

新版软件步步高 提 高 本

中文

Visual Basic 6.0

高级教程

捷新工作室 编著



- 强大的 Visual Studio 6.0 开发工具的重要成员
- 提供更加丰富的资源
- 提供丰富的设计向导
- 内容深入、全面
- 迅速提高编程水平

国防工业出版社

新版软件步步高(提高本)

# 中文 Visual Basic 6.0 高级教程

捷新工作室 编著

国防工业出版社

·北京·

## 图书在版编目(CIP)数据

中文 Visual Basic 6.0 高级教程 / 捷新工作室编著. -  
北京:国防工业出版社,1999.4  
(新版软件步步高·提高本)  
ISBN 7-118-02065-6

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

中国版本图书馆 CIP 数据核字(1999)第 01330 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

\*

开本 787×1092 1/16 印张 23 531 千字

1999 年 4 月第 1 版 1999 年 4 月北京第 1 次印刷

印数:1—4000 册 定价:32.00 元

---

(本书如有印装错误,我社负责调换)

## 总 序

在信息时代,知识成为推动社会生产力发展的一个最重要的因素,知识经济的轮廓在一些发达国家已经出现。以知识为基础的知识经济,其重要特点是信息产业的迅猛发展和产业的信息化,其内核是创新。我国是发展中国家,应该更加重视运用世界最新技术成果,有重点有选择地引进先进技术,增强自主创新能力,逐步实现技术发展的跨越。

在充分估量未来科学技术,特别是以计算机技术为先导的高技术发展对综合国力、社会经济结构和人民生活的巨大影响的基础上,为实现科教兴国战略多干实事,国防工业出版社组织了由数十位在计算机应用开发第一线工作的年富力强的博士、硕士组成的捷新工作室,编写出版《新版软件步步高(基础本)》和《新版软件步步高(提高本)》丛书。前者重在普及,后者追求提高,两者结合起来,力图满足多层次读者的需要。

《新版软件步步高(基础本)》的出版目的是普及新版软件的基本知识、基本操作技术,主要面向初学者,包括初次涉足该领域的机关、公司、企事业单位技术人员、大专院校师生及各类培训班学员,也可作为广大再就业职工理想的培训和学习教材。另外,对新技术感兴趣的读者也可将本丛书作为了解世界最新技术的窗口。

《新版软件步步高(提高本)》的出版目的是为有一定基础的读者找到提高专业技术水平和最新软件开发、操作技术的最佳途径,主要面向中高级读者,包括对该软件有一定基础知识,又希望提高自己专业技术水平的机关、公司、企事业单位技术人员、大专院校师生及各类高级培训班学员。

丛书的共同特点是突出一个“新”字,强调一个“精”字,力争一个“快”字。“新”是指软件的版本新;“精”是指精选的国内外流行最广、叫得最响的新版软件;“快”是指在保证质量的前提下,实现周期短,面市快。

丛书的内容覆盖最新高级语言开发环境(Visual J++ 6.0, Visual Basic 6.0, Visual FoxPro 6.0, Delphi 4.0),微机操作系统(中文 Windows 98, 中文 Windows NT 5.0),图形图像处理软件(Photoshop 5.0, 中文 CorelDRAW 8.0),Internet 浏览器(Internet Explorer 5.0),Web 页面设计环境(FrontPage 98),Internet 开发环境(Visual InterDev 6.0),大型数据库客户端开发工具(PowerBuilder 6.0)等。它们都是 1998 年以来推出的最新版软件。同时,我们还将把握新技术的脉搏,适时充实新的内容。

