
Visual Basic 语言教程

第1章 Visual Basic程序开发环境

Visual Basic是Microsoft公司推出的可视化BASIC语言。在语言功能方面，它与以前的BASIC语言（GWBASIC、BASICA、QuickBASIC等）基本兼容，但功能更强，使用更方便。在这一章中，我们将介绍Visual Basic的特点及Visual Basic 6.0版的集成开发环境。

1.1 Visual Basic的特点和版本

Visual Basic是在原来的BASIC语言的基础上研制而成的，它具有BASIC语言简单而不贫乏的优点，同时增加了结构化和可视化程序设计语言的功能。

1.1.1 Visual Basic的特点

Visual Basic是一种可视化的、面向对象和采用事件驱动方式的结构化高级程序设计语言，可用于开发Windows环境下的各类应用程序。它简单易学、效率高，且功能强大，可以与Windows的专业开发工具SDK相媲美，而且程序开发人员不必具有C/C++编程基础。在Visual Basic环境下，利用事件驱动的编程机制、新颖易用的可视化设计工具，使用Windows内部的应用程序接口（API）函数，以及动态链接库（DLL）、动态数据交换（DDE）、对象的链接与嵌入（OLE）、开放式数据访问（ODBC）等技术，可以高效、快速地开发出Windows环境下功能强大、图形界面丰富的应用软件系统。

随着版本的提高，Visual Basic的功能也越来越强。5.0版以后，Visual Basic推出了中文版，与以前各版本相比，其功能有了质的飞跃，已成为32位的、全面支持面向对象的大型程序设计语言。在推出6.0版时，Microsoft公司在数据访问、控件、语言、向导及Internet支持等方面增加了许多新的功能。

总的来看，Visual Basic有以下主要特点：

1. 可视化编程

在用传统程序设计语言设计程序时，都是通过编写程序代码设计用户界面，在设计过程中看不到界面的实际显示效果，必须编译后运行程序才能观察。如果对界面的效果不满意，还要回到程序中修改。有时候，这种编程-编译-修改的操作可能要反复多次，大大影响了软件开发效率。Visual Basic提供了可视化设计工具，把Windows界面设计的复杂性“封装”起来，开发人员不必为界面设计而编写大量程序代码。只需要按设计要求的屏幕布局，用系统提供的工具在屏幕上画出各种“部件”，即图形对象，并设置这些图形对象的属性。Visual Basic自动产生界面设计代码，程序设计人员只需要编写实现程序功能的那部分代码，从而大大提高了程序设计的效率。

2. 面向对象的程序设计

4.0版特别是5.0版以后的Visual Basic全面支持面向对象的程序设计，但它与一般的面向对象的程序设计语言（如C++）不完全相同。在一般的面向对象程序设计语言中，对象由程序代码和数据组成，是抽象的概念；Visual Basic则是应用面向对象的程序设计方法（OOP），把程序和数据封装起来作为一个对象，并为每个对象赋予应有的属性，使对象成为实在的东西。在设计对象时，不必编写建立和描述每个对象的程序代码，而是用工具画在界面上，Visual Basic自动生成对象的程序代码并封装起来。每个对象以图形方式显示在界面上，都是可视的。

继承和多态是面向对象语言的重要特征，Visual Basic 4.0支持继承，但不支持多态性，而Visual Basic 5.0和6.0版本增加了支持多态性的功能。多态性意味着多个类提供相同的方法或属性，而在调用属性或方法之前不必知道它属于哪一个类，这是一种代码重用的机制，在Visual Basic 5.0和6.0中通过Imolement语句实现。大多数面向对象的程序设计语言通过继承支持多态性，而Visual Basic 5.0和6.0采用组件对象模型（COM）的多个接口的方法实现多态性。

3. 结构化程序设计语言

Visual Basic是在BASIC语言的基础上发展起来的，具有高级程序设计语言的语句结构，接近于自然语言和人类的逻辑思维方式，其语句简单易懂。其编辑器支持彩色代码，可自动进行语法错误检查，具有功能强且使用灵活的调试器和编译器。

Visual Basic是解释型语言，在输入代码的同时，解释系统将高级语言分解翻译成计算机可以识别的机器指令，并判断每个语句的语法错误。在设计Visual Basic程序的过程中，随时可以运行程序，而在整个应用程序设计好之后，可以编译生成可执行文件（.EXE），脱离Visual Basic环境，直接在Windows环境下运行。

在以前的版本中，Visual Basic将应用程序编译成伪代码（P-code），它需要解释程序的支持。Visual Basic 5.0和6.0提供了一个本机代码编译程序，既可以将代码编译成标准的Visual Basic伪代码格式，也可以编译成Intel本机代码（即二进制代码）格式。伪代码是介于Basic程序中的高级指令和计算机处理器执行的低级代码之间的一种中间步骤。在运行期间，Visual Basic将每一句伪代码转换成二进制代码。如果将程序直接编译成本机代码，则取消了伪代码这一中间步骤。本机代码编译提供了用于优化和调试的选项，这些选项是伪代码编译所没有的。同时，编译生成的本机代码文件可以在Visual C++环境中调试和优化。

