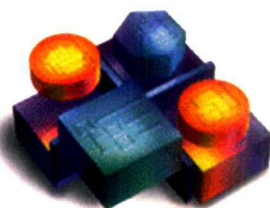




中文 Visual Basic 6.0



赵增敏 朱粹丹 赵朱曦 编著

快速入门与提高

河南科学技术出版社



可视化编程丛书

中文 Visual Basic 6.0 快速入门与提高

赵增敏 朱粹丹 赵朱曦 编著

河南科学技术出版社

内 容 提 要

问世于 1991 年的 Visual Basic 是一种 Windows 应用程序开发工具, 它以其好学易用、功能强大等特点而备受广大程序员的青睐。本书详细地介绍了 Visual Basic 6.0 中文版的开发环境、基本语言、设计方法和编程技巧, 主要内容包括: Visual Basic 使用基础、Visual Basic 编程语言、窗体设计与内部控件使用、ActiveX 控件的使用与制作、菜单与工具栏的设计、数据库程序的设计、动态链接库的使用、多媒体程序的设计、帮助文件的制作。本书语言流畅、内容翔实、条理清晰, 并配备了大量的编程实例, 这些实例中所用到的编程技巧可以直接移植到应用程序中。

本书适用于想用 Visual Basic 开发 Windows 应用程序的初学者, 也适用于具有一定编程经验的专业开发人员。

图书在版编目 (CIP) 数据

中文 Visual Basic 6.0 快速入门与提高/赵增敏等编著. —郑州: 河南科学技术出版社, 1999.8

(可视化编程丛书)

ISBN 7-5349-2389-1

I. 中… II. 赵… III. Basic 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (1999) 第 25889 号

责任编辑 王广照

河南科学技术出版社出版

郑州市农业路 73 号

邮政编码: 450002 电话: (0371) 5721186

河南第二新华印刷厂印刷

全国新华书店发行

开本: 787×1092 1/16 印张: 22 字数: 490 千字

1999 年 8 月第 1 版 1999 年 8 月第 1 次印刷

印数: 1—4 000

ISBN 7-5349-2389-1 / T · 487 定价: 29.00 元

前 言

Microsoft Windows 以其新颖美观的图形用户界面和简单易学的操作方式赢得了广大计算机用户的喜爱和认可，它的推出和不断完善使计算机的操作方式和软件开发过程都发生了革命性的变化。开发基于 Windows 平台的应用程序，已经成为当今计算机软件开发的主流，有不少过去致力于 DOS 应用程序开发的专业程序员和业余爱好者现在也纷纷加入了开发 Windows 应用程序的行列。但是，要设计出一个界面友好、功能强大的 Windows 应用程序，对于编程人员来说却是一个很大的挑战。这是因为编写一个 Windows 应用程序必须包括创建窗口、菜单、图标、字体、对话框以及其他界面元素等项内容，整个设计过程要比传统的 DOS 程序的设计过程麻烦得多，即使最简单的程序也不例外。例如，用 C/C++ 语言配合 SDK 设计一个 Windows 应用程序时，仅仅是在屏幕上开一个窗口，就需要编写 80 多行源程序代码。如果要在界面上添加其他部件并赋予程序各种各样的操作功能，其复杂程度就可想而知了。

Visual Basic 正是在这种背景之下应运而生的，它给 Windows 软件开发者和编程爱好者带来了福音。微软公司总裁比尔·盖茨称它是“用 Basic 语言开发 Windows 应用程序的最强有力的工具”。苹果公司创始人史迪夫·吉布森也发出感叹，称它是“一个令人震惊的奇迹”。而众多的专业程序员和编程爱好者更是对 Visual Basic 表现出了极大的兴趣。目前世界上已经有 300 多万名 Visual Basic 程序员，正如微软公司 Visual Basic 项目经理所说：“哪里有计算机产业，哪里就有 Visual Basic。”

Visual Basic 之所以受到广大程序员的青睐，原因在于它不仅继承了传统 Basic 语言简单易学的优点，同时又提供了许多全新的可视化设计工具，利用这些工具在窗体上绘制各种控件就可以完成用户界面的设计任务，这就像在画图程序中绘画一样简单。在 Visual Basic 中，窗体和控件都是对象，为了让这些对象在程序运行时响应用户操作，我们只要针对它们编写一些短小的称为事件过程的程序就行了。可视化设计、面向对象和事件驱动是 Visual Basic 的基本特色。用 Visual Basic 可以十分轻松地设计出各种不同用途的 Windows 应用程序。例如，利用 Visual Basic 提供的数据库控件，只需编写少量代码或根本不用编写代码就可以完成一个数据库应用程序的设计任务。利用 Visual Basic 提供的 ActiveMovie 控件，连一行语句都不用编写，就能够设计出一个多媒体播放程序，用该程序不仅可以播放波形文件和 MIDI 音乐文件，也可以播放 CD 音乐和动画文件，甚至还可以播放 VCD 影碟。

本书分 9 章来全面地介绍 Visual Basic 6.0 中文版的开发环境、基本语言、设计方法和编程技巧。全书可以划分为两大部分：

前 5 章讲述 Visual Basic 6.0 的基本内容：第 1 章介绍了使用 Visual Basic 所需要的一些基础知识；第 2 章阐述了 Visual Basic 编程语言的主要内容；第 3 章讨论了窗体的设计和内部控件的使用；第 4 章讲述了如何使用和制作 ActiveX 控件；第 5 章讲述菜单和工具栏的设计方法。