我们相信,丛书的出版必将为广大读者开辟跟踪、掌握、运用、进而创造新技术的最佳途径。

由于时间仓促,书中疏漏之处,敬请广大读者指正。

## 前 言

Basic 是一种适用性强、普及面广的程序设计语言,它的语法核心很小,外延性大,因此几乎所有编译系统都对它提供支持。小至计算器、单片机,大到大型计算机,都用到 Basic。作为全世界最大的微机软件制造商的美国微软公司,更是把 Basic 作为重要的程序设计语言加以实现。在“Visual Basic”这个词中,“Visual”指开发图形用户界面(GUI)的方法。不需编写大量代码去描述界面元素的外观和位置,而只要把预先建立的对象拖放到屏幕上的一点即可。“Basic”指的是 BASIC 语言。Visual Basic 在原有 BASIC 语言的基础上进一步发展,至今包含了数百条语句、函数及关键词,其中很多和 Windows GUI 有直接关系。

Visual Basic 是微软公司推出的面向 Windows 程序设计的可视化 Basic 程序设计集成环境,它将较为复杂的 Windows 程序设计简化成初学者和非计算机专业用户容易接受的 Basic 程序设计,这无疑给广大用户提供了在 Windows 环境下开发各自领域的应用程序的能力。

本书的特点在于引导初学者进入 Visual Basic 程序设计的美妙境界,它抓住了初学程序设计者必须掌握的几个关键内容:语言集成环境、语法、函数及过程设计、简单的数据结构及调试方法,同时通过图形设计和动态链接库设计等实用有趣的技术,使读者的综合能力得到加强。

Visual Basic 由于功能强大、简单易用而有着广泛的用户,尤其是在数据库、多媒体、商用系统的用户界面以及一般应用程序的设计等方面,Visual Basic 更是有其他开发环境所不能取代的优越之处。升级到 6.0 版后,其功能和性能都得到了增强,而汉化后的 Visual Basic 6.0 中文版,由于有全新的中文界面,更是方便了我国用户的使用。

Visual Basic 6.0 有三种版本,各自满足不同的开发需要:“Visual Basic 学习版”使编程人员轻松开发 Windows 95 和 Windows NT 应用程序;“专业版”为专业编程人员提供了一整套进行开发的功能完备的工具;“企业版”使得专业编程人员能够开发功能强大的分布式应用程序。本书以三种 Visual Basic 6.0 版本的公共方面为主题,系统、全面地介绍了 Visual Basic 6.0 中文版的高级使用方法。

专业人员可以用 Visual Basic 实现其他任何 Windows 编程语言的功能,而初学者只要掌握几个关键词就可以建立实用的应用程序。从开发个人或小组使用的小工具,到大型企业应用系统,都可用 Visual Basic 来实现。

总之,Visual Basic 提供了开发 Windows 应用程序的最迅速、最简捷的方法。不论是 Windows 应用程序的资深专业开发人员还是初学者,Visual Basic 都为他们提供了整套工具,以方便开发应用程序。

## 内 容 简 介

Visual Basic 由于功能强大、简单易用而有着广泛的用户,尤其是在数据库、多媒体、商用系统的用户界面以及一般应用程序的设计等方面,Visual Basic 更是有其他开发环境所不能取代的优越之处。升级到 6.0 版后,其功能和性能都得到了增强,而汉化后的 Visual Basic 6.0 中文版,由于有全新的中文界面,更是方便了用户的使用。

本书系统、全面地介绍了 Visual Basic 6.0 中文版的高级使用方法。全书共十二章,内容包括工程与程序编译、编程基础、标准控件、界面的进一步完善、文件操作、鼠标与键盘事件、文本与图形、程序调试和测试、程序优化、数据库操作、分发成品软件、创建安装程序。

本书是国防工业出版社《新版软件步步高(提高本)》丛书之一。本书内容全面,循序渐近,范例丰富,适合需要利用 Visual Basic 进行程序设计的专业或非专业高级开发人员参考使用,这对透彻理解 Visual Basic 6.0 中文版的特点有很大的帮助,它能使读者对 Visual Basic 的了解更加完整、更加结构化。如果在阅读的同时能够结合实际开发或使用,效果将更好。