4. 事件驱动编程机制

Visual Basic通过事件来执行对象的操作。一个对象可能产生多个事件，每个事件都可以通过一段程序响应。例如，命令按钮是一个对象，当用户单击该按钮时，将产生一个“单击（Click）”事件，而在产生该事件时执行一段程序，用来实现指定的操作。

在用Visual Basic设计大型应用软件时，不必建立具有明显开始和结束的程序，而是编写若干个微小的子程序，即过程。这些过程分别面向不同的对象，由用户操作引发某个事件来驱动完成某种特定的功能，或者由事件驱动程序调用通用过程来执行指定的操作。这样可以方便编程，提高效率。

5. 访问数据库

Visual Basic 系统具有很强的数据库管理功能。利用数据控件和数据库管理窗口，可以直接建立或处理Microsoft Access 格式的数据库，提供强大的数据存储和检索功能。同时，Visual Basic 还能直接编辑和访问其他外部数据库，如Btrieve、dBASE、FoxPro、Paradox等，这些数据库格式都可以用Visual Basic 编辑和处理。

Visual Basic 提供开放式数据连接（Open DataBase Connectivity），即ODBC功能，可通过直接访问或建立连接的方式使用并操作后台大型网络数据库，如SQL Server、Oracle等。在应用程序中，可以使用结构化查询语言SQL数据标准直接访问Server上的数据库，提供简单的面向对象的库操作指令和多用户数据库访问的加锁机制，以及网络数据库的SQL的编程技术，为单机上运行的数据库提供SQL网络接口，以便在分布式环境中快速而有效地实现客户机/服务器（Client/Server）方案。

6. 动态数据交换（DDE）

利用动态数据交换（Dynamic Data Exchange）技术，可以把一种应用程序中的数据动态地链接到另一种应用程序中，使两种完全不同的应用程序交换数据并进行通信，在Windows 环境下为多个应用程序之间以Client/Server方式建立起一条动态数据链路。当原始数据变化时，可以自动更新链接的数据。Visual Basic 提供了动态数据交换的编程技术，可以在应用程序中实现与其他Windows应用程序建立动态数据交换，在不同的应用程序之间通信。

7. 对象的链接与嵌入（OLE）

对象的链接与嵌入（Object Linking and Embedding）将每个应用程序都看作是一个对象（Object），将不同的对象链接（Link）起来，再嵌入（Embed）某个应用程序，从而得到具有声音、影像、图像、动画、文字等各种信息的集合式的文件。OLE技术（或ActiveX技术）是Microsoft公司对象技术的战略，它把多个应用程序合为一体，将每个应用程序看作是一个对象进行链接和嵌入，是一种应用程序一体化的技术。利用OLE技术，可以方便地建立复合式文档（Compound Document），这种文档由来自多个不同应用程序的对象组成，文档中的每个对象都与原来的应用程序相联系，并可执行与原应用程序完全相同的操作。

在Visual Basic 5.0和6.0中，大多数控件都支持应用程序之间的拖放操作，例如，可以把Word文档的内容拖到文本框中。此外，还可以在控件与控件之间、控件与其他窗口应用程序之间拖放文本和图形。例如，可以在Excel中选择一个单元格，然后将其拖到应用程序的数据绑定（Bunding）网格控件中。在控件之间拖放时，移动的只是数据，不包括代码。

8. 动态链接库（DLL）

Visual Basic 是一种高级程序设计语言，不具备低级语言的功能，对访问机器硬件的操作不太容易实现。但它可以通过动态链接库（DynamicLinkingLibrary）技术将C/C++或汇编语言编写的程序加入到Visual Basic 应用程序中，可以像调用内部函数一样调用用其他语言编写的函数。此外，通过动态链接库，还可以调用Windows应用程序接口（API）函数，实现SDK所具有的功能。

9. 建立用户自己的ActiveX控件

ActiveX控件（以前称OLE控件）是标准的用户界面元素。在Visual Basic 5.0和6.0中，可以建立用户自己的ActiveX控件（在以前的版本中，只能通过Visual C++建立）。由用户建立的控件可以引发事件、支持数据绑定、具有自己的属性和方法，可以和标准控件一样使用，可以把它作为开发环境和运行环境的一部分，为应用程序提供新的功能。利用Visual Basic 5.0和6.0，可以从零开始建立一个新的ActiveX控件，也可以对原有的控件进行修改或者组合多个控件。用Visual Basic 5.0或6.0建立的ActiveX控件具有许多Internet特性，包括异步数据下载和超链接，其主要特征是向外提供对象及其属性和方法，用来实现指定的功能或操作，以便提高开发效率，缩短开发周期。

10. 建立ActiveX文档

ActiveX文档提供内置的视口滚动、超链接和菜单组合，并可把窗体放入Internet浏览器。可以像建立一般的窗体一样建立ActiveX文档，并能包含可插入对象，显示信息框和下一级窗体。由Visual Basic建立的ActiveX文档可用于Microsoft Office。

用Visual Basic建立的ActiveX文档与Visual Basic的窗体大致相同，不能直接运行，只能在ActiveX文档容器中使用，一般通过访问文件激活。ActiveX文档文件的扩展名为.VDP。