后 4 章则属于专题性的内容，讲述一些高级编程技巧：第 6 章详细地介绍了数据库应用程序的设计，不仅讲述了数据控件和数据绑定控件在数据库访问中的应用，也讲述了 Visual

Basic 6.0 中新增的 ADO 数据控件和一些 ActiveX 数据绑定控件的使用方法，此外还讲述了数据报表的制作方法；第 7 章介绍如何使用动态链接库来完成 Visual Basic 无法实现的功能，从而使所开发的应用程序不同凡响；第 8 章介绍多媒体编程技巧，内容包括制作文字的特殊效果、处理图像和播放多媒体文件；第 9 章介绍如何制作 Windows 95 风格的帮助文件，并把它添加到 Visual Basic 应用程序中。

由于作者学识有限，加上成书时间仓促，书中疏漏和错误之处，欢迎广大读者提出宝贵意见。

作 者

1999 年 3 月

目 录

1 Visual Basic 使用基础	(1)	2.1.5 对象引用型 (Object)	(38)
1.1 用 Visual Basic 编写第一个程序 ..	(1)	2.2 常量	(38)
1.1.1 启动 Visual Basic	(1)	2.2.1 普通常量	(38)
1.1.2 设计用户界面	(2)	2.2.2 符号常量	(39)
1.1.3 编写程序代码	(5)	2.3 变量	(41)
1.1.4 运行程序	(6)	2.3.1 变量命名规则	(41)
1.1.5 保存文件	(7)	2.3.2 变量类型声明	(42)
1.1.6 生成 EXE 文件	(8)	2.3.3 变量的作用域	(44)
1.1.7 退出 Visual Basic	(9)	2.3.4 用户自定义类型	(47)
1.2 Visual Basic 的编程机制	(10)	2.3.5 枚举类型	(49)
1.2.1 对象与类的概念	(10)	2.4 运算符与表达式	(50)
1.2.2 对象的属性、方法和事件	(11)	2.4.1 算术运算符与算术表达式	(50)
1.2.3 Visual Basic 的编程机制	(14)	2.4.2 比较运算符与关系表达式	(51)
1.2.4 Visual Basic 工程的结构	(14)	2.4.3 连接运算符与字符表达式	(53)
1.3 Visual Basic 集成开发环境	(18)	2.4.4 逻辑运算符与逻辑表达式	(54)
1.3.1 工作窗口的组织	(18)	2.4.5 运算符的优先级别	(55)
1.3.2 主窗口	(18)	2.5 基本语句	(55)
1.3.3 对象窗口	(21)	2.5.1 注释语句	(56)
1.3.4 工具箱窗口	(22)	2.5.2 赋值语句	(56)
1.3.5 属性窗口	(23)	2.5.3 Set 语句	(57)
1.3.6 工程资源管理器窗口	(25)	2.5.4 多重语句与语句续行	(59)
1.3.7 窗体布局窗口	(26)	2.5.5 结束语句	(59)
1.3.8 代码窗口	(26)	2.6 程序控制结构	(61)
1.4 Visual Basic 工程管理	(29)	2.6.1 判定结构	(61)
1.4.1 工程的基本操作	(29)	2.6.2 循环结构	(65)
1.4.2 改变工程的成员	(31)	2.7 数组	(69)
1.4.3 设置工程的属性	(32)	2.7.1 定长数组	(70)
2 Visual Basic 编程语言	(35)	2.7.2 动态数组	(71)
2.1 数据类型概述	(35)	2.7.3 控件数组	(73)
2.1.1 数值型数据类型	(36)	2.8 过程与函数	(76)
2.1.2 布尔型 (Boolean)	(37)	2.8.1 子过程	(76)
2.1.3 日期型 (Date)	(37)	2.8.2 函数	(78)
2.1.4 字符串 (String)	(37)	2.8.3 参数传递机制	(87)



