

杜国梁 杜大鹏 等编著

Visual Basic 3.0 ~ 4.0 for Windows 程序设计与范例



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

Visual Basic 3.0 ~ 4.0 for Windows 程序设计与范例

杜国梁 杜大鹏 等编著

电子工业出版社

内 容 提 要

本书共分九章(0~8章),前三章介绍了 Visual Basic 软件包基本使用方法,第3章给出了程序界面设计的一些示例,第4章重点介绍了 Visual Basic 程序设计语言的基本知识和基本的程序设计方法,涉及了大部分 Visual Basic 语言的内容。从第5章开始,介绍了使用 Visual Basic 开发 Windows 应用程序的一些较为深入的方面,包括图形编程、Windows API 函数的调用,以及编制符合 Windows 规范的帮助系统的知识。第8章特别介绍了 Visual Basic 的最新版本 4.0 版的使用基础知识,其中包括新的集成开发环境以及编程语言中与 3.0 版的不同之处,以期对有机会接触新版本的读者有所帮助。各章都包括了示例程序,所有程序的注释以及提示信息等均以汉语表达。

本书可作为 Visual Basic 3.0 与 4.0 的入门读物,对于需要较为深入内容的读者也有一定的帮助。

JS443/20
10

Visual Basic 3.0 ~ 4.0 for Windows

程序设计与范例

杜国梁 杜大鹏 等编著

责任编辑 杨丽娟

电子工业出版社出版

电子工业出版社发行 各地新华书店经销

北京科技大学印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:18.25 字数:455 千字

1996 年 10 月第一版 1996 年 10 月第一次印刷

印数:5000 册 定价:22.00 元

ISBN 7-5053-3736-X/TP·1588

前 言

自从 Windows 问世以来,其图形用户界面日益深入人心,在硬件允许的情况下,几乎所有的主流机都装上了 Windows 系统。在 Windows 下开发应用程序形成了一股潮流。由于开发 Windows 应用程序的难度较大再加上开发工具的限制,一开始,开发 Windows 应用程序只是专业程序员的工作。但在 Visual Basic 推出以后,这种情况已大大改观。越来越多的用户认识到 Visual Basic 的优越性,特别是非专业人员尝到了用 Visual Basic 开发 Windows 应用程序的甜头,Visual Basic 的用户越来越多。在中国,由于 Visual Basic 接受汉字,用它开发 Windows 下的中文应用程序也非常合适。最近一年多,Visual Basic 在中国的用户急剧增加。由于软件的流行,与软件使用有关的书籍也就大行其道。很多出版社都出了 Visual Basic 方面的书籍,其中以翻译的居多。与众多的与 Visual Basic 有关的书籍相比,本书有如下特色:

- 适合中国用户使用。由于大多数翻译书未顾及读者对象,多数只翻译了正文而忽略了软件界面与程序代码中的注释,这给英文基础较差的读者造成了阅读上的困难。在本书中,只有关键字才使用英文原样,其余都使用汉字表述,既传达出参数的意义,又与关键字具有明显的区别。所有示例程序中的注释一律使用汉语,一切信息、提示也都用汉字表达,这大大地方便了英语有困难的读者。

- 本书是作者使用 Visual Basic 开发中文应用程序的经验总结,也是学习和体会的结晶。对于程序中何处可以使用汉字,何处不可使用汉字都有明确的说明。有助于读者利用 Visual Basic 开发中文应用程序。作者也使用了一部分 Visual Basic 在线帮助中的示例,但都经过精心地改造,使之适合中国读者的口味。

- 叙述尽量做到深入浅出,特别适合略有 Basic 语言编程知识的读者阅读。同时为照顾高级用户的需要,也加进了一些较为深入的内容。

- 本书中的全部示例程序都经过实际上机验证通过,并以粘贴的方法写入书中,这就减少了程序在录入和排版中可能引入的错误。

本书是多人合作的结果。具体分工情况如下:前四章(0~3章)由杜国梁编写;杜大鹏,第8章并为本书编制了本书所有的示例程序;辛再甫,第5章和第6章;李善茂,第7章并为书稿录入和打印做了不少工作。全书由杜国梁统稿。另外,梁国珍、李秋阳、卢晓宇和杜文华等对本书给予了大力支持,在此一并致谢。