# 目 录

## 第一章 工程与程序编译

|                               |    |
|-------------------------------|----|
| 1.1 工程简介 .....                | 1  |
| 1.2 创建、打开和保存工程 .....          | 3  |
| 1.3 添加、删除和保存文件 .....          | 3  |
| 1.4 在工程中添加控件 .....            | 4  |
| 1.4.1 在工程中添加 ActiveX 控件 ..... | 4  |
| 1.4.2 从工程中删除控件 .....          | 5  |
| 1.4.3 使用其他应用程序的对象 .....       | 6  |
| 1.4.4 使用资源文件 .....            | 6  |
| 1.5 制作可执行文件 .....             | 7  |
| 1.6 设置工程选项 .....              | 8  |
| 1.7 使用外接程序和向导 .....           | 9  |
| 1.7.1 外接程序管理器 .....           | 9  |
| 1.7.2 使用向导 .....              | 9  |
| 1.8 使用多个工程 .....              | 11 |
| 1.8.1 添加或删除工程 .....           | 11 |
| 1.8.2 指定一个启动工程 .....          | 12 |
| 1.9 将工程编译成本地代码 .....          | 12 |
| 1.10 命令行开关 .....              | 14 |
| 1.11 管理应用程序设置值 .....          | 14 |
| 1.11.1 创建或保存应用程序设置值 .....     | 15 |
| 1.11.2 检索应用程序设置值 .....        | 15 |
| 1.11.3 删除应用程序设置值 .....        | 16 |
| 1.12 使用资源文件 .....             | 17 |
| 1.13 使用模板 .....               | 18 |
| 1.14 条件编译 .....               | 19 |

## 第二章 编程基础

|                                  |    |
|----------------------------------|----|
| 2.1 Visual Basic 6.0 的代码窗口 ..... | 22 |
| 2.1.1 代码编辑窗口的编辑操作 .....          | 23 |
| 2.1.2 自动完成编码 .....               | 24 |
| 2.1.3 代码编辑窗口的选项 .....            | 24 |
| 2.2 Visual Basic 中的语句 .....      | 25 |

|                                |    |
|--------------------------------|----|
| 2.2.1 将单行语句分成多行 .....          | 25 |
| 2.2.2 将多个语句合并到同一行上 .....       | 26 |
| 2.2.3 在代码中添加注释 .....           | 26 |
| 2.3 数字系统和命名 .....              | 27 |
| 2.3.1 理解数字系统 .....             | 27 |
| 2.3.2 Visual Basic 的命名约定 ..... | 27 |
| 2.4 常量、变量、属性和赋值 .....          | 28 |
| 2.4.1 常数 .....                 | 28 |
| 2.4.2 变量 .....                 | 29 |
| 2.4.3 数据类型 .....               | 32 |
| 2.4.4 赋值语句 .....               | 37 |
| 2.5 顺序、选择和循环 .....             | 38 |
| 2.5.1 选择结构 .....               | 38 |
| 2.5.2 循环 .....                 | 40 |
| 2.6 过程和函数 .....                | 43 |
| 2.6.1 Sub 过程 .....             | 44 |
| 2.6.2 Function 过程(函数) .....    | 45 |
| 2.6.3 使用过程 .....               | 46 |
| 2.6.4 参数传递 .....               | 47 |
| 2.7 数组的使用 .....                | 51 |
| 2.7.1 数组的定义 .....              | 52 |
| 2.7.2 多维数组 .....               | 53 |
| 2.7.3 动态数组 .....               | 53 |