11. Internet组件下载

在传统的软件发行中，用户一般用Setup程序安装组件或使组件升级到新版本。在安装过程中，通过Setup程序确定选项，检查需要的文件是否已全部安装。在Internet中，一般通过浏览器来使用应用程序，无需运行Setup程序。当在浏览器上访问应用程序时，除了已有的文件之外，其余所需要的文件全部拷贝到用户的硬盘上。对于用户来说，在Web上安装和使用应用程序几乎没有区别，用Web浏览器（例如Internet Explorer）展开应用程序可以为用户建立一个一致的环境。这样，当用户处在Web上时，就可以从一个文档移动到另一个文档，还可以在HTML页面和ActiveX文档之间定位（不管它们是用Visual Basic建立的，还是由其他ActiveX部件，比如Microsoft Word和Microsoft Excel提供的）。

在以前的版本中，用Setup向导只能建立传统的安装程序；而在Visual Basic 5.0和6.0版中，可以用Setup向导将应用程序打包成能在Web上展开的形式，从而在Internet中下载。

12. 枚举类型

Ada、Pascal、C等语言中都有枚举类型。枚举类型提供了一种方便的方法，用来处理有关的常数或使名称与常数数值相关联。例如，可以为与星期日期相关联的一组整数常数声明一个枚举类型，然后在代码中使用星期的名称而不使用其整数数值。Visual Basic 5.0和6.0版本新增加了枚举类型，并可包含在ActiveX的类型库中。

以上介绍了Visual Basic的一些主要特性。除上述特性外，Visual Basic 5.0和6.0还新增加了其他功能，包括有关多线程和线程安全的ActiveX组件、远程自动化工具直接支持远程操作、ActiveX数据对象（ADO）和ADO数据控件，增强了远程数据对象（RDO）和远程数据控件（RDC）的功能，并具有声明、触发、管理自定义事件的功能等。

1.1.2 Visual Basic的版本

Microsoft公司于1991年推出Visual Basic 1.0版，获得巨大成功，于1992年秋天推出2.0版，1993年4月推出3.0版，1995年10月推出4.0版，1997年推出5.0版，1998年推出6.0版。随着版本的改进，Visual Basic已逐渐成为简单易学、功能强大的编程工具。不但在一般的程序设计方面比传统的程序设计语言具有明显的优势，而且可作为多媒体创作工具，支持程序跳转，即从静态目标跳转到音乐、声音、动画、图形等目标；同时具有脱机播放动画的能力，能较好地控制动画的清晰度、运动速度和方向、动画过渡技术如淡入淡出、渐隐渐现等。Visual Basic的软件费用支出低、易学习及其功能的突出特点，使之易于被非专业人员掌握并使用。

从1.0版到4.0版，Visual Basic只有英文版。5.0版以后的Visual Basic在推出英文版的同时，推出了中文版，这大大方便了中国用户。

Visual Basic 6.0包括3种版本，分别为学习版、专业版和企业版。这些版本是在相同的基础上建立起来的，因此大多数应用程序可在3种版本中通用。3种版本适合于不同的用户层次。

(1) 学习版：Visual Basic的基础版本，可用来开发Windows 9x和Windows NT应用程序。该版本包括所有的内部控件（标准控件）和网格（Grid）控件、Tab对象以及数据绑定（DataBound）控件。

(2) 专业版：该版本为专业编程人员提供了一整套用于软件开发的功能完备的工具。它包括学习版的全部功能，同时包括ActiveX控件、Internet控件、Crystal Report Writer和报表控件。

(3) 企业版：可供专业编程人员开发功能强大的组内分布式应用程序。该版本包括专业版的全部功能，同时具有自动化管理器、部件管理器、数据库管理工具、Microsoft Visual Source Safe面向工程版的控制系统等。

这3种版本中，企业版本功能最全，而专业版本包括了学习版本的功能。用户可根据自己的需要购买不同的版本。但是应注意，企业版本的价格较高，如果不是绝对需要，一般不必购买，以免造成不必要的浪费。对于大多数用户来说，专业版本完全可以满足需要。

本书使用的是Visual Basic 6.0中文企业版，其内容可用于专业版和学习版，所有程序可以在专业版和学习版中运行。

Visual Basic 6.0是专门为Microsoft的32位操作系统设计的，可用来建立32位的应用程序。在Windows 9x、Windows NT或Windows 2000环境下，用Visual Basic 6.0的编译器可以自动生成32位应用程序。这样的应用程序在32位操作系统下运行，速度更快，更安全，并且更适合在多任务环境下运行。

1.2 Visual Basic的启动与退出

Visual Basic应用程序的开发是在一个封闭的集成环境中完成的。为了用Visual Basic开发应用软件，必须了解这个集成环境。

Visual Basic 6.0可以在多种操作系统下运行，包括Windows 95、Windows 98、Windows NT 4.0和Windows 2000。为了叙述方便，在本书中一般称作Windows。此外，除非特殊说明，“Visual Basic”一般指的是Visual Basic 6.0。

Visual Basic 6.0是Visual Studio 6.0套装软件中的一个成员，它可以和Visual Studio 6.0一起安装，也可以单独安装。单独安装的Visual Basic 6.0中文版包括4张光盘，其中两张为MSDN。安装方式不同，启动方式也略有区别。在这里，我们假定使用的Visual Basic 6.0是单独安装的。

