的特殊处理 ..... (52)
Title Bar Clock的集成 ..... (54)
屏幕格式的设计 ..... (55)

第3章 DOS ZIP程序的Windows接口 ..... (64)
VB ZIP Shell的使用 ..... (65)
其它的用户界面要素 ..... (68)
VB ZIP Shell的内部机制 ..... (70)
设计目标 ..... (70)
指定命令行参数 ..... (71)
启动PKZIP和PKUNZIP ..... (74)
功能越多，工作越多 ..... (75)
读取ZIP文件 ..... (80)
屏幕格式的设计问题 ..... (82)
屏幕格式的运行时装配 ..... (85)
命令行参数 ..... (90)
VB ZIP Shell的改进 ..... (91)
小结 ..... (92)
第4章 打印队列管理程序 ..... (147)
介绍一些打印方法 ..... (147)
VB Q的使用 ..... (149)
把文件送入打印队列 ..... (150)
建立打印文件与应用程序之间的对应关系 ..... (153)
VB Q的报告功能 ..... (153)
VB Q的内部机制 ..... (156)
CenterDialog例程 ..... (159)
列出菜单的ID标识 ..... (160)
借用File Manager的对话框 ..... (162)
VB Q Keys ..... (165)
KeysPrint例程 ..... (168)
基本的技术 ..... (173)
访问Registration Database ..... (173)
使用File Manager提供的“拖－放”支持 ..... (175)
输入并且监视延迟打印时间 ..... (177)
屏幕格式的设计 ..... (179)
VB Q的改进 ..... (180)
第5章 字体演示程序 ..... (231)
VB Typographer的使用 ..... (231)
VB Typographer的内部机制 ..... (234)
准备工作 ..... (236)
选择字体 ..... (240)
设置字体的其它属性 ..... (241)

改变VB Typographer主屏幕格式的大小 ..... (242)
VB Typographer 的打印例程 ..... (244)
单词的折行处理 ..... (245)
打印输出垂直位置的监控 ..... (248)
打印底线的校准 ..... (249)
打印行间距的控制 ..... (249)
其它的实用例程 ..... (250)
第6章 多剪贴板支持程序 ..... (286)
Clips的使用 ..... (287)
处理捕获到的剪贴板正文 ..... (288)
Clips的内部机制 ..... (292)
监视Windows的剪贴板 ..... (293)
命运伸出了幸运之手 ..... (294)
捕获Windows剪贴板的内容 ..... (300)
Clips的文件系统 ..... (302)
公用对话框 ..... (303)
监视Clips内容的修改情况 ..... (306)
CheckChanges()函数 ..... (307)
有选择地组合Clips项 ..... (308)
Combine的工作方式 ..... (309)
组合所有的Clips项 ..... (309)
跟踪Clips的状态 ..... (311)
ToggleMenu例程 ..... (311)
ToggleSettings例程 ..... (312)
LabelIt例程 ..... (312)
限制与错误捕捉 ..... (313)
第7章 桌面菜单程序 ..... (343)
DeskMenu的使用 ..... (343)
DeskMenu的由来 ..... (347)
DeskMenu的内部机制 ..... (348)
建立Windows桌面窗口对象的子类对象 ..... (349)
调用DMENU.DLL ..... (351)
处理DMENU.DLL的消息 ..... (351)
Sub Main()过程 ..... (353)
隐藏程序的图标 ..... (354)
退出Windows ..... (358)
加载Program Manager的菜单 ..... (359)
获取程序组数据 ..... (360)

获取程序项数据 ..... (363)
GetItems例程 ..... (364)
重新加载DeskMenu的菜单 ..... (367)
跟踪弹出式菜单 ..... (369)
启动应用程序 ..... (370)
RunProg例程 ..... (371)
使用专门的.INI文件 ..... (373)
产生加载过程的动画效果 ..... (374)
第8章 VB程序构造辅助程序 ..... (405)
VB MAKer的使用 ..... (406)
重复的.VBX文件 ..... (407)
描述特制控制 ..... (408)
处理.MAK文件 ..... (409)
启动.MAK文件 ..... (411)
VB MAKer的内部机制 ..... (411)
创建半动态连接列表框 ..... (412)
选择列表项 ..... (415)
在列表框之间传送数据项 ..... (416)
增加控制到数据库中 ..... (417)
增加新记录到.VBX数据库中 ..... (423)
删除数据库的记录 ..... (424)
根据描述信息查找控制 ..... (425)
打开现有的项目 ..... (426)
创建新的.MAK文件 ..... (429)
存储项目文件 ..... (430)
启动Visual Basic ..... (431)
清除VB MAKer的数据库 ..... (434)
附录A 工具和参考书 ..... (475)
附录B 配套软盘的安装指南 ..... (479)