## 第三章 标准控件

|                                  |    |
|----------------------------------|----|
| 3.1 Visual Basic 6.0 的控件 .....   | 55 |
| 3.1.1 控件分类 .....                 | 55 |
| 3.1.2 内部控件 .....                 | 56 |
| 3.1.3 标准 ActiveX 控件 .....        | 57 |
| 3.1.4 更新旧版 Visual Basic 控件 ..... | 58 |
| 3.1.5 控件命名约定 .....               | 58 |
| 3.1.6 使用控件值 .....                | 59 |
| 3.2 使用控件数组 .....                 | 60 |
| 3.2.1 为何使用控件数组 .....             | 60 |
| 3.2.2 应用程序示例 .....               | 60 |



|                                |     |                                                      |     |
|--------------------------------|-----|------------------------------------------------------|-----|
| 4.3.4 “字体”对话框的使用 .....         | 132 | 5.5.1 打开二进制型访问的文件 .....                              | 169 |
| 4.3.5 “打印”对话框的使用 .....         | 134 | 5.5.2 在长度可变的字段中保存<br>信息 .....                        | 169 |
| 4.3.6 帮助文件的显示 .....            | 135 |                                                      |     |
| 4.4 Windows 95 控件 .....        | 136 | <b>第六章 鼠标与键盘事件</b>                                   |     |
| 4.4.1 ImageList 控件 .....       | 136 | 6.1 响应鼠标事件 .....                                     | 172 |
| 4.4.2 ListView 控件 .....        | 138 | 6.1.1MouseDown 事件 .....                              | 173 |
| 4.4.3 ProgressBar 控件 .....     | 138 | 6.1.2MouseMove 事件 .....                              | 174 |
| 4.4.4 Slider 控件 .....          | 139 | 6.1.3MouseUp 事件 .....                                | 175 |
| 4.4.5 StatusBar 控件 .....       | 139 | 6.2 检测鼠标按钮 .....                                     | 176 |
| 4.4.6 TabStrip 控件 .....        | 141 | 6.2.1 结合MouseDown 和 MouseUp 事件<br>使用 Button 参数 ..... | 178 |
| 4.4.7 Toolbar 控件 .....         | 141 | 6.2.2 结合 MouseMove 使用 Button<br>参数 .....             | 178 |
| 4.4.8 TreeView 控件 .....        | 142 | 6.2.3 使用 Button 参数增强图形鼠标<br>应用程序 .....               | 180 |
| 4.4.9 RichTextBox 控件的使用 .....  | 143 | 6.3 检测 Shift、Ctrl 及 Alt 键的<br>状态 .....               | 181 |
| 4.4.10 SSTab 控件的使用 .....       | 144 | 6.4 拖放 .....                                         | 183 |
| 4.5 多窗体与多文档用户界面 .....          | 145 | 6.4.1 启动自动拖动模式 .....                                 | 183 |
| 4.5.1 界面样式 .....               | 145 | 6.4.2 改变拖动图标 .....                                   | 183 |
| 4.5.2 多文档用户界面 .....            | 147 | 6.4.3 放下对象时的响应 .....                                 | 184 |