开机并进入中文Windows后，可以用多种方法启动Visual Basic。

1. 第一种方法

使用“开始”菜单中的“程序”命令，其操作如下：

- (1) 单击Windows环境下的“开始”按钮，弹出一个菜单，把鼠标光标移到“程序”命令上，将弹出下一个级联菜单。
- (2) 把鼠标光标移到“Microsoft Visual Basic 6.0中文版”，弹出下一个级联菜单，即Visual Basic 6.0程序组。
- (3) 单击“Microsoft Visual Basic 6.0中文版”，即可进入Visual Basic 6.0编程环境。

2. 第二种方法

使用“我的电脑”，其操作如下：

- (1) 双击“我的电脑”，弹出一个窗口。单击Visual Basic 6.0所在的硬盘驱动器盘符，将打开相应的驱动器窗口。
- (2) 单击驱动器窗口中的vb60文件夹，打开“VB60”窗口。
- (3) 双击“vb6.exe”图标，即可进入Visual Basic 6.0编程环境。

3. 第三种方法

使用“开始”菜单中的“运行”命令，其操作如下：

- (1) 单击“开始”按钮，弹出一个菜单。单击“运行”命令，将弹出一个对话框。
- (2) 在“打开”栏内输入Visual Basic 6.0启动文件的名字（包括路径），例如
c:\vb98\vb6.exe
- (3) 单击“确定”按钮，即可启动Visual Basic 6.0。

4. 第四种方法

建立启动Visual Basic 6.0的快捷方式（具体操作请参见有关资料）。

用上面介绍的任何一种方法启动Visual Basic 6.0后，将首先显示版权屏幕，说明此份程序拷贝的使用权属于谁。稍候，显示“新建工程”对话框，如图1.1所示。图中所示的是“新建”选项卡，如果单击“现存”或“最新”选项卡，可分别显示现有的或最新的Visual Basic 应用程序文件名列表，从中选择要打开的文件名。



图1.1 “新建工程”对话框（“新建”选项卡）

“新建”选项卡对话框显示了可以在Visual Basic 6.0中使用的工程类型，即可以建立的应用程序，主要有：

- 标准EXE程序：建立一个标准的EXE工程。
- ActiveX EXE和ActiveX DLL程序：这两种应用程序只能在专业版和企业版中建立。从功能上说，这两种程序是一致的，只是包装不一样，ActiveX EXE程序包装成可执行文件，而ActiveX DLL程序包装成动态链接库（DLL）。
- ActiveX控件：只能在专业版或企业版中建立，主要用于开发用户自定义的ActiveX控件。
- VB应用程序向导：用于在开发环境中直接建立新的应用程序框架。用户在开发自己的工程时，可以使用这个工具。
- 数据工程：为编程人员提供开发数据报表应用程序的框架，选择该图标后，将自动打开数据环境设计器和数据报表设计器。
- IIS应用程序：这是Visual Basic 6.0版新增加的工程类型，可以用Visual Basic代码编写服务器方的Internet应用程序，用来响应由浏览器发出的用户需求。
- 外接程序：选择该类型后，用户可以建立自己的Visual Basic外接程序，并在开发环境中自动打开链接设计器。
- ActiveX文档EXE和ActiveX文档DLL程序：选择该类型，可以建立在超链接环境（即Web浏览器，如Microsoft Internet Explorer）中运行的Visual Basic应用程序。
- DHTML应用程序：只能在专业版或企业版中建立。用DHTML应用程序可以编写响

应HTML页面操作的Visual Basic代码，并可处理过程传送到服务器。

▪ VB企业版控件：该图标不用于建立应用程序，而是在工具箱中加入企业版控件图标。选择该图标后，企业版控件将出现在工具箱中。

在对话框中选择要建立的工程类型（例如“标准EXE”），然后单击“打开”按钮，即进入Visual Basic 6.0集成环境，如图1.2所示。

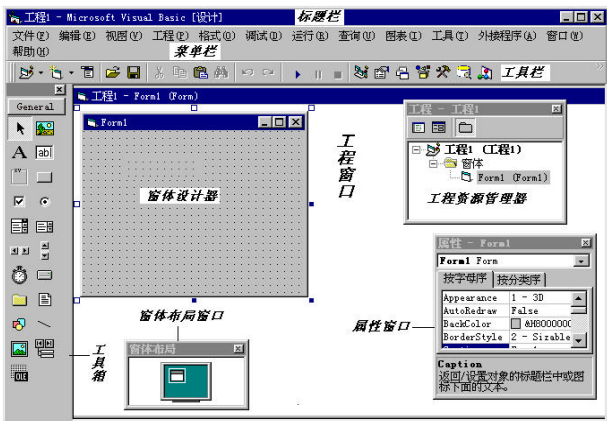


图1.2 Visual Basic 6.0编程环境（MDI方式）

Visual Basic 6.0提供了多种工程类型，以满足不同的需要。每次启动Visual Basic时，都要显示“新建工程”对话框。在一般情况下，可能使用“标准EXE”工程，因此没有必要在每次启动Visual Basic时显示该对话框，这可以通过选择对话框（见图1.1）左下角的“不再显示这个对话框”选项来实现（用鼠标单击小方框，使框内有“√”）。选择该选项后，再选择“标准EXE”，然后单击“打开”按钮，进入Visual Basic开发环境。以后再启动Visual Basic时，即不再显示“新建工程”对话框，而直接进入开发环境。