3 窗体与控件	(91)	4.1.2 使用 RichTextBox 控件	(158)
3.1 窗体设计.....	(91)	4.1.3 使用 UpDown 控件	(162)
3.1.1 设置窗体的属性	(91)	4.1.4 使用 MonthView 控件 ...	(164)
3.1.2 调用窗体的方法	(93)	4.2 制作 ActiveX 控件	(169)
3.1.3 响应窗体的事件	(98)	4.2.1 制作 ActiveX 控件的基本	
3.1.4 设计多文档界面	(105)	知识	(169)
3.1.5 启动应用程序	(107)	4.2.2 制作三维立体字控件 ...	(170)
3.1.6 结束应用程序	(109)	4.2.3 制作可滚动图片框控件	(179)
3.2 控件的基本操作	(110)	5 菜单与工具栏	(185)
3.2.1 利用工具箱绘制控件	(110)	5.1 菜单设计	(185)
3.2.2 利用鼠标或键盘调整控件	(111)	5.1.1 菜单概述	(185)
3.2.3 利用菜单命令复制和删除		5.1.2 用菜单编辑器创建菜单	(187)
控件	(111)	5.1.3 菜单的命名约定	(189)
3.2.4 利用窗体编辑器设置控件		5.1.4 创建子菜单	(190)
的格式	(112)	5.1.5 创建菜单控件数组	(191)
3.2.5 控件的默认属性和命名前		5.1.6 运行时修改菜单	(193)
缀	(113)	5.1.7 编写菜单控件的代码 ...	(196)
3.3 使用内部控件	(114)	5.1.8 设计弹出式菜单	(196)
3.3.1 命令按钮控件(CommandButton)		5.1.9 MDI 应用程序中的菜单	(199)
.....	(114)	5.2 工具栏设计	(203)
3.3.2 标签控件 (Label)	(116)	5.2.1 创建工具栏的主要步骤	(203)
3.3.3 文本框控件 (TextBox)	(117)	5.2.2 把 ToolBar 控件添加到工具	
3.3.4 滚动条控件 (HScrollBar 和		箱中	(204)
VscrollBar)	(125)	5.2.3 在 ImageList 控件中添加图	
3.3.5 选项按钮控件 (OptionButton)		像	(204)
.....	(127)	5.2.4 在 ImageList 控件与 ToolBar	
3.3.6 复选框控件 (CheckBox)	(129)	控件之间建立关联	(207)
3.3.7 列表框控件 (ListBox) ...	(131)	5.2.5 为工具栏添加按钮	(208)
3.3.8 组合框控件 (ComboBox)		5.2.6 为工具栏添加按钮菜单	(213)
.....	(137)	5.2.7 为工具栏编写事件过程	(215)
3.3.9 图片框控件 (PictureBox)		6 数据库程序设计	(217)
和图像控件 (Image) ...	(138)	6.1 使用数据控件	(217)
3.3.10 直线控件 (Line) 和形状		6.1.1 关系型数据库概述	(217)
控件 (Shape)	(140)	6.1.2 使用数据控件	(219)
3.3.11 定时器控件 (Timer) ...	(143)	6.2 使用 ActiveX 数据绑定控件 ...	(228)
3.3.12 文件管理控件	(145)	6.2.1 使用数据绑定列表框和	
4 ActiveX 控件	(151)	数据绑定组合框	(228)
4.1 使用 ActiveX 控件	(151)	6.2.2 使用 ADO 数据控件和	
4.1.1 使用 CommonDialog 控件	(151)	数据网格控件	(231)



6.3 利用代码操作记录	(240)	8.3.1 用 Animation 控件播放无声 动画	(307)
6.3.1 记录集的定位	(240)	8.3.2 用 Multimedia 控件播放 多媒体文件	(309)
6.3.2 查找记录	(245)	8.3.3 用 ActiveMovie 控件制作 多媒体浏览器	(315)
6.3.3 用代码更新记录集	(247)	9 帮助文件制作	(317)
6.3.4 控制事务	(249)	9.1 帮助文件的制作流程	(317)
6.3.5 确认数据库的变化	(251)	9.2 建立主题文件	(317)
6.3.6 数据访问专用的 UpDate 方 法	(252)	9.2.1 建立和保存主题文件 ...	(318)
6.4 数据报表设计	(253)	9.2.2 在帮助主题中添加各种注 脚	(318)
6.4.1 数据报表设计器概述	(253)	9.2.3 在帮助主题中添加不滚动 区域	(321)
6.4.2 创建一个简单的数据报表	(255)	9.2.4 与帮助主题建立链接 ...	(322)
6.4.3 在数据报表中计算	(259)	9.2.5 在帮助主题中添加位图	(323)
6.4.4 向数据报表添加日期、时 间、页号和标题	(262)	9.2.6 为帮助主题创建超图形	(323)
7 使用动态链接库	(263)	9.2.7 在帮助主题中添加动画	(325)
7.1 DLL 过程的声明和调用	(263)	9.3 建立目录文件	(325)
7.1.1 使用 DLL 过程示例	(263)	9.3.1 建立目录文件的步骤 ...	(326)
7.1.2 声明 DLL 过程	(265)	9.3.2 设置缺省的帮助主题窗口	(327)
7.1.3 访问 Windows API	(270)	9.3.3 为目录选项卡添加标题	(327)
7.2 向 DLL 过程传递参数	(272)	9.3.4 为目录选项卡添加主题	(328)
7.2.1 向 DLL 过程传递字符串	(272)	9.3.5 改变标题或主题的级别	(329)
7.2.2 向 DLL 过程传递数组 ...	(274)	9.4 建立帮助工程文件	(329)
7.2.3 向 DLL 过程传递自定义 数据	(276)	9.4.1 创建帮助工程文件	(330)
7.2.4 向 DLL 过程传递其他信息	(278)	9.4.2 指定主题文件	(330)
7.3 Windows API 函数应用示例	(280)	9.4.3 映射帮助主题	(331)
7.3.1 使窗口总是位于最前面 ...	(281)	9.4.4 指定目录文件	(333)
7.3.2 判断鼠标指针移出控件 ...	(283)	9.4.5 指定位图文件的位置 ...	(333)
7.3.3 拖动无标题栏窗体	(284)	9.4.6 定制帮助窗口	(334)
8 多媒体程序设计	(287)	9.4.7 保存帮助工程文件	(337)
8.1 文字的特殊效果	(287)	9.5 编译与测试帮助文件	(337)
8.1.1 文字的阴影效果	(287)	9.5.1 编译帮助工程文件	(337)
8.1.2 文字的浮雕效果	(288)	9.5.2 测试帮助文件	(338)
8.1.3 文字的渐变效果	(290)	9.6 为应用程序添加帮助	(338)
8.2 图像的处理	(292)	9.6.1 为应用程序指定帮助文件	(338)
8.2.1 PaintPicture 方法	(293)	9.6.2 为界面元素设置帮助主题	(339)
8.2.2 图像的缩放、翻转和拉伸	(294)		
8.2.3 图像的视觉过渡	(299)		
8.3 多媒体文件的播放	(307)		