| 4.5.3 使用 MDI 窗体及其子窗体 .....     | 148 | 6.4.4 启动拖动或停止拖动时的<br>控制 .....                        | 184 |
| 4.5.4 MDI 用户界面的菜单设计 .....      | 150 | 6.4.5 改变控件的位置 .....                                  | 186 |
| <b>第五章 文件操作</b>                |     | 6.5 OLE 拖放 .....                                     | 187 |
| 5.1 Visual Basic 的文件操作命令 ..... | 153 | 6.5.1 启动自动 OLE 拖放 .....                              | 189 |
| 5.1.1 Shell 函数 .....           | 154 | 6.5.2 OLE 拖放 DataObject 对象 .....                     | 189 |
| 5.1.2 文件处理函数 .....             | 155 | 6.5.3 OLE 拖放如何工作 .....                               | 191 |
| 5.2 Visual Basic 中的文件控件 .....  | 156 | 6.5.4 启动 OLE 拖动操作 .....                              | 192 |
| 5.2.1 文件列表框 .....              | 156 | 6.5.5 将 OLE 拖动源拖到 OLE 拖放目标<br>的上方 .....              | 194 |
| 5.2.2 目录列表框 .....              | 158 | 6.5.6 将 OLE 拖动源放到 OLE 拖放目<br>标上 .....                | 197 |
| 5.2.3 驱动器列表框 .....             | 159 | 6.5.7 用鼠标和键盘修改放下效果和<br>用户反馈 .....                    | 200 |
| 5.2.4 连接所有文件控件 .....           | 160 | 6.5.8 创建自定义数据格式 .....                                | 200 |
| 5.3 使用顺序文件 .....               | 160 | 6.5.9 从资源管理器中拖动文件 .....                              | 201 |
| 5.3.1 从文件中读取信息 .....           | 162 | 6.6 自定义鼠标指针 .....                                    | 203 |
| 5.3.2 给现存文件增加信息 .....          | 163 | 6.7 响应键盘事件 .....                                     | 204 |
| 5.3.3 通用顺序文件 .....             | 164 | 6.7.1 编写低级键盘处理程序 .....                               | 204 |
| 5.3.4 将特殊字符送到顺序文件 .....        | 165 | 6.7.2 KeyPress 事件 .....                              | 205 |
| 5.3.5 修改顺序文件 .....             | 165 |                                                      |     |
| 5.4 使用随机存取文件 .....             | 167 |                                                      |     |
| 5.4.1 声明变量 .....               | 167 |                                                      |     |
| 5.4.2 定义记录类型 .....             | 167 |                                                      |     |
| 5.4.3 在类型定义中声明字段变量 .....       | 168 |                                                      |     |
| 5.4.4 定义其他变量 .....             | 168 |                                                      |     |
| 5.4.5 编辑随机型访问打开的文件 .....       | 168 |                                                      |     |
| 5.5 使用二进制文件 .....              | 169 |                                                      |     |