在缺省情况下，Visual Basic 6.0的集成开发环境为传统的Windows MDI（多文档界面）方式（图1.2），也可以用SDI（单文档界面）方式启动Visual Basic设计。在多数情况下，使用SDI方式可能更方便。为了把编程环境变为SDI方式，可执行“工具”菜单中的“选项”命令，打开“选项”对话框，选择“高级”选项卡，在对话框中选择“SDI开发环境”选项，然后单击“确定”按钮。这样设置后，退出Visual Basic，然后重新启动，即可按SDI方式进入Visual Basic集成开发环境。

从图1.2可以看出，启动Visual Basic后，屏幕上分为若干部分，包括标题栏、菜单栏、工具栏、工具箱、窗体设计器、工程资源管理器窗口、窗体布局窗口和属性窗口。为了能清楚地看到每个部分，对原来的各部分进行缩放并重新排列。用户在启动Visual Basic后，所看到的各部分的排列情况可能与图1.2所示的有微小差别，一些窗口会重叠。实际上，和Windows下的窗口一样，集成开发环境中的每个窗口都可以在屏幕上移动、缩小、放大或关闭。此外，

Visual Basic 保存上一次使用时屏幕上各部分最后的排列方式，并作为下一次启动 Visual Basic 后的屏幕布局。

为了退出 Visual Basic，可先打开“文件”菜单，执行其中的“退出”命令（或按 Alt+Q）。如果当前程序已修改过并且没有存盘，系统将显示一个对话框，询问用户是否将其存盘，此时选择“是”按钮则存盘，选择“否”按钮则不存盘。在上述两种情况下，都将退出 Visual Basic，回到 Windows 环境。

1.3 主窗口

主窗口也称为设计窗口。启动 Visual Basic 后，主窗口位于集成环境的顶部，该窗口由标题栏、菜单栏和工具栏组成（参见图 1.2）。

1.3.1 标题和菜单

1. 标题栏

标题栏是屏幕顶部的水平条，显示应用程序的名字。用户与标题栏之间的交互关系由 Windows 处理，而不是由应用程序处理。启动 Visual Basic 后，标题栏中显示的信息为：

工程1-Microsoft Visual Basic[设计]

方括号中的“设计”表明当前的工作状态是“设计阶段”。随着工作状态的不同，方括号中的信息不同，可能会是“运行”或“Break”，分别代表“运行阶段”或“中断阶段”。这三个阶段也分别称为“设计模式”、“运行模式”和“中断模式”。

2. 菜单栏

在标题栏的下面是集成环境的主菜单。菜单栏中的菜单命令提供了开发、调试和保存应用程序所需要的工具。Visual Basic 6.0 中文版的菜单栏共有 13 个菜单项，即文件、编辑、视图、工程、格式、调试、运行、查询、图表、工具、外接程序、窗口和帮助。每个菜单项含有若干个菜单命令，执行不同的操作。用鼠标单击某个菜单项，即可打开该菜单，用鼠标单击菜单中的某一条就能执行相应的菜单命令。例如，单击“文件”，可以打开文件菜单，如图 1.3 所示。打开菜单后，如果单击“打开工程”，可以打开已有的工程文件；而如果单击“工程另存为”，就可以保存文件，等等。在以后的叙述中，我们把形如上面的操作记为“执行文件菜单中的‘打开工程’命令”、“执行文件菜单中的‘工程另存为’命令”。

菜单中的命令分为两种类型，一类是可以直接执行的命令，这类命令的后面没有任何信息（例如“保存工程”）；另一类在命令名后面带有省略号（例如“打开工程”），需要通过打开“对话框”来执行。在用鼠标单击后一类命令后，屏幕上将显示一个对话框，利用该对话框，可以执行各种有关的操作。在“文件”菜单中，“新建工程”、“保存工程”等是可以直接执行的命令，而“打开工程”、“工程另存为”等命令必须通过对话框来执行。此外，从“文件”菜单可以看出，在有些命令的后面还带有其他信息，例如

打开工程...	Ctrl+O
保存工程	Ctrl+S



图1.3 “文件”菜单

其中，“Ctrl+O”等叫做“热键（或快捷键）”。在菜单中，热键列在相应的菜单命令之后，与菜单命令具有相同的作用。使用热键方式，不必打开菜单就能执行相应的菜单命令。例如，按Ctrl+O，可以立即执行“打开工程”命令。注意，只有部分菜单命令能通过热键执行。

上面介绍了通过鼠标和热键执行菜单命令的方法。除鼠标外，也可以通过键盘执行菜单命令。只有在打开菜单后，才能选择所需要的命令，执行相应的操作。Visual Basic 6.0提供了多种打开菜单和选择菜单的方法，用户可以根据自己的兴趣或习惯选用其中的一种。