作 者

一九九六年三月

目 录

绪 论	(1)
0.1 Visual Basic 简介	(1)
0.2 Visual Basic for Windows 所要求的软硬件条件	(2)
0.3 Visual Basic 3.0 专业版的组成	(2)
0.4 Visual Basic 应用程序组成	(3)
0.5 Visual Basic 应用程序的开发步骤	(4)
第 1 章 VISUAL BASIC 应用程序的集成开发环境	(5)
1.1 主窗口	(5)
1.1.1 标题栏	(5)
1.1.2 菜单条	(6)
1.1.3 工具条	(6)
1.1.4 其它成分	(7)
1.2 工具箱窗口(ToolBox)	(7)
1.3 窗体窗口(Form)	(8)
1.4 项目窗口(Project)	(8)
1.5 属性窗口(Properties)	(9)
1.6 File(文件)菜单	(10)
1.7 Option(选项)菜单	(15)
1.8 使用帮助信息	(17)
1.8.1 利用 Help(帮助)菜单	(18)
1.8.2 上下文相关的帮助	(18)
1.8.3 从帮助信息中复制代码	(18)
1.9 一个简单的 Visual Basic 程序	(18)
第 2 章 程序代码的编辑操作	(22)
2.1 使用代码窗口	(22)
2.2 使用 Edit(编辑)菜单中的命令	(23)
2.2.1 Edit(编辑)菜单中的命令用法	(23)
2.2.2 式样匹配和常规表达式	(26)
2.3 使用 View(查看)菜单中的命令	(26)
第 3 章 VISUAL BASIC 应用程序界面设计实例	(28)
3.1 软件封面	(28)
3.2 设计工具条按钮和状态栏	(32)
3.2.1 设计要求	(32)

3.2.2 窗体及控件的属性	(33)
3.2.3 程序代码	(35)
3.2.3.1 鼠标事件介绍	(36)
3.2.3.2 程序代码的处理说明	(38)
3.3 选项卡的设计	(39)
3.3.1 窗体的设计	(40)
3.3.2 窗体与控件的属性	(40)
3.3.3 程序代码	(44)
3.4 使用鼠标拖放操作的程序界面	(45)
3.4.1 拖放操作	(45)
3.4.2 DragDrop 与 DragOver 事件	(46)
3.4.3 Drag 方法	(46)
3.4.4 拖放功能的设计步骤	(47)
3.4.5 示例:以拖放方式显示图形文件的图形	(47)
3.4.5.1 窗体的设计	(47)
3.4.5.2 窗体和控件的属性设置	(47)
3.4.5.3 程序代码	(49)
3.5 在文本框中输入指定的字符	(51)
3.5.1 常用的键盘事件	(51)
3.5.1.1 KeyDown, KeyUp 事件	(51)
3.5.1.2 使用 KeyDown 和 KeyUp 事件过程的例子	(53)
3.5.1.3 KeyPress 事件	(54)
3.5.2 实例:控制文本框中的输入内容	(54)
3.5.2.1 窗体设计	(55)
3.5.2.2 窗体及控件的属性	(55)
3.5.2.3 程序代码	(57)
3.6 在窗体间移动控件	(59)
3.6.1 窗体及控件的设计与属性	(59)
3.6.2 所需代码	(60)
3.6.2.1 代码模块	(60)
3.6.2.2 Form1 窗体模块	(60)
3.6.2.3 Form2 窗体模块	(62)
3.6.2.4 运行中的窗体	(63)
3.6.3 滚动比窗体大的图形	(63)
3.6.3.1 窗体与控件的设计	(64)
3.6.3.2 程序代码说明	(64)
3.6.3.3 程序的运行	(66)
第4章 VISUAL BASIC 语言概要	(68)

4.1 Visual Basic 源程序的书写规则	(68)
4.2 Visual Basic 中的数据类型	(69)
4.2.1 整型数据(Integer)	(70)
4.2.2 长整型数据(Long)	(70)
4.2.3 单精度类型(Single)	(71)
4.2.4 双精度类型(Double)	(71)
4.2.5 货币类型(Currency)	(72)
4.2.6 字符串类型(String)	(72)
4.2.7 变体数据类型(Variant)	(72)
4.3 变量、常量(数)、运算符与表达式	(74)
4.3.1 变量与常量	(74)
4.3.1.1 创建变量与常量	(74)
4.3.1.2 命名规则	(75)
4.3.1.3 变量使用中的注意事项	(76)
4.3.1.4 用户自定义的数据类型	(76)
4.3.1.5 数组	(77)
4.3.2 Visual Basic 中的运算符	(79)
4.3.2.1 算术运算符	(80)
4.3.2.2 表达式	(82)
4.3.2.3 关系运算符	(83)
4.3.2.4 逻辑运算符	(83)
4.3.2.5 字符串拼接运算符(&)	(85)
4.3.2.6 运算符的优先级	(86)
4.4 语句与过程	(86)
4.4.1 赋值语句	(87)
4.4.2 Visual Basic 与用户直接交互的语句和函数	(88)
4.4.2.1 InputBox, InputBox \$ 函数	(88)
4.4.2.2 MsgBox 函数和 MsgBox 语句	(88)
4.4.3 条件语句	(91)
4.4.3.1 If 语句	(91)
4.4.3.2 Select Case 语句	(93)
4.4.4 循环语句	(96)
4.4.4.1 For...Next 循环语句	(96)
4.4.4.2 Do...Loop 语句	(98)
4.4.4.3 While...Wend 语句	(99)
4.4.5 结束程序语句	(101)
4.4.6 Visual Basic 中的内部函数简介	(101)
4.4.6.1 数学函数	(101)
4.4.6.2 数据类型转换函数	(103)