|                                |     |
|--------------------------------|-----|
| 6.7.3 KeyDown 和 KeyUp 事件 ..... | 206 |
| 6.7.4 编写窗体级键盘处理程序 .....        | 208 |
| 6.8 中断后台处理 .....               | 210 |
| 6.8.1 允许用户中断任务 .....           | 210 |
| 6.8.2 使用 DoEvents .....        | 211 |

## 第七章 文本与图形

|                                                   |     |                                      |     |
|---------------------------------------------------|-----|--------------------------------------|-----|
| 7.1 使用字体 .....                                    | 214 | 7.7.7 动态地改变控件大小 .....                | 240 |
| 7.1.1 为应用程序选择字体 .....                             | 214 | 7.7.8 创建简单动画 .....                   | 240 |
| 7.1.2 检查可用的字体 .....                               | 214 | 7.8 使用图形方法 .....                     | 241 |
| 7.1.3 设置字体特征 .....                                | 215 | 7.8.1 图形方法画图的原理 .....                | 242 |
| 7.1.4 把 Font 属性应用到特定的<br>对象 .....                 | 216 | 7.8.2 清除画图区 .....                    | 242 |
| 7.1.5 FontTransparent 属性 .....                    | 216 | 7.8.3 画点 .....                       | 242 |
| 7.2 在窗体和图片框上显示文本 .....                            | 217 | 7.8.4 画各种直线和形状 .....                 | 244 |
| 7.2.1 使用 Print 方法 .....                           | 217 | 7.8.5 绘制方框 .....                     | 245 |
| 7.2.2 文本的截断 .....                                 | 217 | 7.8.6 画圆 .....                       | 246 |
| 7.2.3 在指定位置显示 Print 输出 .....                      | 218 | 7.8.7 画圆弧 .....                      | 248 |
| 7.3 格式化数字、日期和时间 .....                             | 219 | 7.8.8 画椭圆 .....                      | 248 |
| 7.3.1 命名的格式 .....                                 | 220 | 7.8.9 在任意位置绘制图形 .....                | 249 |
| 7.3.2 数字的格式 .....                                 | 221 | 7.8.10 指定线宽 .....                    | 250 |
| 7.3.3 打印格式化的日期和时间 .....                           | 221 | 7.8.11 指定实线或虚线 .....                 | 250 |
| 7.4 使用选定的文本 .....                                 | 221 | 7.8.12 DrawMode 的显示控制作用 .....        | 251 |
| 7.5 使用剪贴板对象传送文本和<br>图形 .....                      | 222 | 7.8.13 窗体加载时创建图形 .....               | 252 |
| 7.5.1 使用剪贴板剪切、复制和粘贴<br>文本 .....                   | 222 | 7.9 使用颜色 .....                       | 252 |
| 7.5.2 在剪贴板上使用多种格式 .....                           | 224 | 7.9.1 在运行时指定颜色 .....                 | 253 |
| 7.5.3 检查剪贴板上的数据格式 .....                           | 225 | 7.9.2 使用 RGB 函数 .....                | 253 |
| 7.6 坐标系统概述 .....                                  | 226 | 7.9.3 使用颜色属性 .....                   | 253 |
| 7.6.1 改变对象的坐标系统 .....                             | 227 | 7.9.4 使用 256 种颜色 .....               | 255 |
| 7.6.2 创建自定义刻度 .....                               | 228 | 7.9.5 管理多个调色板 .....                  | 256 |
| 7.6.3 刻度的转换 .....                                 | 230 | 7.10 使用图片对象 .....                    | 258 |
| 7.7 使用图形控件 .....                                  | 230 | 7.10.1 使用图片对象数组 .....                | 258 |
| 7.7.1 给应用程序添加图片 .....                             | 231 | 7.10.2 使用图片对象代替 Windows<br>API ..... | 259 |
| 7.7.2 窗体和控件的图形属性概述 .....                          | 233 | 7.11 打印 .....                        | 259 |
| 7.7.3 用 AutoRedraw 创建持久的<br>图形 .....              | 233 | 7.11.1 从应用程序中打印 .....                | 260 |
| 7.7.4 用 ClipControls 裁剪区域 .....                   | 235 | 7.11.2 使用 Printer 对象 .....           | 261 |
| 7.7.5 用 AutoRedraw 和 ClipControls 使图<br>形分层 ..... | 236 | 7.11.3 捕获打印机错误 .....                 | 264 |
| 7.7.6 动态地移动控件 .....                               | 237 | 第八章 程序调试和测试                          |     |
|                                                   |     | 8.1 调试工具及其功能 .....                   | 265 |
|                                                   |     | 8.2 程序调试的基本方法 .....                  | 266 |
|                                                   |     | 8.2.1 设计、运行和中断模式 .....               | 266 |
|                                                   |     | 8.2.2 使用调试窗口 .....                   | 266 |
|                                                   |     | 8.2.3 使用中断模块 .....                   | 267 |
|                                                   |     | 8.2.4 使用断点 .....                     | 269 |
|                                                   |     | 8.2.5 使用监视窗口 .....                   | 270 |
|                                                   |     | 8.3 设置错误陷阱 .....                     | 271 |
|                                                   |     | 第九章 程序优化                             |     |
|                                                   |     | 9.1 优化速度 .....                       | 275 |