(1) 第一种方法

① 按F10或Alt键，激活菜单栏，此时第一个菜单项（“文件”）被加上一个浅色的框。

② 按菜单项后面的括号中的字母键，打开菜单，下拉显示该菜单项的命令。菜单被打开后，各菜单命令后面的括号内都有一个字母。

③ 按菜单命令后面括号中的字母键，即可执行相应的命令。

(2) 第二种方法

① 按F10或Alt键，激活菜单栏，此时第一个菜单项（“文件”）被加上一个浅色的框。

② 用“→”或“←”把条形光标移到需要打开的菜单上，按回车键，打开该菜单。

③ 菜单被打开后，条形光标覆盖在第一个或上一次执行的菜单命令上。用“↑”或“↓”把条形光标移到所需要的命令上，按回车键即可执行条形光标所在位置的菜单命令。

(3) 第三种方法

① 按下Alt键，不要松开，接着按需要打开菜单项后面括号中的字母键，然后松开（Alt键松开），该菜单即被打开。

② 按菜单命令后面括号中的字母键，即可执行指定的菜单命令。

例如，为了执行“文件”菜单中的“打开工程”命令，可以如下操作：按住Alt键，不要

松开，接着按F键，先后松开F键和Alt键，再按O键即可。我们把上述过程记作：Alt+F,O。

除上述3种方法外，有些菜单命令还可以通过热键执行（见前面）。对于没有热键的菜单命令，只能通过上述3种方式执行。

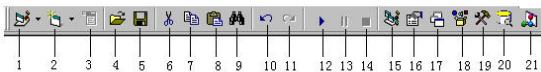
菜单打开后，在屏幕上显示相应的菜单命令。如果打开了不适当或不需要的菜单，或者执行菜单命令时打开了不需要的对话框，可以用Esc键关闭。

Visual Basic 应用程序的编辑、编译、连接、运行、调试及文件的打开、保存等都可以通过相应的菜单命令实现，其用法与上面介绍的类似。

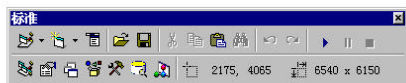
1.3.2 工具栏

Visual Basic 6.0提供了4种工具栏，包括编辑、标准、窗体编辑器和调试，并可根据需要定义用户自己的工具栏。在一般情况下，集成环境中只显示标准工具栏，其他工具栏可以通过“视图”菜单中的“工具栏”命令打开（或关闭）。每种工具栏都有固定和浮动两种形式。把鼠标光标移到固定形式工具栏中没有图标的地方，按住左按钮，向下拖动鼠标，或者双击工具栏左端的两条浅色竖线，即可把工具栏变为浮动的；双击浮动工具栏的标题条，则可将之变为固定工具栏。

固定形式的标准工具栏位于菜单栏的下面，即主窗口的底部，它以图标的形式提供了部分常用菜单命令的功能。只要用鼠标单击代表某个命令的图标按钮，就能直接执行相应的菜单命令。工具条中有21个图标，代表21种操作，如图1.4(a)所示。大多数图标都有与之等价的菜单命令。图1.4(b)是浮动形式的标准工具栏。



(a)



(b)

图1.4 标准工具栏

表1.1列出了工具栏中各图标的作用（表中的编号见图1.4(a)）。

表1.1 工具栏图标

编号	名称	作用
1	添加工程	添加一个新工程，相当于“文件”菜单中的“添加工程”命令
2	添加窗体	在工程中添加一个新窗体，相当于“工程”菜单中的“添加窗体”命令
3	菜单编辑器	打开菜单编辑对话框，相当于“工具”菜单中的“菜单编辑器”命令
4	打开工程	用来打开一个已经存在的Visual Basic工程文件，相当于“文件”菜单中的“打开工程”命令
5	保存工程（组）	保存当前的Visual Basic工程（组）文件，相当于“文件”菜单中的“保存工程（组）”命令

编 号	名 称	作 用
6	剪切	把选择的内容剪切到剪贴板, 相当于“编辑”菜单中的“剪切”命令
7	复制	把选择的内容复制到剪贴板, 相当于“编辑”菜单中的“复制”命令
8	粘贴	把剪贴板的内容复制到当前插入位置, 相当于“编辑”菜单中的“粘贴”命令
9	查找	打开“查找”对话框, 相当于“编辑”菜单中的“查找”命令
10	撤消	撤消当前的修改
11	重复	对“撤消”的反操作
12	启动	用来运行一个应用程序, 相当于“运行”菜单中的“启动”命令
13	中断	暂停正在运行的程序(可以用“启动”按钮或Shift+F5继续), 相当于热键Ctrl+Break或“运行”菜单中的“中断”命令
14	结束	结束一个应用程序的运行并回到设计窗口, 相当于“运行”菜单中的“结束”命令
15	工程资源管理器	打开工程资源管理器窗口, 相当于“视图”菜单中的“工程资源管理器”命令
16	属性窗口	打开属性窗口, 相当于“视图”菜单中的“属性窗口”命令
17	窗体布局窗口	打开窗体布局窗口, 相当于“视图”菜单中的“窗体布局窗口”命令
18	对象浏览器	打开“对象浏览器”对话框, 相当于“视图”菜单中的“对象浏览器”命令
19	工具箱	打开工具箱, 相当于“视图”菜单中的“工具箱”命令
20	数据视图	打开数据视图窗口
21	组件管理器	管理系统中的组件(Component)

