

第一章 Visual Basic 概论

1990年5月,美国微软公司(Microsoft)推出 Windows 3.0 版的窗口操作系统后,彻底改变了个人计算机的面貌。本章的重点包括:

1. Windows 简史。
2. Basic 的发展过程。
3. Visual Basic 的特点。
4. 面向对象事件驱动程序设计。

1.1 Windows 简史

Windows 的图形用户界面(Graphical User Interfaces, GUI)已经成为软件开发的必然趋势。Windows 最早的开发可以追溯到 1983 年,但最早的 1.0 版却是在 1985 年 11 月推出。当时的电脑只有 2 个 360K 的软盘(没有硬盘),256K 的 RAM 和 8088 的 CPU。而产生的图形是无法重叠的。

1987 年 11 月,推出 Windows 2.0 版,这时所产生的图形可以相互重叠,但只能在 80x86 的真实模式(Real mode)下运行,也就是说只能在 1MB 的 RAM 下运行程序。

1990 年 5 月,微软公司推出 Windows 3.0 版,九个月之内就销售了三百万套,许多人也开始用它取代 DOS 成为新一代的操作系统。它的特点之一就是能支持 16MB 的 RAM。如果加上 386 增强模式的虚拟内存,则可高达 64MB。

1992 年 4 月,Windows 3.1 版问世,它改良了字体,将大部粗糙的棱角修饰得更平滑,称为真实字体(True type),同时也加强了许多强大的功能。图 1-1 是微软 Windows 的发展历史。



图 1-1 微软 Windows 的发展历史

1.1.1 窗口的构造

一个典型的窗口构造如图 1-2,由于窗口在构造上十分类似,因此如果熟悉它的功能与构造,在操作上就能得心应手。同样地,在使用 Visual Basic 之前,必须了解窗口的图形用户界面,这对设计 Windows 程序将有很大的帮助。窗口的各部分组织与功能含义如下:

系统菜单框(System Menu Box)

系统菜单框位于窗口的左上角。它包含了许多控制窗口的系统功能。例如:移动窗口的位罝、改变窗口的大小或关闭窗口等。如果将鼠标的光标移动窗口的系统菜单框上并按一下

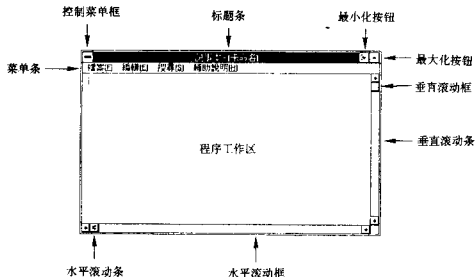


图 1-2 窗口的构造

鼠标左边的按钮,窗口会拉下系统菜单,如图 1-3 所示。

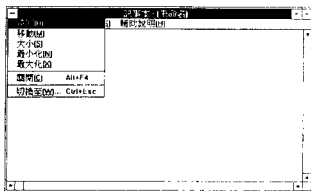



图 1-3 系统菜单框

标题栏 (Title Bar)

标题栏位于窗口的最上方。标题栏上会显示应用程序的标题名称。它可以用来移动窗口的位置,只要将鼠标的光标移到标题栏上,并按住鼠标的按钮不放,然后移动鼠标则窗口也会跟着移动。到达定位后,放开按钮,窗口就会固定在位置上。另外,如果我们打开了许多窗口,要辨别哪一个是在使用中的窗口,则只要注意标题栏的颜色即可,因为它的标题栏颜色会不同于其他窗口标题栏的颜色。

最小化按钮与最大化按钮 (Minimize Box and Maximize Box)

最小化按钮与最大化按钮在标题栏的右方。最小化按钮是个倒三角形。如果使用鼠标按下这个按钮,则整个窗口将缩小成一个图标 (Icon), 显示在屏幕的左下角。如果想要再打

开这个窗口。此时,原先的最大化按钮形状会变成一对上下的三角形(即)。如果要恢复原状,再按一次此按钮即可。

菜单栏(Menu Bar)

菜单项位于标题栏的下方。例如在图 1-4 的程序管理器菜单栏包含了:

- 文件[F]
- 选项[O]
- 窗口[W]
- 帮助[H]