4.4.6.3 随机数函数	(104)
4.4.6.4 字符串函数	(105)
4.4.6.5 日期时间函数	(107)
4.4.6.6 数据格式化函数—Format[\$]	(108)
4.4.7 事件过程	(116)
4.4.8 自定义的过程	(116)
4.4.8.1 创建自定义的过程	(116)
4.4.8.2 自定义过程的一般形式	(117)
4.4.8.3 过程的调用	(118)
4.4.8.4 过程调用的深入讨论	(123)
4.5 文件的操作与读写	(128)
4.5.1 直接操作 DOS 文件系统的语句和函数	(128)
4.5.1.1 直接操作 DOS 文件系统的语句	(128)
4.5.1.2 了解文件系统信息的函数	(129)
4.5.1.3 代表文件属性的符号常数	(130)
4.5.1.4 使用操作 DOS 文件系统的语句与函数的示例	(131)
4.5.2 文件的读写	(133)
4.5.2.1 文件的打开与关闭	(133)
4.5.2.2 顺序文件的读写	(135)
4.5.2.3 随机文件与二进制文件的读写	(140)
4.6 一个实用的应用程序	(142)
4.6.1 几个用到的函数与语句	(142)
4.6.1.1 Shell 函数	(142)
4.6.1.2 DoEvents() 函数语句	(143)
4.6.1.3 Sendkeys 语句	(144)
4.6.2 窗体的设计	(146)
4.6.3 程序代码	(148)
4.6.3.1 辅助文件	(153)
第 5 章 图形编程	(157)
5.1 屏幕坐标	(157)
5.1.1 屏幕与窗体的缺省坐标系	(157)
5.1.2 窗体(图片框)坐标系的定制	(157)
5.1.3 定制坐标系的另一种方法	(159)
5.2 绘图时使用颜色	(159)
5.2.1.1 使用 RGB 函数	(159)
5.2.1.2 使用 QBColor 函数	(159)
5.2.1.3 使用十六进制颜色代码	(160)
5.2.1.4 使用 Constant.txt 文件中定义的颜色	(161)

5.3 图形方法	(162)
5.3.1 Line 方法	(162)
5.3.2 Circle 方法	(164)
5.3.3 Pset 方法	(165)
5.3.4 Point 方法	(165)
5.4 与图形方法有关的属性	(165)
5.4.1 CurrentX、CurrentY 属性	(166)
5.4.2 DrawWidth、DrawStyle 属性	(166)
5.4.3 FillColor 和 FillStyle 属性	(167)
5.4.4 DrawMode 属性	(168)
5.4.5 AutoRedraw 属性	(174)
5.5 打印机对象	(175)
第 6 章 Windows API 函数的调用	(177)
6.1 前言	(177)
6.2 Visual Basic 与 Windows	(177)
6.3 什么是 Windows API 函数	(177)
6.3.1 Windows 内核库(KERNEL.EXE)	(178)
6.3.2 Windows 的用户界面管理库(USER.EXE)	(178)
6.3.3 Windows 的图形设备界面库(GDI.EXE)	(178)
6.4 在 Visual Basic 中访问 Windows API 函数	(179)
6.4.1 找到有用的 Windows API 函数	(179)
6.4.2 学会如何使用函数调用	(179)
6.4.3 声明外部例程	(179)
6.4.4 在程序中使用 API 例程	(181)
6.4.5 在调试之前保存项目文件	(182)
6.4.6 调试项目	(182)
6.5 关于 Bits(位)、Bytes(字节)和 Words(字)	(182)
6.6 Visual Basic 与 Dlls 间的参数传递	(184)
6.6.1 引用与传值	(185)
6.6.2 安全的类型声明	(187)
6.7 消息	(188)
6.7.1 发送消息	(188)
6.7.2 接收非 Visual Basic 消息	(189)
6.8 使用 Windows API 调用的例子	(189)
6.8.1 调用 GetFreeSpace()	(189)
6.8.2 动画	(190)
6.8.3 演奏乐曲	(192)
6.8.4 播放声音文件	(192)