第1章 VB源代码库管理程序

VB源代码库管理程序（VB Code Librarian）可以帮助用户维护Visual Basic 可重用库，库中的子程序和函数可以在Visual Basic的项目开发中得到重复使用。本章介绍的主要技巧包括：

- 编写高速顺序文件的访问例程；
- 利用列表框的Sorted属性对其进行索引；
- 为应用程序提供热键访问；
- 使用Windows的剪贴板。

我们编写的每个子程序或者函数都不同程度地具有一定的可重用性，它们都有可能成为一个允许用户反复重用的软件构件，它们可以被需要用到这些功能的其它例程直接调用。

因此，我们编写的每一行代码都可能节省我们将来的开发时间、编程时间和排错时间。

当然，实际上并不是我们编写的每个子程序都有可能得到很好的重用，这还要取决于它们的编码风格及其所涉及的应用领域。如果这两个方面没有太大的问题，那么多数子程序都可以得到重用。例如，如果我们编写了一个屏幕格式居中显示例程、一个读取私有.INI文件信息的例程或者一个从全路径名中抽取文件名的例程，那么这些例程在大项目的开发过程中都可能得到反复地使用。

当我们结束老项目的开发，开始新项目之时，正是老项目中许多例程在新项目中得到重用的绝好之机。至少在理论上，我们没有理由不这样做。实际上，这样做非常有意义。尽管如此，遗憾的是Visual Basic并没有提供一个支持这种重用的好办法。事实上，通过Visual Basic实现在新项目中重用老项目中的例程简直困难极了。

为了实现这一需求，我们可以有几种办法。例如，我们可以把老项目中包含所需可重用例程的屏幕格式或者模块整个地加入到新项目中。虽然我们可以从老项目中找出那些需要重用的文件，并且通过AddFile命令把它们加入到新的项目中，但整个工作过程实在是太不方便，而且充满了危险。一方面，如果我们要把这些文件变成新项目的一个组成部分，就可能有无意中改变这些文件的危险，因而可能对老项目中的原有代码产生破坏。另一方面，由于Visual Basic在创建.EXE文件时要对项目文件中的所有代码进行编译，而不是只编译那些实际用到的例程，因此如果我们只需要用到老项目中这些模块或者屏幕格式中很少的几个例程，那么最后生成的.EXE文件大小就会严重失调。

另一种办法是一旦我们把需要重用的例程拷贝到另一个模块之后，就切断这些例程同老项目中原有文件之间的一些联系。这样虽然解决了上面的问题，但它同前一种方法一样地愚蠢和容易出错。首先，由于Visual Basic在模块一级不允许两个例程具有相同的名字，这实际上意味着我们无法在模块之间拷贝例程。因此，我们只能先从老项目的原来文件中把需要重用的例程裁剪下来，然后再把它粘贴在一个新文件中。这样，当Visual Basic提示我们

是否需要在关闭这个新文件之前把修改过的老项目文件存储起来时，如果我们一时糊涂错误地选择了存储操作的话，就同样地会破坏老项目中的原有文件。另外，由于Visual Basic的全局函数声明或者常量并不与实际使用它们的例程存储在一处，而是全部存储在相应例程所在模块的Declarations部分，所以在使用这种方法时，用户有时可能会忽略这一事实，忘记把重用这些例程所必需的全局函数声明或常量也拷贝到新的模块中。总而言之，这种方法需要：

(1)把含有重用例程的那个老项目文件加入到新项目中；

(2)从这个老项目文件中裁剪出那些需要重用的例程；

(3)把裁剪出的重用例程粘贴到新项目文件的某个模块中；

(4)从新项目中除去这个老项目文件。

如果我们希望重用老项目文件中的多个例程，那么就需要重复上述的第(2)步和第(3)步操作。如果我们需要补上开始时漏掉了的全局声明，那么必须从第(1)步开始再来一遍，把这些例程需要用到的全局声明检索出来。

实现不同Visual Basic项目之间的代码重用当然并不只上述两种办法。例如，我们可以打开一个由正文编辑程序存储起来的ASCII模块文件或者屏幕格式文件，如由Notepad的ASCII Save存储的文件，然后把其中所需要的例程拷贝到指定的文件中；我们还可以利用模块文件或者屏幕格式文件的硬拷贝输出，把需要重用的例程手工输入到新的项目中。但我认为这些方法都不能特别地令人满意，因为无论哪种方法都是间接使用的，而且比较容易出错。因此，没有哪个用户会愿意使用这样的方法。