等四项菜单。只要用鼠标按下菜单,窗口将拉下菜单中的所有选项清单。如图 1-5 是我们将鼠标移至“文件[F]”菜单,按下菜单所产生的图形。

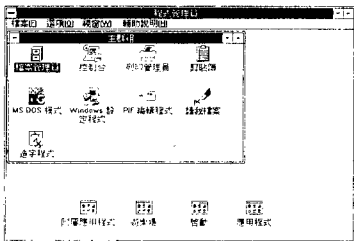


图 1-4 程序管理器

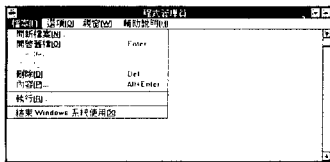


图 1-5 程序管理器的文件菜单

程序工作区(Client Area)

在菜单栏下方是程序工作区。它是一个长方形的区域,为用户提供了运行 Windows 程序所使用的工作空间。

水平滚动条及垂直滚动条(Horizontal Scroll Bar and Vertical Scroll Bar)

水平滚动条及垂直滚动条分别位于窗口的下方及右方的边界上,它们都各有一个滚动钮及两个显示滚动方向的箭头,它们的主要用途是水平或垂直地滚动窗口工作内的数据。而滚动工作区内的数据有三种方法:

- 使用箭头:用鼠标轻按滚动条两端的箭头,窗口内的数据会随着箭头的方向移动,它的移动范围较小。
- 使用滚动钮:用鼠标按住滚动钮不放,然后往左右(或上下)推动至定位放开。
- 使用滚动条:将鼠标在水平滚动条(或垂直滚动条)上按一下,则窗口内的数据会随之滚动,这种方式移动的范围较大。

窗口的边界(Frame)

窗口的边界就是窗口的外框。它的尺寸大小可以任意调整,只要将鼠标移到窗口的边框,则光标的形状会改变双箭头的形状(即 \leftrightarrow)。此时按住鼠标左边的按钮不放,移动鼠标,则窗口的大小会随鼠标的移动而改变。

1.2 Basic的发展过程

1963年美国达特茅斯学院(Dartmouth College)的约翰柯梅尼(John Kemeny)和汤姆斯库兹(Thomas Kurtz)两位教授开发出BASIC语言(Beginner's All purpose Symbolic Instruction Code,初学者万能符号指令码)。它主要是使用在学术界上,教导学生使用容易学习的计算机程序语言。这对于当时写程序必须熟悉复杂的作业控制语言(Job Control language)来说是一大进步。早期的BASIC语言可以说是一种“玩具语言”(toy language),无法适用于真实的程序工作上。

1970年代中期,个人计算机风行后,软微公司出产的DOS都装有GW-BASIC。虽然GW-BASIC在做简单的运算或工作十分快速,但软件工程界仍然不将它当作一回事。

1982年,微软公司推出Quick Basic。它继承了GW-BASIC的风格并提供更多的功能和现代语言结构的特点,例如:子程序(Subprograms)、先进的绘图能力和音乐程序。它也可以在解释器(Interpreter)和编译器(Compiler)下工作,产生可运行的程序。

1991年开发Windows程序的Visual Basic 1.0诞生了,它提供了比Quick Basic更强大的功能,对Windows程序开发者,它的出现真是一大福音。1992年微软公司推出Visual Basic 2.0版,1993年,Visual Basic 3.0版问世,它提供了更多的窗口应用程序。在下一节我们将讨论Visual Basic的特点,至于先前两版如何更新到3.0版,则请参考附录A。图1-6为BASIC的发展过程。

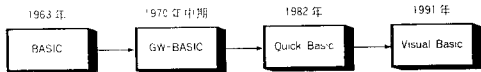


图 1-6 BASIC的发展过程

1.3 Visual Basic 的特点

各种语言都有它的特点,对于需要开发窗口软件的工作者来说,Visual Basic 是一个十分理想的窗口开发工具。以下就是 Visual Basic 的特性:

用户可以在短时间内成为 Windows 程序员

用 C 语言或窗口软件开发工具包(Windows Software Development Kit, SDK)开发应用程序,将会发现程序过于冗长而且繁杂。主要是因为图形用户界面设计就占了 80%~90% 的程序长度,而真正主题部分只占 10%~20% 而已。Visual Basic 所提供的界面设计工具,将很容易地创造所需要的图形界面,因而可以将精力花费在程序本身,增加软件程序的效率。