6.8.5 动画与声音的演示程序	(193)
第7章 建立 Visual Basic 应用程序中的帮助系统	(203)
7.1 Windows 帮助系统的主要功能简介	(203)
7.1.1 “目录”按钮	(203)
7.1.2 “搜索”按钮	(204)
7.1.3 “历史记录”按钮	(204)
7.1.4 “<<”和“>>”按钮	(204)
7.2 帮助系统的开发步骤	(204)
7.3 编辑帮助专题源文件	(205)
7.3.1 帮助专题源文件的组成	(205)
7.3.2 帮助专题源文件的格式	(205)
7.3.3 使用 MS-WORD 编辑帮助专题源文件的方法	(205)
7.3.3.1 指定专题名、标识字符串、关键字和浏览顺序号	(205)
7.3.3.2 指定“热点”	(206)
7.3.3.3 创建不可滚动的标题区域	(208)
7.3.3.4 使用图形和图象	(208)
7.3.3.5 指定图形中的“热点”	(208)
7.4 建立帮助项目源文件	(210)
7.5 编译帮助源文件	(214)
7.6 在 Visual Basic 中调用 WinHelp 函数(Windows API 函数之一)	(214)
7.6.1 函数声明及函数的参数	(214)
7.6.2 调用方法	(215)
7.7 Visual Basic 应用程序中帮助系统的设计	(216)
7.7.1 上下文关联的帮助(F1 支持)	(216)
7.7.2 帮助模式(SHIFT + F1 支持)	(216)
7.7.3 在帮助菜单中调用帮助系统	(217)
7.7.4 退出帮助系统	(218)
7.7.5 在帮助模式中使用定制光标	(218)
7.7.5.1 编制用户的包括光标资源的 DLL	(218)
7.7.5.2 在 VB 应用程序中使用定制光标的方法	(219)
第8章 Visual Basic 4.0 使用初步	(222)
8.1 Visual Basic 4.0 的新增功能简介	(222)
8.1.1 集成开发环境	(222)
8.1.2 OLE 自动化	(222)
8.1.3 面向对象的程序设计方法	(223)
8.1.4 与 Visual Basic 应用程序版(Application Edition)兼容	(223)
8.1.5 条件编译与 32 位代码的支持	(223)

8.1.6 新的数据库引擎与数据约束控件	(223)
8.2 Visual Basic 4.0 的集成开发环境	(224)
8.2.1 新增命令	(224)
8.2.1.1 File(文件)菜单	(225)
8.2.1.2 Edit(编辑)菜单	(226)
8.2.1.3 View(视图)菜单	(227)
8.2.1.4 Insert(插入)菜单	(229)
8.2.1.5 Run(运行)菜单	(229)
8.2.1.6 Tools(工具)菜单	(229)
8.2.1.7 Add-Ins(添加项)菜单	(235)
8.2.2 集成开发环境的其它改进	(236)
8.2.2.1 属性窗口	(236)
8.2.2.2 代码窗口	(236)
8.2.2.3 调试窗口	(237)
8.3 API Viewer (API 函数查看工具)简介	(239)
8.3.1 API Viewer 概述	(239)
8.3.2 装入有关的 API 文件	(240)
8.3.3 将文本文件转换为 Jet 数据库文件	(240)
8.3.4 查看 Declares、Constants 或是 Types	(241)
8.3.5 将所需的 Declares、Constants 或 Types 引入 Visual Basic 代码中	(241)
8.4 语言概要与程序设计实例	(241)
8.4.1 基本概念	(241)
8.4.1.1 与对象有关的概念	(241)
8.4.1.2 与 OLE 自动化有关的概念	(242)
8.4.2 语言概要	(243)
8.4.2.1 数据类型	(243)
8.4.2.2 内部常数的使用	(245)
8.4.2.3 变量与过程的作用域	(246)
8.4.2.4 使用命名参量(Named Argument)	(247)
8.4.2.5 For Each...On...Next 语句	(248)
8.4.2.6 Set 语句	(249)
8.4.2.7 属性过程(Property Procedure)	(249)
8.4.2.8 Create Object 和 GetObject 函数	(251)
8.4.2.9 Err 对象	(255)
8.4.2.10 使用定制控件	(256)
8.4.2.11 条件编译指令	(257)
8.4.3 程序设计方法与实例	(258)
8.4.3.1 对象应用程序的设计方法与实例	(258)
8.4.3.2 集成环境中的 Add-Ins(添加项)的设计方法与实例	(266)