在工具栏的右侧还有两个栏, 分别用来显示窗体的当前位置和大小, 其单位twip, 1英寸等于1440twip。左边一栏显示的是窗体左上角的坐标, 右边一栏显示的是窗体的长×宽。

twip是一种与屏幕分辨率无关的计量单位, 无论在什么屏幕上, 如果画了一条1440twip的直线, 打印出来都是1英寸。这种计量单位可以确保在不同的屏幕上都能保持正确的相对位置或比例关系。

在Visual Basic中, twip是缺省单位, 可以通过Scalemode属性改变。

除上面几个部分外, 在主窗口的左上角和右上角还有几个控制框, 其作用与Windows下的普通窗口相同。

1.4 其他窗口

标题栏、菜单栏和工具栏所在的窗口称为主窗口。除主窗口外, Visual Basic 6.0的编程环境中还有其他窗口, 包括窗体设计器窗口、属性窗口、工程管理器窗口、工具箱窗口、调色板窗口、代码窗口和立即窗口。本节介绍其中的部分窗口。

1.4.1 窗体设计器和工程资源管理器

1. 窗体设计器窗口

窗体设计器窗口简称窗体(Form), 是应用程序最终面向用户的窗口, 它对应于应用程序的运行结果。各种图形、图像、数据等都是通过窗体或窗体中的控件显示出来的。当打开一个新的工程文件时, Visual Basic建立一个空的窗体, 并命名为Formx(这里的x为1、2、3……)。

启动Visual Basic后, 窗体的名字为Form1, 其操作区中布满了小点(见图1.5), 这些小

点是供对齐用的。如果想清除这些点，或者想改变点与点之间的距离，执行“工具”菜单中的“选项”命令（“通用”选项卡）。

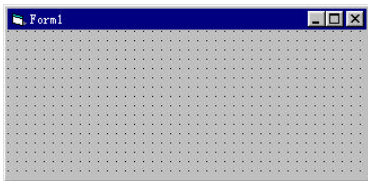


图1.5 窗体

在窗体的左上角是窗体的标题，右上角有3个图标，其作用与Windows下普通窗口中的图标相同。

在设计应用程序时，窗体就像是一块画布，在这块画布上可以画出组成应用程序的各个构件。程序员根据程序界面的要求，从工具箱（见后）中选择需要的工具，并在窗体中画出来，这样就完成了应用程序设计的第一步。

2. 工程资源管理器窗口

在工程资源管理器窗口中，含有建立一个应用程序所需要的文件的清单。工程资源管理器窗口中的文件可以分为6类，即窗体文件（.frm）、程序模块文件（.bas）、类模块文件（.cls）、工程文件（.vbp）、工程组文件（.vbproj）和资源文件（.res）。图1.6所示为含有两个工程、多个窗体、多个程序模块和类模块的工程资源管理器窗口。



图1.6 工程资源管理器窗口（展开）

在工程资源管理器窗口中，括号内是工程、窗体、程序模块、类模块等的存盘文件名，括号外是相应的名字（即Name属性）。每个工程名左侧都有一个方框，当方框内为“-”号时，该工程处于“展开”状态（见图1.6），此时如果单击“-”号方框，则变为“折叠”状

态，方框内的“-”号变为“+”号，如图1.7所示。

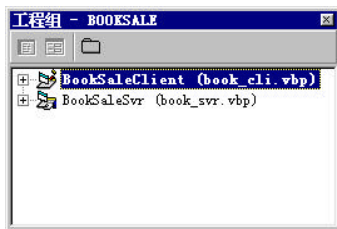


图1.7 工程资源管理器窗口(折叠)

在工程资源管理器窗口中有以下几类文件：

(1) 工程文件和工程组文件

工程文件的扩展名为.VBP，每个工程对应一个工程文件。当一个程序包括两个以上的工程时，这些工程构成一个工程组，工程组文件的扩展名为.VBG。用“文件”菜单中的“新建工程”命令可以建立一个新工程，用“打开工程”命令可以打开一个已有的工程，用“添加工程”命令可以添加一个工程。

(2) 窗体文件

窗体文件的扩展名为.FRM，每个窗体对应一个窗体文件，窗体及其控件的属性和其他信息（包括代码）都存放在该窗体文件中。一个应用程序可以有多个窗体（最多可达255个），因此可以有多个以.FRM为扩展名的窗体文件。

执行“工程”菜单中的“添加窗体”命令或单击工具栏中的“添加窗体”按钮可以增加一个窗体，执行“工程”菜单中的“删除”命令可以删除当前的窗体。每建立一个窗体，工程资源管理器窗口中就增加一个窗体文件，每个窗体都有不同的名字，可以通过属性窗口设置（Name属性），其缺省名字为FormX（X为1、2、3...），相应的缺省文件名为FormX.frm（X为1、2、3...）。