它是一个面向对象的程序设计软件

Visual Basic 是一种面向对象和事件驱动(Event-driven)的程序语言。它是 90 年代软件程序设计的趋势。依据这种设计方式,程序员不需要再跟着程序的流程循序开发,而是根据不同的事件运行不同的过程。它的详细说明请参阅下一节。

使用动态链接程序库(Dynamic Link Libraries, DLL)技术

为节省内存的空间,将链接的步骤往后移,直到程序运行时才链接。某个函数被调用时,才将这个函数放入内存链接。当然,也允许好几个程序使用这个函数,减少内存的浪费。这种在需要时才将函数放入内存中进行链接的方式,称为动态链接。使用 DLL 有四个好处:

- 节省内存的空间,可以共享程序资源。
- 可以加强程序运行的速度。

基本上,Visual Basic 的运行速度比 C 语言慢,所以可将 C 语言写成 DLL,供 Visual Basic 使用,以提高 Visual Basic 的速度。

- 具有扩充程序的能力。

有些 Visual Basic 很难达到的功能,可通过不同语言的特性制成 DLL,以扩充 Visual Basic 语言本身的能力。

- 许多软件公司提供功能强大的 DLL。

由于许多软件公司都从事于 DLL 的开发,因此,不必自己开发所有程序。在进行大的项目程序时,可以集成不同的 DLL,产生新的应用程序,以增加软件开发的效率。

使用动态数据交换(Dynamic Data Exchange, DDE)技术

所谓动态数据交换就是同时运行不同的应用程序,各程序之间互相交换数据,它的优点就是不同程序间的数据可以互相交换。

使用对象链接与嵌入(Object Linking and Embedding)技术

所谓对象链接与嵌入就是程序可以将其其他应用程序产生的对象以链接或嵌入的方式,放入自己的文件内,达到对象共享的目的。

1.4 Visual Basic 程序设计概念:面向对象概念

面向对象的技术也是经过时间的演变而形成的,程序设计的演变如图 1-7 所示。

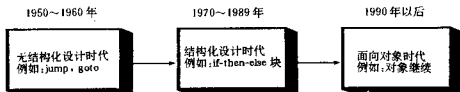


图 1-7 程序设计的演变

在 1950 年到 1960 年之间,程序设计是个混沌初开的时代,毫无结构可言,对程序的写可以说是件十分痛苦的事;1970 年到 1980 年间属于结构化程序设计的时代,这个时期的语言(如 Pascal、C 和 Ada)都极具代表性,而写程序也必须按照程序控制的顺序,执行时一行再一行地运行。

1990 年以后,软件设计展现新的风貌,它就是面向对象的技术。什么是对象呢?简单地说,对象就是数据(data)与程序码(code)互相结合的综合体。如果很熟悉 Windows 的话,会发觉每个图标(Icon)就是一个对象(如图 1-8),当用鼠标在图标上按两下,就表示已经选择了它并运行这个图标的程序码。这种程序的运行顺序是由外来的事件所决定的,而不是按照既定的顺序一行一行地运行。

因此面向对象程序可以说是由“事件驱动”(Event-driven)的程序。程序的运行完全依赖于事件的发生,如果没有任何事情发生,则对象处于停顿状态。而编写 Visual Basic 程序也十分简单,只要将程序码加到各对象或图标内并加以组合,就算大功告成了。

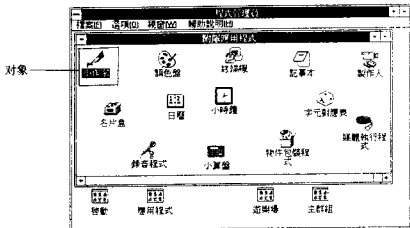


图 1-8 对象

1.5 习 题

1. 简述 Windows 的历史发展。
2. 说明 Basic 的演变。
3. Visual Basic 有何特点?
4. 简述窗口的构造。
5. 什么叫对象?