0 绪 论

0.1 Visual Basic 简介

Microsoft Windows 的出现,为 PC 计算机用户提供了一个直观的图形用户界面,使应用程序更易于学习和使用。它的图标、菜单、工具按钮以及对话框等,使用户摆脱了传统的命令行工作方式,人机交互变得生动有趣。屏幕上的多窗口可同时运行多个程序,各个程序间很容易互通信息,这一切都突破了 DOS 的限制,大大方便了用户,越来越多的 PC 用户转到了 Windows 环境下。有经验的用户,在商用软件不能满足他们的特殊需要时,往往想开发自己的应用程序。一旦用户尝试开发自己的 Windows 应用程序,立即会发现,开发 Windows 应用程序何其困难。通常 Windows 应用程序是用 C 语言开发的,困难不在语言的使用上,而在于用户界面的设计和掌握复杂的 Windows 的内部机制。程序员必须为屏幕上的每个元素编写具体的代码,去创建每一个对象,程序经过编译、连接,运行时也许会发现,得到的并不是所期望的,一切还得从头做起。不是专业程序员要想自己开发 Windows 应用程序,真是困难重重。开发 Windows 应用程序的困难,使非专业的用户望而却步。

Visual Basic 的出现,使这种情况顿时改观。Visual Basic 编程系统,引用部分面向对象的机制,用一种巧妙的方法把 Windows 的编程复杂性封装起来,提供了一种所见即所得的可视界面设计方法,同时保留了 Basic 语言的简单易用性,设计出的应用程序完全符合 Windows 规范。界面的设计几乎不用编写代码,窗口、菜单、字体、对话框及滚动条等都可在屏幕上直接“画”出来,可任意改变它的大小和外观。应用程序的开发人员可把注意力集中到应用程序的功能上而不是界面上。

Visual Basic 的另一个突出的特点是,它采用事件驱动的编程机制。事件驱动机制是非常适合图形用户界面的编程方式。传统的编程方法是一种面向过程,按顺序进行的工作,程序设计人员总是在想,这步运行完之后,下一步该干什么了。程序按预定的流程运行。但在图形用户界面的应用程序中,用户的动作掌握着程序的运行流向。程序响应用户的动作。程序员只要编写响应用户动作的代码,各个动作之间不一定有什么联系。这样一个应用程序的代码就不是一个大而冗长的程序,而是一段段没有紧密联系的相对较短的代码,这使得程序既易于编写又易于维护。

Visual Basic 提供了一个易学易用的应用程序集成开发环境,在这个环境中,用户可设计界面,编写代码,调试程序,最后可把应用程序编译成可执行文件,脱离 Visual Basic 环境而能在 Windows 中独立运行。同时还提供将应用程序制成可分发的形式(1.2M 或 1.44M 的安装软盘),这一功能在一般的开发环境中是不多见的。

Microsoft Visual Basic 目前有 for Windows 和 for DOS 两种版本,但 for DOS 的版本不多见,最为流行的是 Visual Basic 3.0 for Windows Professional Edition(专业版)。本书打算就此版本向有兴趣的读者介绍该软件使用的一般知识,程序设计的方法和某些比较高级但却常用

的专题。本书假设读者具有 Windows 的一般知识,或者至少学过一种 Basic 语言。若具有 QuickBasic 的编程知识,对学习 Visual Basic 十分有利。

0.2 Visual Basic for Windows 所要求的软硬件条件

1. 硬件要求:Visual Basic 对于硬件的要求不是很高,只要计算机能在 386 增强模式下运行 Windows 即可很好地运行 Visual Basic。最低的硬件配置为,386 SX 以上处理器,至少 2 兆内存,30 兆自由硬盘空间。VGA(单色或彩色)显示器。鼠标虽可任选,但没有鼠标会大大降低效率并显得十分笨拙。

2. 软件要求:Microsoft Windows 3.1 以上,运行在 386 增强模式下。若要编写中文应用程序,应使用中文 windows,或西文 windows 加中文之星或其它中文平台。

0.3 Visual Basic 3.0 专业版的组成

当在 Windows 中正确地安装了 Visual Basic 3.0 for Windows 之后,在 Windows 的程序管理器窗口中双击 VB 的程序组图标,则打开了 Visual Basic 程序组,显示如下窗口,其中列出了 Visual Basic 软件包的组成部分:

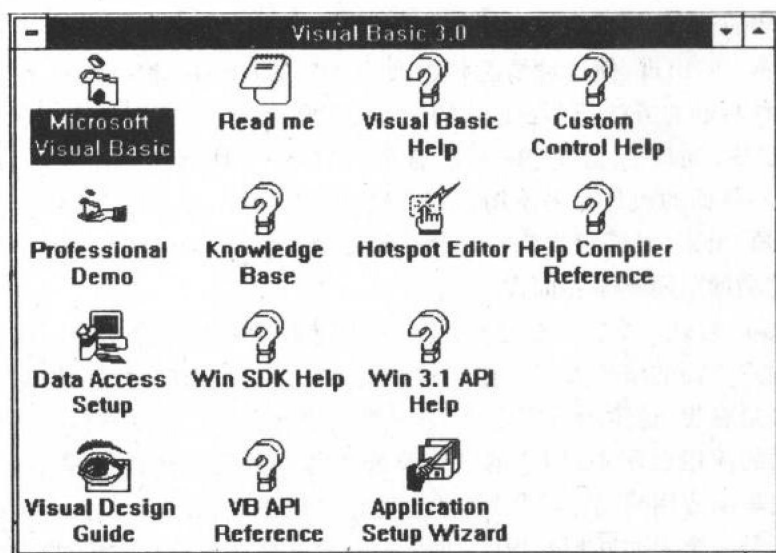


图 0-1 Visual Basic 3.0 程序组

- Visual Basic:这是 Visual Basic 的集成开发环境,程序开发的大部分工作都是在此环境中完成的。

- Read me:这是 Visual Basic 软件包的自说明文件,它包括了软件说明书附印以来又修改过的内容。

- Visual Basic Help:这是集成开发环境的帮助信息。从集成开发环境中的主窗口中的帮助菜单也可访问它。这部分内容十分有用,它几乎包括了 Visual Basic 中的所有有关内容的帮助信息。经常打开仔细阅读,对于提高在 Visual Basic 中的编程能力会有帮助。

● **Custom Control Help**:这是定制控件(所谓控件就是 Windows 中的执行一定功能的控制部件,例如,按钮、文本框、滚动条、列表框以及菜单等等)的帮助信息。Visual Basic 3.0 专业版中除有 Visual Basic 本身内建的控件工具之外,还带在很多 Microsoft 和第三方软件公司开发的控件,这些控件另需 Windows 的动态连接库(VBX)的支持,被称为定制控件。这里包括了有关定制控件的详细帮助信息。这在开发工作中也是十分有用的。

● **Professional Demo**:这是专业版的演示程序。运行这个程序会给你关于 Visual Basic 功能的初步印象。这里也有编程可借鉴的信息。

● **Knowledge Base**:Visual Basic 的知识库。这里的内容既涉及用户容易搞错的概念和方法,也包括有对较难解决的编程问题的解决方法,分门别类地列出了大约有近于 100 个问题的解决方法,其中附有示例,用户可将这些示例代码粘贴进自己的程序中使用。

● **Hotspot Editor**:热点编辑器。帮助是 Windows 应用程序的必要的组成部分之一,没有帮助信息的 Windows 应用程序还不能算是完善的。Visual Basic 中有一个帮助编译器,可用于对帮助源文件进行编译,以便成为 Windows 的帮助系统可以识别的信息。热点编辑器是制作帮助系统的辅助工具。它能定义图形中的“热点”,既可用于跳转也可用于弹出说明信息。关于此编辑器的用法在叙述到建立 Visual Basic 应用程序的帮助系统时,还要详细加以介绍。

● **Help Compiler Reference**:从名称上看,它是帮助编译器的参考资料。但实际上并非如此。它是关于帮助系统中使用的“宏”的参考资料。这部分在本书中未加深入的阐述。

● **Data Access Setup**:数据访问部件的安装程序。Visual Basic 中包括了 ODBC (Open DataBase Connectivity)部件用于访问数据库,这是 ODBC 的安装程序。

● **Win SDK Help**:Windows SDK (Software Development Kit)的帮助文件。与 SDK 中的完全相同。它提供了 Windows API 函数的背景知识,是深入 Visual Basic 编程的重要参考资料。

● **Win 3.1 API Help**:关于 Windows API 函数在 Visual Basic 中使用的帮助信息。其中列出 700 多 API 函数在 Visual Basic 中的函数声明语句,所有的符号常数的定义。用户只要将其中的语句和定义复制到自己的代码中即可使用。这既方便了使用也避免了产生错误的一大根源。此部分与本书的后续章节关系密切。

● **Visual Design Guide**:可视部件设计指南。对于 Windows 标准部件的大小、形状和颜色等的规范说明。用户在设计应用程序的界面时可以参考。

● **Application Setup Wizard**:应用程序的安装程序的生成向导。应用程序开发的最后一步就是将应用程序制成可分发的形式,本工具用于完成这一任务。Windows 应用程序比较复杂,不是简单地复制文件就能安装的。为了简化用户的工作,有必要生成一个安装程序,帮助用户正确安装软件,同时也减轻了用户的负担。这是软件易用性的不可缺少的一环。