正是因为上述各种方法都不能令人满意地解决不同Visual Basic项目之间的代码重用问题，我意识到必须找出一个更好的办法。我认为解决这个问题的较好办法是为Visual Basic建立一个例程库，然后让用户通过这个库来存储和访问今后常会用到的例程。本章给出的VB源代码库管理程序（VB Code Librarian）就是这样的一个实用程序。

VB Code Librarian的使用

VB Code Librarian可以实现在不同Visual Basic项目之间传送函数和例程的过程自动化，并且可以大大地简化这一过程。VB Code Librarian 支持可重用例程库的建立，这使我们在开发Visual Basic项目时可以很容易地重用库中的例程。当用户使用库中的例程时，VB Code Librarian 并不会因此而破坏老项目中的原有例程。在用户寻找可重用例程时，VB Code Librarian 还可以使用户避免一个文件一个文件地进行搜索。

库例程的访问

就VB Code Librarian而言，所谓代码库就是一个可以在任何Visual Basic 开发项目中得到重用的例程或函数的集合。该代码库的使用涉及到两个步骤：第一步是库例程的建立，即把老项目的可重用例程或函数加入到该代码库中；第二步是库例程的访问，即把该代码库中的可重用例程或函数送到新的开发项目中。经过精心的设计，VB Code Librarian在这两个方面都可以对用户起到辅助作用。

显然，如果例程没有入库就无法访问，所以在访问库例程之前必须先在库中加入一些

例程。由于建立代码库的基本思路是库例程只需输入一次但可以使用多次，所以在VB Code Librarian中我们侧重的是如何使库例程的访问更方便。为了达到这一目的，我们把VB Code Librarian启动之后的缺省动作设置为库例程的访问。因此，当VB Code Librarian启动之后，屏幕上就显示出一个含有库中所有例程之标题的列表框，如图1.1所示。

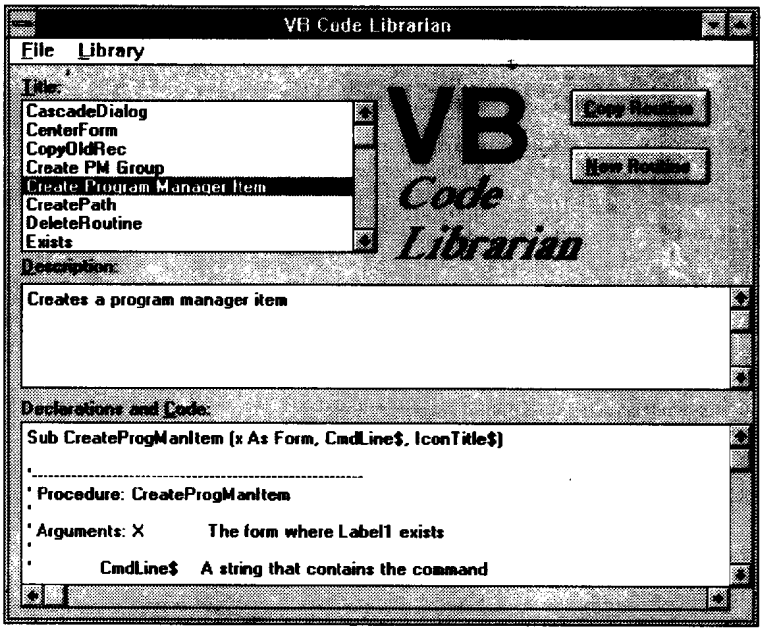


图1.1 VB Code Librarian的主屏幕

在图1.1中，我们看到在Title列表框的下面有两个编辑框。其中上面的编辑框被VB Code Librarian用来显示当前被选例程的描述信息，下面的编辑框用来显示该例程的源代码。这两个框中的内容都可以编辑，所以用户随时都可以改变该例程的描述信息或者修改其源代码。

当开发一个Visual Basic项目时，如果需要用到某个库例程，那么用户只需在Title列表框中选择相应的标题，然后按动主屏幕上的Copy Routine 按钮或者选择其Library子菜单中的Copy Routine，VB Code Librarian就会显示出一个如图1. 2所示的消息框，告诉我们所选例程的代码已经被拷贝到Windows的剪贴板中。

当所选例程的代码被拷贝到Windows的剪贴板之后，我们只要将光标移到当前代码模块或者屏幕格式的Declarations部分，然后按动组合键Shift+Insert，就可以把它粘贴到当前的Visual Basic项目中。

以上，就是我们重用一个库例程所需要进行的全部工作。

库例程的创建

库例程的创建同库例程的访问一样简单，它也要用到Windows的剪贴板。

为了创建一个新的库例程，我们需要调出VB Code Librarian的New Routine屏幕。为了调出这个屏幕格式，我们可以选择Library菜单中的New Routine或者按动主屏幕上的New