第二章 Visual Basic 的安装与系统介绍

Visual Basic 提供一个十分方便的安装程序,只要按照屏幕的指示,就可以很顺利的把 Visual Basic 安装在 Windows 下。本章的重点将包括:

1. Visual Basic 的系统需求
2. Visual Basic 的安装
3. Visual Basic 的启动方式
4. Visual Basic 的系统环境介绍
5. 鼠标的使用方法

2.1 Visual Basic 的系统需求

在安装 Visual Basic 之前,必须先检查计算机硬件设备和软件环境是否符合 Visual Basic 的要求。以下是 Visual Basic 最基本的需求:

硬件方面

1. 具有 80286 或更高微处理机的 IBM PC 或兼容计算机。
2. 一个硬盘。
3. 一个 5 $\frac{1}{4}$ 英寸或 3 $\frac{1}{2}$ 英寸软盘驱动器。
4. 一台 EGA、VGA、8514、Hercules 或兼容的屏幕显示器(其中以 VGA 的效果最佳)。
5. 至少 1MB 的内存(RAM)。
6. 一个鼠标。

软件方面

1. MS-DOS 3.1 版以上的操作系统。
2. Microsoft Windows 3.0 版以上的中文版或英文版软件。

2.2 Visual Basic 的安装

Visual Basic 可以在中文版或英文版的 Windows 下安装。如果安装在中文 Windows 下,则可以输入中文数据。本书将在中文版 Windows 3.1 下操作此软件。

安装 Visual Basic 主要是使用 Disk1 的 SETUP.EXE 文件。安装程序将帮助用户安装 Visual Basic、帮助系统(Help System)、示例应用程序(Sample Applications)、图标库(Icon Library)和其他磁盘内的软件。在这儿要注意的是,不能使用 COPY 或 Diskcopy 命令将 Visual Basic 拷贝到硬盘内。必须使用 SETUP 程序,因为它会将 Visual Basic 解压缩并安装至合适的目录内。

2.2.1 备份 Visual Basic

在安装 Visual Basic 之前,最好将 Visual Basic 备份(Backup),以免原版磁盘遭到破坏而

无法使用。备份的方法有下列三种：

在 MS-DOS 下使用 COPY 或 Diskcopy 命令

- a. 如果所使用的 A、B 驱动器大小都相同，则在 A 驱动器下键入

A: \>DISKCOPY A: B;

或

A: \>COPY *.* B;

- b. 如果 A、B 驱动器不同大小，则选择适合磁盘的磁盘驱动器（现在假设是 A 驱动器）放母片，键入

A: \>DISKCOPY A: A;

或

A: \>COPY *.* B;

在 Windows 下拷贝文件

- 进入主群组的文件管理器(File Manager)。
- 拉下“文件[F]”菜单。
- 选择“复制[C]...”命令。屏幕会出现如图 2-1 的对话框。

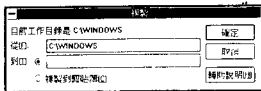


图 2-1 复制对话框

- 在对话框窗口的“从[F]”正文框输入 Visual Basic 母片的磁盘驱动器名称，在“到[T]”正文框输入空白磁盘的磁盘驱动器名称，然后按下“确定”命令按钮即可。

在 Windows 下复制磁盘

- 进入主群组的文件管理器(File Manager)。
- 拉下“磁盘[D]”菜单。
- 选择“复制磁盘[C]...”命令。如果有两个磁盘驱动器的话，屏幕会出现如图 2-2 复制磁盘对话框。如果计算机只有一个磁盘驱动器的话，屏幕不会出现此对话框。

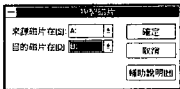


图 2-2 复制磁盘的对话框

- 假设 Visual Basic 母片在 A 驱动器，空白磁盘在 B 驱动器。那么在“来源磁盘在[S:]”

正文框内键入 A,而在“目的磁盘在[D:]”正文框内,键入 B,并按下“确定”钮。
e.接着屏幕会出现一个警告窗口,要求确认复制磁盘的命令,按下“是”命令钮即可。

2.2.2 Visual Basic 的安装程序

Visual Basic 3.0 for Windows 的 SETUP 程序只能在 Windows 下才能运行,而它的安装步骤如下:

1. 将 Disk1 磁盘插入 A 或 B 磁盘驱动器。
2. 进入程序管理器(Program Manager)的文件菜单或文件管理器(File Manager),选择“选择”命令,此时将看到运行的对话框。然后在命令行(Command Line)正文框内,键入:
a: \setup
或
b: \setup