1 Visual Basic 使用基础

随着 Windows 操作系统的日益普及，开发符合 Windows 规范的应用程序已经成为计算机软件行业的一种时尚。Visual Basic 就是 Windows 应用程序的开发工具之一，其 6.0 版是作为 Visual Studio 6.0 的一个成员而出现的，它和 Visual C++、Visual FoxPro 等开发工具一道构成了微软公司的可视化编程语言家族。本章将结合一个具体的程序范例来介绍 Visual Basic 的编程步骤，阐述 Visual Basic 中的一些基本概念，并对 Visual Basic 的集成开发环境作一个简单的描述。以上这些内容都是用 Visual Basic 编程时应当掌握的基础知识。

1.1 用 Visual Basic 编写第一个程序

我们使用 Visual Basic 6.0 来编写第一个 Windows 风格的应用程序，该程序运行时首先在桌面上打开一个窗口，窗口中包含一个命令按钮，当用鼠标单击这个命令按钮时，会在窗口中显示出一句欢迎词——“欢迎您用 Visual Basic 6.0 编程！”

1.1.1 启动 Visual Basic

在任务栏左端单击开始按钮 ，然后依次指向程序、Microsoft Visual Studio 6.0，再单击 Microsoft Visual Basic 6.0，如图 1.1 所示，即可启动 Visual Basic。

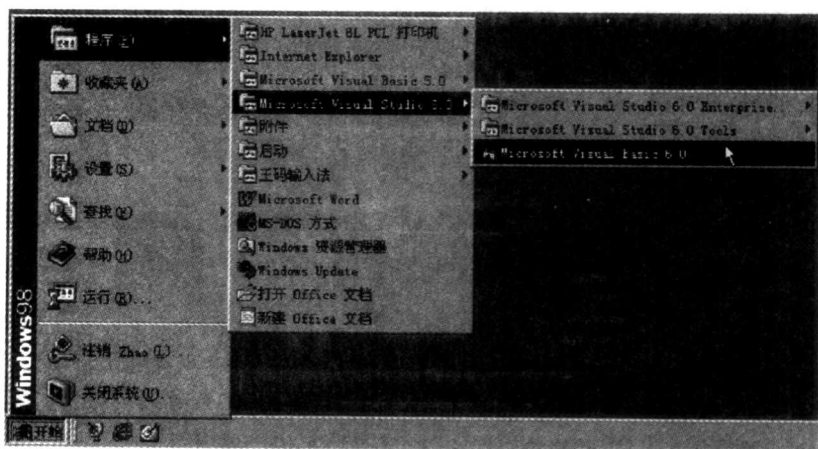


图 1.1 用开始菜单启动 Visual Basic

启动 Visual Basic 后，会出现一个漂亮的 Splash 窗口，其中显示出软件产品的徽标、版本和版权信息以及用户资料等，当这个窗口消失后，便出现了如图 1.2 所示的新建工程对话框，用来提示新建工程的类型，默认的工程类型是标准 EXE，新建工程对话框包含新建、



现存和最新 3 个选项卡。新建选项卡给出新建工程的类型，现存选项卡用于打开一个已经存在的工程，利用最新选项卡则可以打开最近修改过的工程。

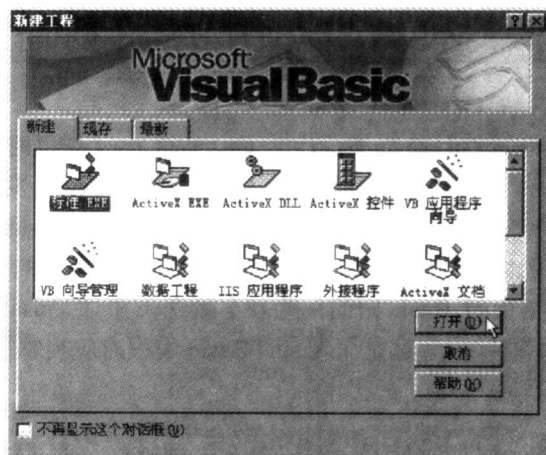


图 1.2 新建工程对话框

1.1.2 设计用户界面

启动 Visual Basic 以后，我们将在屏幕上看到如图 1.3 所示的集成开发环境（IDE），它包含若干个工作窗口，其中有一些窗口是连接在一起的。我们看到，主窗口位于最上方，