(3) 标准模块文件

标准模块文件也称为程序模块文件，其扩展名为.BAS，它是为合理组织程序而设计的。标准模块是一个纯代码性质的文件，它不属于任何一个窗体，主要在大型应用程序中使用。

标准模块由程序代码组成，主要用来声明全局变量和定义一些通用的过程，可以被不同窗体的程序调用。标准模块通过“工程”菜单中的“添加模块”命令来建立。

(4) 类模块文件

Visual Basic 提供了大量预定义的类，同时也允许用户根据需要定义自己的类。用户通过类模块来定义自己的类，每个类都用一个文件保存，其扩展名为.CLS。

(5) 资源文件

资源文件中存放的是各种“资源”，是一种可以同时存放文本、图片、声音等多种资源的文件。资源文件由一系列独立的字符串、位图及声音文件（.wav、.mid）组成，其扩展名为.res。资源文件是一个纯文本文件，可以用简单的文字编辑器（如NotePad）编辑。

除上述几类文件外，在工程资源管理器窗口中还有3个按钮，分别为“查看代码”、“查看

对象”和“切换文件夹”。如果单击工程管理器窗口中的“查看代码”按钮，相应文件的代码将在代码窗口中显示出来。当单击“查看对象”按钮时，Visual Basic将显示相应的窗体。在一般情况下，工程管理器窗口中的项目不显示文件夹，单击“切换文件夹”按钮，可显示各类文件所在的文件夹，如图1.8所示。如果再单击一次该按钮，则取消文件夹显示。

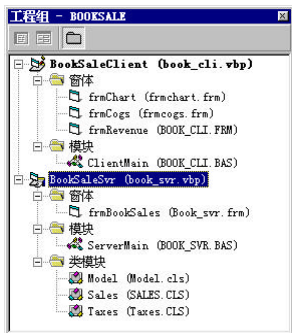


图1.8 用“切换文件夹”按钮显示文件夹

用Visual Basic设计应用程序时，通常先设计窗体（即界面），然后编写程序。设计完窗体后，只要双击窗体的任一部位，就可以切换到代码窗口，与单击“查看代码”按钮的作用相同。

1.4.2 属性窗口和工具箱窗口

1. 属性窗口

属性窗口主要是针对窗体和控件设置的，在Visual Basic中，窗体和控件称为对象。每个对象都可以用一组属性来刻画其特征，而属性窗口就是用来设置窗体或窗体中的控件的属性的。

图1.9示出了属性窗口的一部分。窗口中的属性按字母顺序排列，可以通过窗口右部的垂直滚动条找到任一个属性。除窗口标题外，属性窗口分为4个部分，分别为对象框、属性显示方式、属性列表和对当前属性的简单解释。

对象框位于属性窗口的顶部，可以通过单击其右端向下的箭头下拉显示列表，其内容为应用程序中每个对象的名字及对象的类型。启动Visual Basic后，对象框中只含有窗体的信息。随着窗体中控件的增加，将把这些对象的有关信息加入到对象框的下拉列表中。

属性显示方式分为两种，即按字母顺序和按分类顺序，分别通过单击相应的按钮来实现。图1.9是按字母顺序显示的属性列表，如果单击“按分类序”按钮，则按分类顺序显示属性列表，如图1.10所示。

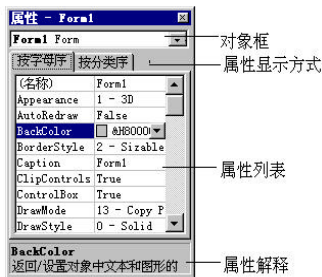


图1.9 属性窗口（按字母序）



图1.10 属性窗口（按分类序）

在属性列表部分，可以滚动显示当前活动对象的所有属性，以便观察或设置每项属性的当前值。属性的变化将改变相应对象的特征。

每选择一种属性（条形光标位于该属性上），在“属性解释”部分都要显示该属性名称和功能说明。

每个Visual Basic对象都有其特定的属性，可以通过属性窗口设置，对象的外观和对应的操作由所设置的值来确定。有些属性的取值是有一定限制的，例如对象的可见性只能设置为True或False（即可见或不可见），而有些属性（如标题）可以为任何文本。在实际的应用程序设计中，不可能也没必要设置每个对象的所有属性，很多属性可以使用缺省值。

只有在设计阶段才能激活属性窗口。可以用下述方法激活属性窗口：

- (1) 用鼠标单击属性窗口的任何部位。
- (2) 执行“视图”菜单中的“属性窗口”命令。
- (3) 按F4键。
- (4) 单击工具栏上的“属性窗口”按钮。

2. 工具箱窗口

工具箱窗口由工具图标组成，这些图标是Visual Basic应用程序的构件，称为图形对象或控件（Control），每个控件由工具箱中的一个工具图标表示。

在一般情况下，工具箱位于窗体的左侧。工具箱中的工具分为两类，一类称为内部控件或标准控件，一类称为ActiveX控件。启动Visual Basic后，工具箱中只有内部控件。

工具箱主要用于应用程序的界面设计。在设计阶段，首先用工具箱中的工具（即控件）在窗体上建立用户界面，然后编写程序代码。界面的设计完全通过控件来实现，可以任意改变其大小，移动到窗体的任何位置。

除上述几种窗口外，在集成环境中还有其他窗口，包括窗体布局窗口、代码编辑器窗口、立即窗口、本地窗口和监视窗口等。