视 Visual Basic 放在 A 驱动器或 B 驱动器而定,如图 2-3 所示。

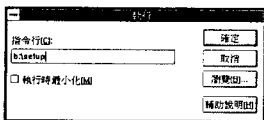


图 2-3 输入 Setup 命令

并按下确定(OK)钮,就开始进入安装程序了。

3. 接着屏幕出现安装画面,如图 2-4 所示,它的意思是:

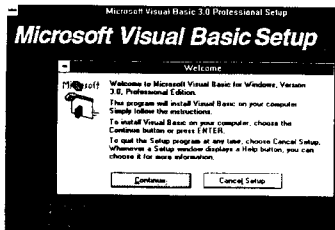


图 2-4 安装画面

欢迎使用 Microsoft Visual Basic for Windows 3.0 专业版。

只要遵循下列指示就可将 Visual Basic 安装至计算机内。

若想安装 Visual Basic, 请选 Continue 钮或按 ENTER 键。

若任何时候想退出安装, 请选择 Cancel Setup 钮。在任何时候若出现帮助(Help)按钮, 只要选它就可以获得更多帮助。

所以请按 **C**ontinue 钮或 **C** 键, 表示要安装 Visual Basic。

- 接着屏幕会出现用户信息的画面, 如图 2-5 所示。它要求输入用户的姓名和公司名称, 如果 Visual Basic 软件只供个人使用, 则只需要键入姓名即可, 接着按 **C**ontinue 钮或 **C** 键。

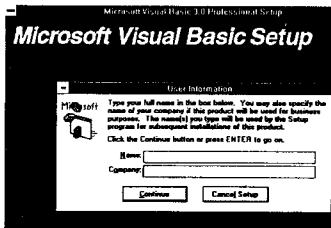


图 2-5 用户信息画面

- 屏幕将出现确认用户信息的画面, 如图 2-6, 以确认刚输入的数据, 如果正确, 请按 Continue 钮或 **C** 键。如果想更正所输入的数据, 请按 Retry 钮或 **R** 键。现在假设数据无误, 则按 **C**ontinue 钮或 **C** 键。

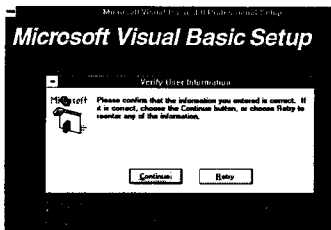


图 2-6 确认用户信息画面

6. 屏幕将出现安装 Visual Basic 位置的画面,如图 2-7。它的意思如下:

Visual Basic 会安装在下列的目录,如果目录不存在,会在您的计算机创建一个新目录。

如接受所建议的目录,选择 **Continue** 钮。如要更改所建议的目录,请用倒退键 (BACKSPACE) 删除文字,并键入适当的磁盘驱动器与目录。

现在假设我们接受它所建议的目录,就是将 Visual Basic 放在 C:\VB 内。按 **Continue** 钮或 C 键。

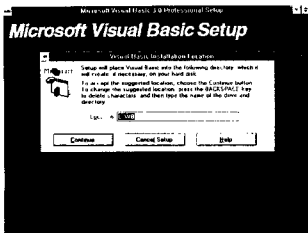


图 2-7 安装 Visual Basic 位置画面

7. 屏幕将出现目录不存在的画面,如图 2-8。它会说 C:\VB 目录不存在,是否要建立它。如果要,请按 **Create Directory** 钮或 C 键。如果要改变目录,请按 **Change Directory** 钮或 h 键。现在请按 **Create Directory** 钮或 C 键。

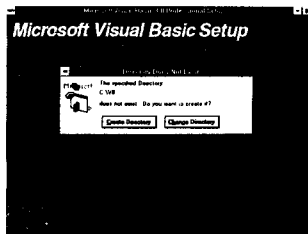


图 2-8 目录不存在画面

8. 接着出现图 2-9 的画面。它的意思是：

安装程序将在 Windows 目录内建立一些文件。这些文件将不会安装在 Visual Basic 目录内。

如果在下一个屏幕选择 Custom 方式安装 Visual Basic。安装程序将告诉安装这套软件各个单项所需的空间。

现在请按 **[OK]** 钮。

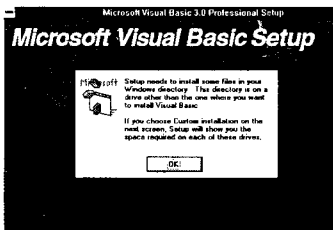


图 2-9 有些 Visual Basic 文件将安装在 Windows 目录的信息画面

9. 幕继续出现安装选项的画面，如图 2-10。它的意思是：

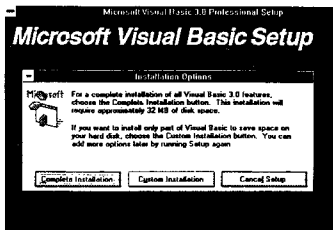


图 2-10 安装选项画面

若要完全安装整个 Visual Basic 3.0 版，请选择 **[Complete Installation]** 钮，专业版程序约需 32MB 的硬盘空间（标准版约需 12.5MB 的硬盘空间）。