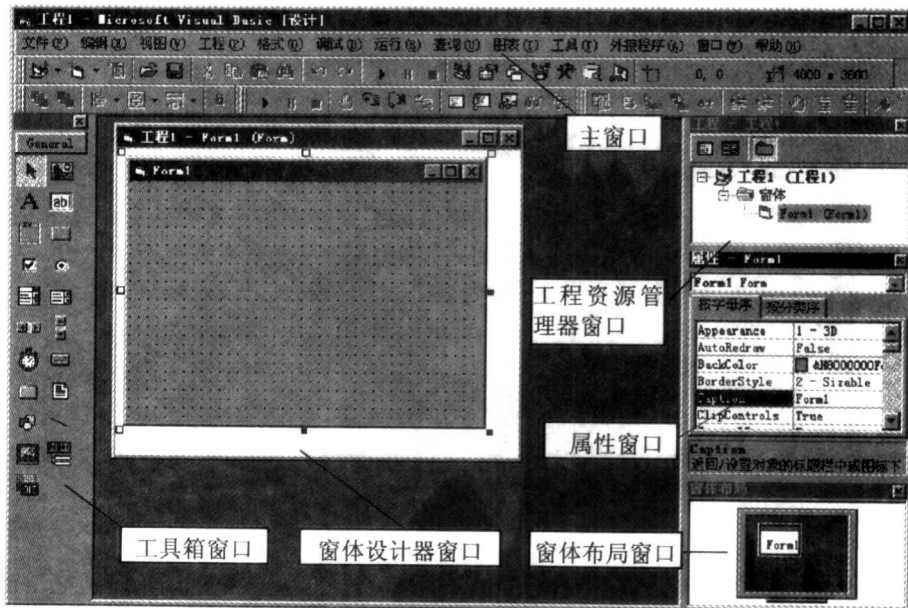


图 1.3 Visual Basic 集成开发环境



它由标题栏、菜单栏和工具栏所组成。主窗口是整个集成开发环境的控制中心，关闭该窗口，就意味着退出 Visual Basic。

我们看到，主窗口的标题文字是“工程 1 - Microsoft Visual Basic [设计]”，由此我们可以得到两个信息：一个是新建的工程默认名称——工程 1，另一个是当前所处的工作模式——设计，即现在正处在设计模式下。在设计模式下，既可以设计程序的图形用户界面，也可以编写实现程序功能所需的代码。下面，我们首先来设计程序的图形用户界面。

1.1.2.1 窗体对象

以窗口形式运行是 Windows 应用程序的显著特征之一，设计用户界面通常就是从建立窗口开始的。在 Visual Basic 中新建一个标准 EXE 工程时，系统自动为我们准备了一个现成的窗口，这便是位于对象窗口中的那个标题为 Form1 的窗口。不过，用 Visual Basic 的术语来说，处在设计模式时，应当把这个窗口称为窗体 (Form)。

1.1.2.2 在窗体上添加控件

a. 在窗体上添加命令按钮。在工具箱中单击 CommandButton 按钮，此时该按钮将凹陷下去，呈现状，表明它已被选中，然后把鼠标指针移到窗体上，鼠标指针将变成十字形状，在将要放置命令按钮左上角的位置上按下鼠标左键，并向右下角拖动，拖出一个灰色的方框，该方框表示按钮的轮廓，其右下角出现一个黄色的文字框，表示命令按钮当前的宽度和高度，当大小合乎需要时松开鼠标左键，一个命令按钮便画好了，如图 1.4 所示。

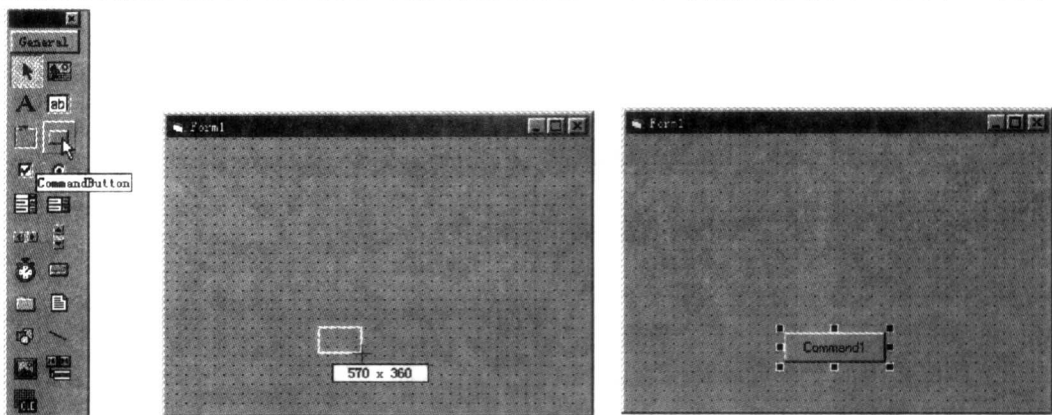


图 1.4 在窗体上绘制一个命令按钮

b. 在窗体上画一个标签。在工具箱中单击 Label 按钮，以选定该按钮，然后把鼠标指针移到窗体上，此时鼠标指针将自行变成十字形状，在将要放置标签左上角的位置上按下鼠标左键，接着向右下方拖动鼠标，从而画出一个适当大小的标签，如图 1.5 所示。

1.1.2.3 设置窗体和控件的属性

a. 修改窗体的标题文字。在属性窗口上方的对象选择框中选取 Form1，然后双击 Caption 属性名，使其右侧的属性值 Form1 变成高亮显示，表明已被选定，此时键入新标题——“用 Visual Basic 设计的第一个程序”，如图 1.6 所示。

b. 把命令按钮上显示的文字改成“欢迎”。在属性窗口上方的对象选择框中选取 Command1，然后双击 Caption 属性名，使其当前值 Command1 进入选定状态，接着切换到