● **其它**:图 0-1 中未能将 Visual Basic 软件包中的所有程序(用户选择全部安装时)列出。还有其它的应用工具,本书不打算介绍。

0.4 Visual Basic 应用程序组成

在开发过程中 Visual Basic 应用程序由以下文件组成:

1. 项目文件(*.MAK)在 Visual Basic 中一个应用程序被称为一个项目,项目文件中保

持着组成项目的所有源文件的记录。在 Visual Basic 中,用户并不直接与项目文件打交道。程序员利用一个被称为项目窗口的对象,来维护项目文件。

2. 窗体文件(*.FRM;*.FRX)在 Windows 中应用程序界面是由一个个窗口组成的,一个窗口有一个窗体文件,因此一个应用程序可有一个到多个窗体文件。它包括界面上的所有对象的记录,以及与窗体有关的代码。窗口中的图形保存在另外的文件中,其后缀为 .FRX。

3. 代码模块文件(.BAS)可为任何窗体中的过程调用的程序代码,放在另外的文件中保存,这就是代码模块文件。一个应用程序可有多个代码模块文件。

4. 定制控制文件(.VBX)只有在应用程序中使用了定制控制部件时,才需要这些文件。在 Visual Basic 中不能创建这些文件,这是由 Visual Basic 软件包中带来的或是向第三方软件开发商处买来的。

0.5 Visual Basic 应用程序的开发步骤

利用 Visual Basic 来开发应用程序,大致有以下几个步骤:

1. 设计应用程序的界面。应用程序要与用户打交道,因此需要人机交互的界面。在 Windows 中,使用的是图形用户界面。在传统的 Windows 应用程序的开发中,界面的设计占去了开发的大部分时间,这是应用程序开发过程中的主要工作量所在。Visual Basic 大大地简化了界面设计的难度,使得程序员可把注意力集中到程序的功能上而不是界面上。在 Visual Basic 中界面设计是一个可视的过程,这正是 Visual Basic 的名称的由来。

2. 编写响应用户输入动作的代码。当界面设计完成以后,要使程序运行,就得编写代码,以使程序与用户交互,响应用户的动作,完成用户指定的工作。

3. 调试程序。不用说大的应用程序,即使是一个小的程序,错误总是难免的。必须经过调试,才能正确地执行预定的任务。

4. 将应用程序的源程序编译为可执行的文件,利用 Visual Basic 中的 SETUP WIZARD (制作安装盘的向导)将应用程序制成可分发的形式(安装盘)。

前三步都可在集成开发环境中完成。第四步虽然不能在集成开发环境中完成,但使用的工具却是包括在 Visual Basic 软件包中。而这一步却是软件易用性中不可缺少的一环。一个软件必须安装在计算机中才能使用,软件的安装是软件与用户打交道的第一个步骤。大型应用程序十分复杂,用手工正确安装几乎是不可能的。制作安装盘就为用户解决了使用软件的第一步,因此是十分必要的。

这里介绍开发的步骤只是给读者一个初步的印象和基本的知识,以便于掌握本书的思路,更好地阅读和使用本书。

第 1 章 Visual Basic 应用程序的集成开发环境

假设你的计算机中已经安装好 Visual Basic 软件,像启动一般的 windows 应用程序一样启动 Visual Basic(在以下的叙述中简称为 VB),即在 VB 程序组中用鼠标双击 VB 图标,VB 即被启动。从而可以看到如下屏幕图景:

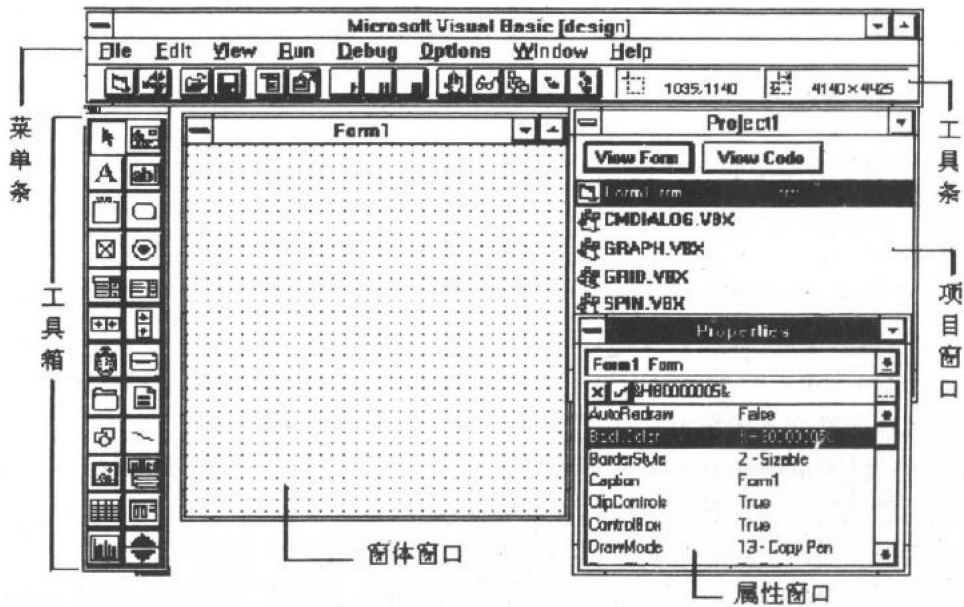


图 1-1 Visual Basic 集成开发环境

屏幕中包括以下几个独立的窗口,现对每一部分分别加以介绍。

1.1 主窗口

主窗口是图 1-1 中的屏幕图的最上部,单独列出如图 1-2,由标题栏、菜单条和工具条等组成。

1.1.1 标题栏

标题栏中的标题为 Microsoft Visual Basic[design],说明此时集成环境处于设计(design)模式。在运行程序时标题中的括号中的文字会变为 run,而在程序暂停处于调试阶段时,又会变为 debug。括号中的文字指明集成开发环境正在处于什么状态下。标题栏的左端带有一个小矩形的方框是 Windows 系统中的每个窗口都有的控制菜单框,可打开 Windows 的系统控制菜单,用于将窗口变形移动,或切换到其它的应用程序。标题栏的右边是最大化按钮和最小化按钮。这些都是 Windows 程序的标准部件。

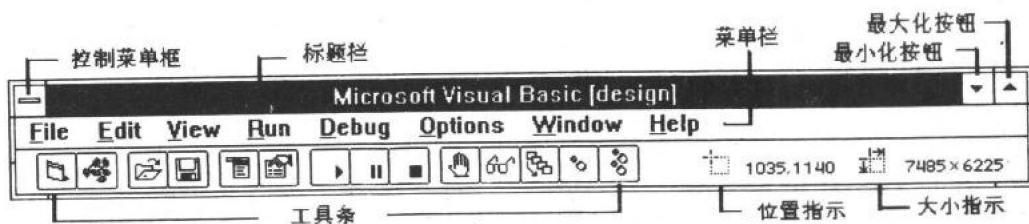


图 1-2 Visual Basic 主窗口

1.1.2 菜单条

菜单条中包括 8 个下拉式菜单,每个菜单中都有很多菜单项,这是程序开发过程中需要的各种命令。这里只介绍每个菜单的基本功能,每个菜单中的命令的用法留待用到时再加以详细叙述。

- File(文件)菜单用于维护程序文件(项目文件、窗体文件和代码文件等),执行诸如增加文件、删除文件、保存文件和打印等文件处理功能。
- Edit(编辑)菜单中的命令用于程序源代码的编辑,执行诸如复制、剪切、移动、粘贴以及查找和替换文本的编辑功能。
- View(查看)菜单中的命令负责集成开发环境的主窗口的外观显示(控制着工具条的显示与隐藏),以及代码窗口中过程的创建与显示。
- Run(运行)菜单用于控制程序的运行、暂停、继续(调试时)或结束。
- Debug(调试)菜单的命令与 Run 菜单中的命令配合使用用于程序的调试。可单步运行、设置断点和观察运行结果等。
- Option(选项)菜单用于设置集成环境中的环境设置,诸如制表位的长度、键入代码时是否检查语法,断点处的颜色,各种代码的颜色区别等,此外还设置程序的启动窗体或模块。
- Windows(窗口)菜单控制着对于调色板、工具箱窗口、菜单设计窗口、项目窗口和属性窗口的显示与隐藏。
- Help(帮助)菜单中的命令可调用 Windows 的帮助系统,显示 Visual Basic 的帮助信息和联机教学演示程序。

1.1.3 工具条

工具条位于菜单条的下面(图 1-2),其中共有十四个工具按钮。它们或分为三组。第一组为左面四个按钮,与文件菜单中的某些命令对应。第二组由紧靠前一组的两个按钮组成,与 Windows 菜单中的命令对应。其余的八个按钮可算为第三组,用于程序的运行与调试。

工具条可以隐藏起来,从 View 菜单中单击 Toolbar,则可使其隐去。当工具条显示时,Toolbar 命令前有选择标记。