如果只要安装部分的 Visual Basic，以节省硬盘空间。请选择 **[Custom Installation]** 钮。

以后可以重新运行安装程序增加所没有安装的部分。

现在假设我们有足够的硬盘空间,可以安装整个专业版程序,所以按

Complete Installation 钮或 C 键。

10. 屏幕接着出现安装 ODBC 驱动程序的画面,如图 2-11。它的意思是:

Visual Basic 3.0 将提供一组 ODBC 驱动程序,可安装到程序中,用来开发应用程序,以获取 Oracle 或 SQL 服务器的数据。

希望安装 ODBC 驱动程序吗?(以后也可以重新运行安装程序安装它们)。

现在假设我们不需要安装 ODBC 驱动程序,所以选择 **Don't install ODBC** 钮(如果需要安装此驱动程序,则选择 **Install ODBC** 钮)。然后安装程序就开始正式安装 Visual Basic 了,当所有安装过程无误后,将出现 Visual Basic 3.0 的图标,表示已经成功地安装好 Visual Basic 了。

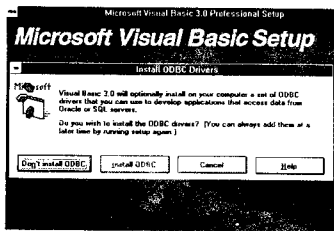


图 2-11 安装 ODBC 驱动程序画面

2.3 Visual Basic 的启动方式

启动 Visual Basic 的方法有三种:

1. 首先进入 Windows 并打开 Visual Basic 群组,如图 2-12 所示。然后连接两下 Microsoft Visual Basic 图标,则出现 Visual Basic 整个环境,如图 2-13。

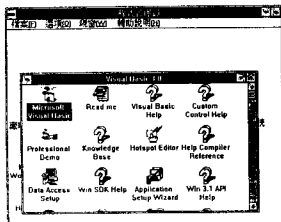


图 2-12 启动 Visual Basic

2. 进入 Windows 的主群组, 打开文件管理器 (File Manager) 窗口, 在 VB 目录下, 找到 VB.exe 并连接两下, 同样地会出现图 2-13 的画面。

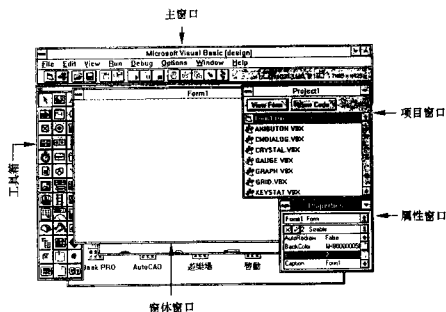


图 2-13 Visual Basic 的整个窗口

3. 若要在 MS-DOS 下打开 Visual Basic。
- 首先在 AUTOEXEC.BAT 文件内加入 Visual Basic 的路径, 即
PATH C:\VB;
 - 重新运行 AUTOEXEC.BAT 文件, 即
C:\>AUTOEXEC
 - 然后运行 VB.exe 文件, 即
C:\>VB

这样也能进入 Visual Basic 的窗口环境。图 2-13 是 Visual Basic 3.0 专业版 (Professional) 的画面,如果是标准版,在工具箱内只有 21 种工具。

2.4 Visual Basic 的系统环境介绍

进入 Visual Basic 的集成环境后,如图 2-13。可以看到 Visual Basic 的窗口环境包括了五大部分:

1. 主窗口(Main Window)
2. 工具箱(Toolbox)
3. 窗体(Form)
4. 项目窗口(Project Window)
5. 属性窗口(Properties Window)

2.4.1 主窗口(Main Window)