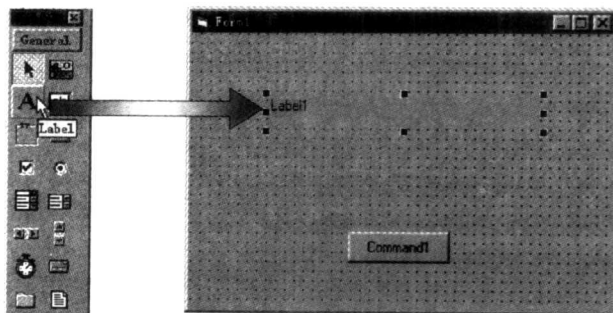


图 1.5 在窗体上绘制一个标签控件

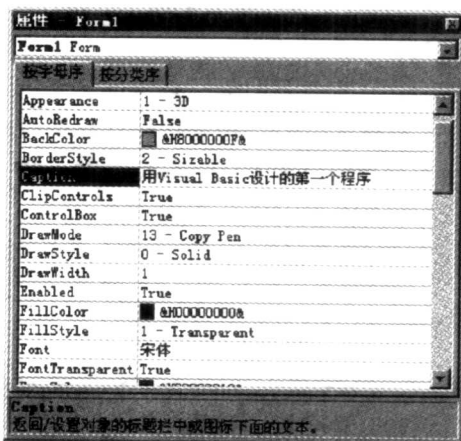


图 1.6 在属性窗口中修改窗体的标题文字

中文输入法,并键入新的标题文字——“欢迎”,命令按钮表面上的文字随之更新。

c. 把标签中的文字删除掉。在属性窗口上方的对象选择框中选取 Label1,然后双击 Caption 属性名,使其当前值 Label1 进入选定状态,接着按下 DELETE 键,删除该标签中的文字内容,使之变成一个空白标签。

d. 设置字体大小。在标签中和命令按钮表面上显示的文字,默认的字体为小五号,看起来不够醒目,应当把字体调大一些。为此,我们单击工具箱左上角的指针按钮,然后把鼠标指针移到窗体上,

此时鼠标指针变成十字形状,拖出一个虚线选择框把标签和命令按钮都包围起来,在属性窗口中双击 Font 属性名,打开字体对话框,把字体大小设置为五号,如图 1.7 所示。

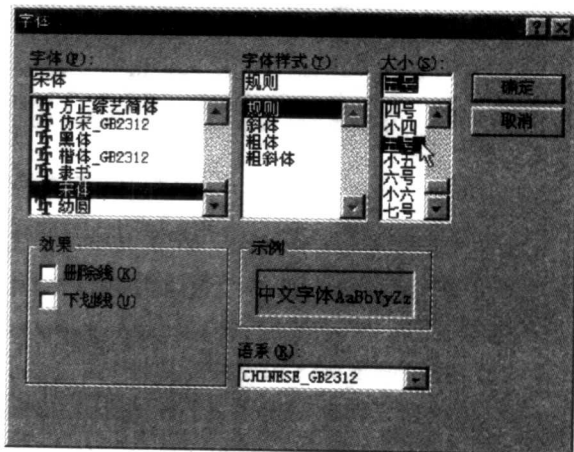


图 1.7 设置字体大小



e. 为标签添加边框。在属性窗口上方的对象选择框中选取 Label1，然后单击 BorderStyle 属性名，此时看到其右侧出现一个下拉式列表，当前值为 0 - None，即标签没有边框。要在标签四周加上边框，可以从列表中选择 1 - Fixed Single，如图 1.8 所示。

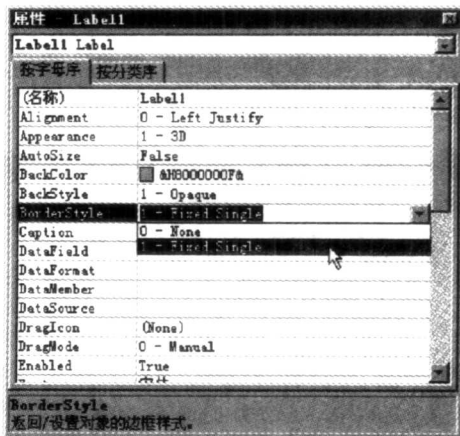


图 1.8 为标签加上边框

以上我们绘制了一个命令按钮和一个标签，并对窗体、命令按钮和标签的一些默认设置进行了修改。至此，范例程序的用户界面已经设计好了，其效果如图 1.9 所示。

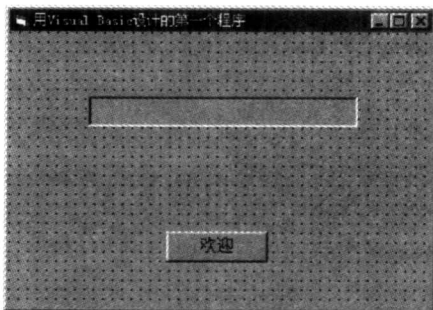


图 1.9 范例程序的用户界面

1.1.3 编写程序代码

在 1.1.2 中，通过在窗体上添加一个命令按钮和一个标签，已经完成了程序用户界面的设计工作。要实现程序的功能，即在单击命令按钮时用标签显示一行文字，还必须打开代码窗口，并在这个窗口中编写实现程序功能的 Visual Basic 程序语句。我们在窗体上双击命令按钮，以打开代码窗口，这时可以看到此窗口中已经出现了以下两条语句：

```
Private Sub Command1_Click()  
End Sub
```

其中 Command1 是命令按钮的名称，Click 是该按钮的一个事件名称。以上语句的作用是定义一个叫做 Command1_Click 的过程，在程序运行时单击命令按钮，将触发该按钮的 Click 事件，从而执行这个过程中包含的语句（过程体），这个过程称为命令按钮的一个事件过程。



在默认情况下，过程体中不包含任何语句，若在程序运行时单击命令按钮，程序是不会作出反应的。为了赋予程序预定的功能，我们在事件过程中填写以下两行语句：

```
Label1.Caption = "欢迎您用 Visual Basic 6.0 编程！"
```

```
Label1.Refresh
```