主窗口主要是提供程序设计时所需要的功能,例如:打开文件、存文件、退出 Visual Basic 等等。它包括了四个部分(如图 2-14):

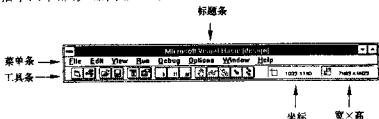


图 2-14 主窗口

1. 标题栏
2. 菜单(Menu)
3. 工具条(Toolbar)
4. 坐标度量

标题栏

Visual Basic 的标题栏除了显示 Microsoft Visual Basic 标题外,还显示了当前软件的工作状态。它共有三种工作状态(如图 2-15 所示)。

1. 设计(design)状态:一旦打开 Visual Basic 之后,就自动进入此状态内。它表示用户可设计程序。
2. 运行(run)状态:程序设计完毕后,下一步是运行程序。当运行程序时,将出现此工作状态。
3. 调试(debug)状态:如果想观看程序的错误之处,则在此状态下,屏幕将出现调试窗口(Debug Window)供检查。也可以在此窗口测试 Visual Basic 命令,每运行一条命令,窗口将立即出现结果。

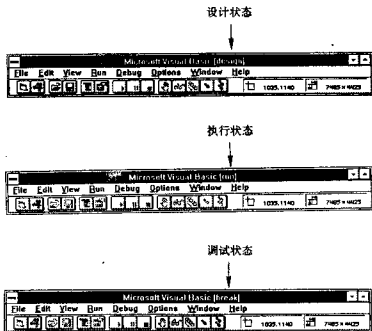


图 2-15 Visual Basic 的工作状态

菜单

菜单主要用于提供设计程序时所必需的功能,它是下拉式的选择列表(如图 2-16),以便程序设计人员选择命令或其他工具。例如,可以打开新的项目、窗体、打印或存储项目等。它的详细功能,请参考附录 B 中 Visual Basic 菜单介绍。

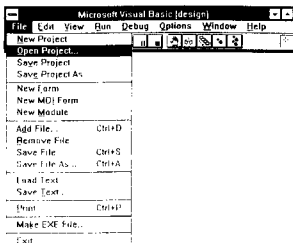
















图 2-16 File 下拉菜单

工具条(Toolbar)

工具条总共有 14 个功能钮,它主要提供能直接快速使用的命令,只按一个图标就可执行图标所代表的功能,当然这 14 项功能钮代表的功能都可以在菜单上找到。下面是功能钮图标所代表的含义:

图 标	含 义
	建立一个新的窗体(Form),也可以用 File 菜单内的 New Form 命令,执行相同的功能。
	建立一个新的模块(module),也可以用 File 菜单的 New Module 命令,执行相同的功能。
	打开现有的项目(Project),也可以用 File 菜单的 Open Project 命令,执行相同的功能。
	存储当前的项目,也可以用 File 菜单的 Save Project 命令,执行相同的功能。
	显示菜单设计窗口(Menu Design Window);也可以用 Window 菜单的 Menu Design 命令,执行相同的功能。
	显示属性窗口(Properties Window)。若属性窗口没在屏幕上出现,则按此图标以便显示它,也可以用 Window 菜单的 Properties 命令执行相同的功能。
	运行所设计的应用程序。也可以用 Run 菜单的 start 命令,执行相同的功能。
	当运行程序时,使程序暂停运行。也可以用 Run 菜单的 Break 命令执行相同的功能。
	当运行程序时,使程序停止运行,并回到程序的设计状态下,以便继续编辑或设计程序,也可以用 Run 菜单的 End 命令执行相同的功能。
	在当前的运行行上切换断点(breakpoint)。此功能在程序调试过程中,使程序运行到断点所在的行或解除该断点,也可以用 Run 菜单的 Toggle Break 命令,执行相同的功能。
	在程序编码窗口(Code Window)内显示当前选项的值,也可以用 Debug 菜单的 Instant Watch 命令执行相同的功能。
	显示当前程序调用(active calls)的结构。也可以用 Debug 菜单的 Calls 命令执行相同的功能。
	在程序码窗口(Code Window)内,一次运行一条命令。也可以用 Debug 菜单的 Single Step 命令执行相同的功能。
	在程序码窗口(Code Window)内,一次运行一个过程(Procedure)或语句(statement),也可以用 Debug 菜单的 Procedure Step 命令执行相同的功能。