在这里，第 1 行语句的功能是把等号右边的字符串作为标签中显示的内容，第 2 行语句的功能是刷新显示标签中的内容。此时的代码窗口如图 1.10 所示。

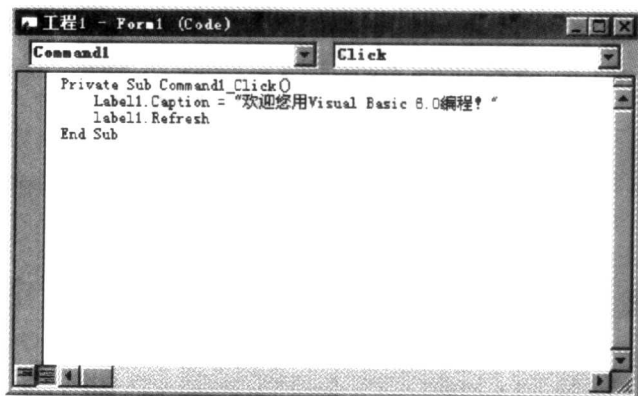


图 1.10 代码窗口

1.1.4 运行程序

要在 Visual Basic 集成开发环境中运行所设计的应用程序，可以选择下列操作之一：

- 从运行菜单中选择启动命令。
- 按快捷键 F5。
- 单击工具栏上的启动按钮

在程序运行期间，对象窗口、工具箱窗口、属性窗口、代码窗口以及窗体布局窗口一概不见了，主窗口标题栏中的“设计”字样则被“运行”所取代，表明现在已经由设计模式切换到运行模式，设计模式下的窗体窗口在程序运行期间变成了标准的 Windows 应用程序窗口。

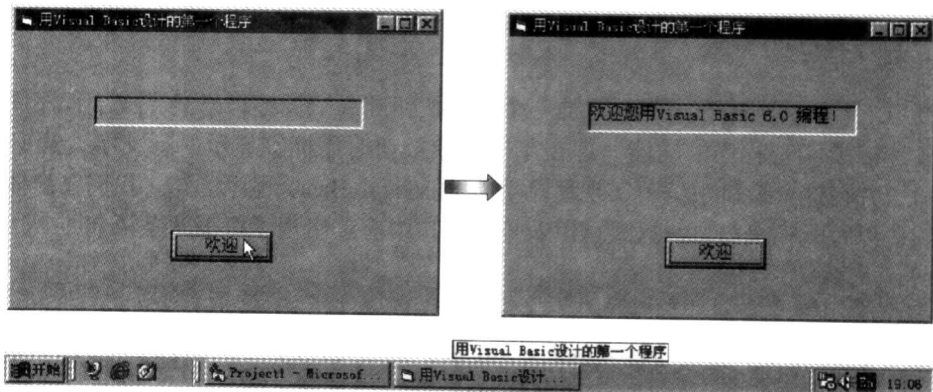


图 1.11 范例程序的运行结果



序窗口，而且在 Windows 任务栏上出现了相应的按钮。假如用鼠标单击窗口中的“欢迎”按钮，或者按下 ENTER 键，就会在标签中显示一句欢迎词——“欢迎您用 Visual Basic 6.0 编程！”，如图 1.11 所示。

要结束程序的运行，从运行模式切换到设计模式，可以选择下列操作之一：

- 从运行菜单中选择结束命令。
- 在工具栏单击结束按钮.
- 单击应用程序窗口右上角的关闭按钮.
- 双击应用程序窗口左上角的控制菜单框.
- 按快捷键 ALT + F4。

1.1.5 保存文件

下面介绍如何把所设计的范例程序保存到磁盘上。当第一次存盘时，会出现相应的对话框，在这些对话框中可以指定文件的存放路径，并提供有关的文件名。以后再存盘时，除非要更改文件的存放路径或文件名，通常是按照第一次存盘时所作的设置加以保存，因而不出现对话框。在程序开发过程中经常保存文件是一个好习惯，这样会避免因停电或死机而造成数据丢失。

保存文件时，从文件菜单中选择保存工程命令，或在工具栏上单击保存工程按钮, 这时将出现如图 1.12 所示的文件另存为对话框。

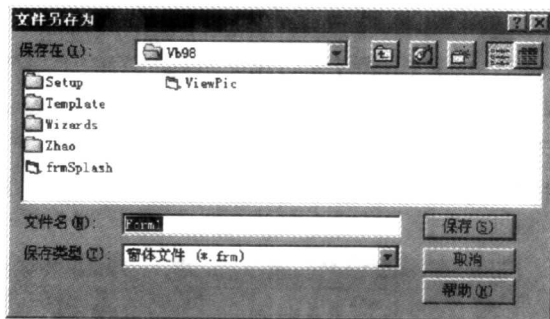


图 1.12 文件另存为对话框

单击保存在下拉式列表框右端的向下箭头, 并在打开的列表框中选择一个驱动器图标，使这个驱动器上的目录显示出来；然后双击一个用于保存文件的目录，使之出现在保存在方框内；接着在文件名文本框中输入文件名 frmHello，以取代 Visual Basic 暂时命名的 Form1，文件扩展名则是由系统自动提供的.frm；最后单击保存按钮，在磁盘建立一个名为 frmHello.frm 的文件，这个文件称为窗体文件，它用于存储窗体及其所包含对象的属性设置和相关的程序代码。

保存文件时，最好选择一个专门的目录来保存自己建立的文件，而不要把这些文件与 Visual Basic 的系统文件混放在一起。要建立新的目录，可以利用 Windows 操作系统的标准功能来完成。如果已经打开了如图 1.12 所示的对话框，则可以单击新建目录按钮来建立一个目录。



保存窗体文件以后,接着会出现如图 1.13 所示的工程另存为对话框。



图 1.13 工程另存为对话框

在文件名文本框中输入 Hello,以取代 Visual Basic 命名的“工程 1”,文件扩展名则是由系统自动提供的.vbp,单击保存按钮,在磁盘上建立一个名为 Hello.vbp 的文件,这个文件称为工程文件,它用于保存程序设计过程所用到的各个模块的信息。

保存文件时,一般应将一个应用程序的工程文件和窗体文件存放在同一个目录中,但也可以存放在不同的目录中,工程文件中记录着应用程序中所有模块文件存放的路径。

1.1.6 生成 EXE 文件

在 1.1.4 中,我们已经在 Visual Basic 的集成开发环境中运行了第一个范例程序,但这是以纯解释方式来运行的,而且程序的启动和退出都是借助于集成开发环境中的相应功能来实现的。为了提高应用程序的运行速度,并使之脱离 Visual Basic 的集成开发环境,能够在 Windows 操作系统环境中独立地运行,还需要把工程制作成一个 EXE 文件。

为此,可以从文件菜单中选择生成 Project.exe 命令,以打开如图 1.14 所示的生成工程对话框。

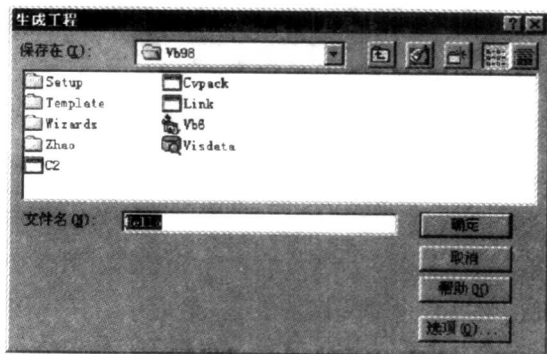


图 1.14 生成工程对话框

选择存放 EXE 文件的驱动器和目录以后,在文件名文本框中输入一个文件名,或者接受 Visual Basic 的默认设置,即以工程文件名作为要生成的 EXE 文件的名称。例如,在范例程序中,默认的 EXE 文件名为 Hello.exe。接下来单击确定按钮,此时会在工具栏右侧显



示出 Compiling... 字样。如果程序顺利地通过了编译,继而会显示出 Writing EXE... 字样。当这些字样消失以后,EXE 文件就已经建好了。我们可以在 Windows 资源管理器或我的电脑窗口中找到这个 EXE 文件的图标,用鼠标单击或双击此图标,便可以运行相应的应用程序,如图 1.15 所示。

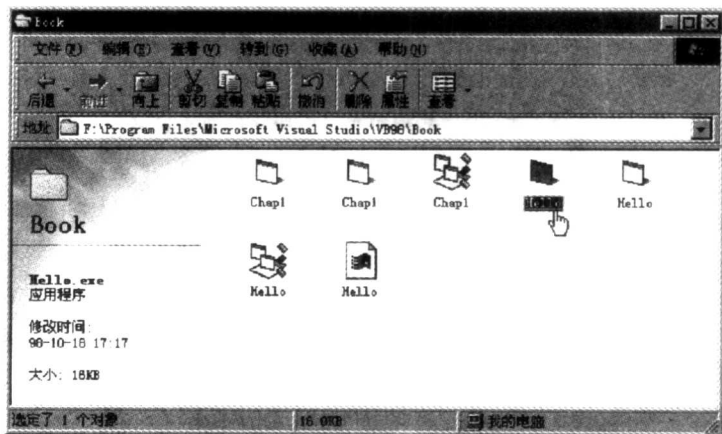


图 1.15 在“我的电脑”窗口中查看生成的 EXE 文件

生成 EXE 文件时,如果希望对编译过程进行干预,则应在单击确定按钮前进行以下操作:单击选项按钮,打开工程属性对话框,并在此对话框中对应用程序的版本号、版本信息、命令行参数以及编译方式(伪代码或本机代码)等选项加以设置。

EXE 文件生成以后,就可以利用 Windows 操作系统的标准功能来运行应用程序了。常用的启动方式有以下几种:

- 在 Windows 资源管理器或我的电脑窗口中单击或双击 EXE 文件图标。
- 在开始菜单中选择运行命令,然后在对话框中输入 EXE 文件的路径和文件名,并单击确定按钮。
- 在 Windows 桌面上或开始菜单中为 EXE 文件建立快捷方式,以便快速启动。

当应用程序独立于 Windows 操作环境下时,就不能再利用 Visual Basic 集成开发环境的功能来结束运行了,这时应当借助于程序本身所具有的功能来退出运行状态。我们在设计第一个程序时,并没有专门为它添加退出功能,不过,设计模式下的窗体在运行时已经成为一个标准的 Windows 应用程序窗口,其左上角有一个控制菜单框,右上角有一个关闭按钮,通过双击控制菜单框或单击关闭按钮,均可关闭该窗口,从而结束应用程序的运行。假如窗口既没有控制菜单,也没有关闭按钮,就需要求助于操作系统了。这时可以按下 CTRL + ALT + DEL 键,然后在关闭程序对话框中选择该程序,并单击结束任务按钮。

1.1.7 退出 Visual Basic

要退出 Visual Basic 的集成开发环境,可以选择下列操作之一:

- 从文件菜单中选择退出命令。
- 双击主窗口左上角的控制菜单框