|                                           |     |                                    |     |
|-------------------------------------------|-----|------------------------------------|-----|
| 9.1.1 优化真实速度 .....                        | 275 | 组成 .....                           | 321 |
| 9.1.2 优化显示速度 .....                        | 279 | 11.2 安装盘制作向导的使用 .....              | 321 |
| 9.1.3 优化感觉速度 .....                        | 281 | 11.2.1 安装向导的启动 .....               | 321 |
| 9.2 优化大小 .....                            | 282 | 11.2.2 安装向导每一步的操作 .....            | 322 |
| 9.2.1 减小代码大小 .....                        | 283 | 11.3 从属文件及其生成 .....                | 328 |
| 9.2.2 修剪图形 .....                          | 285 | 11.3.1 从属文件解剖 .....                | 329 |
| 9.2.3 分段应用程序 .....                        | 286 | 11.3.2 创建从属文件 .....                | 330 |
| 9.2.4 优化对象 .....                          | 288 |                                    |     |
| 9.3 编译型和解释型程序 .....                       | 289 |                                    |     |
| <b>第十章 数据库操作</b>                          |     | <b>第十二章 创建安装程序</b>                 |     |
| 10.1 用 Data 控件访问数据库 .....                 | 291 | 12.1 安装向导和安装工具包 .....              | 332 |
| 10.1.1 Data 控件能做什么 .....                  | 291 | 12.1.1 启动 Visual Basic 安装向导 .....  | 332 |
| 10.1.2 Data 控件的属性 .....                   | 292 | 12.1.2 安装向导的功能 .....               | 332 |
| 10.1.3 创建简单的数据库应用程序 .....                 | 292 | 12.1.3 创建安装程序的步骤 .....             | 333 |
| 10.1.4 数据库的设计和结构 .....                    | 295 | 12.1.4 安装向导的高级功能 .....             | 334 |
| 10.1.5 记录集 .....                          | 296 | 12.2 从属文件 .....                    | 334 |
| 10.2 使用数据绑定控件 .....                       | 296 | 12.2.1 分析从属文件 .....                | 334 |
| 10.2.1 绑定控件的类型 .....                      | 297 | 12.2.2 使用安装向导生成从属文件 .....          | 336 |
| 10.2.2 添加、更新和删除记录 .....                   | 298 | 12.3 发布选项 .....                    | 337 |
| 10.2.3 使用 DBListBox 与<br>DBComboBox ..... | 298 | 12.4 使用安装向导模板 .....                | 339 |
| 10.2.4 使用 DBGrid .....                    | 301 | 12.5 安装数据访问部件 .....                | 339 |
| 10.2.5 在 DBGrid 中使用范围 .....               | 304 | 12.5.1 为 ODBC 应用程序创建安装<br>磁盘 ..... | 339 |
| 10.2.6 显示 DBGrid 中被计算的<br>字段 .....        | 304 | 12.5.2 安装远程自动化和分布式 COM<br>部件 ..... | 340 |
| 10.3 用代码操作记录 .....                        | 304 | 12.5.3 Internet 部件下载 .....         | 342 |
| 10.3.1 记录集的定位 .....                       | 305 | 12.6 使用安装工具包 .....                 | 347 |
| 10.3.2 查找指定的记录 .....                      | 306 | 12.6.1 修改安装工程 .....                | 347 |
| 10.3.3 用代码更新记录集 .....                     | 308 | 12.6.2 决定需要发布的文件 .....             | 348 |
| 10.3.4 控制事务 .....                         | 310 | 12.6.3 运行时文件 .....                 | 348 |
| 10.3.5 确认数据库的变化 .....                     | 312 | 12.6.4 安装文件 .....                  | 348 |
| 10.3.6 数据访问专用的 Update 方法 .....            | 313 | 12.7 决定在用户计算机上安装文件<br>的位置 .....    | 349 |
| 10.4 处理文件 .....                           | 314 | 12.8 决定发布磁盘的布局 .....               | 354 |
| 10.4.1 文件访问类型 .....                       | 314 | 12.8.1 压缩安装文件 .....                | 354 |
| 10.4.2 文件访问函数和语句 .....                    | 314 | 12.8.2 创建发布磁盘 .....                | 354 |
| 10.4.3 使用顺序文件访问 .....                     | 315 | 12.9 检测安装程序 .....                  | 354 |
| 10.4.4 使用随机文件访问 .....                     | 317 | 12.10 允许用户删除应用程序 .....             | 355 |
| 10.4.5 使用二进制文件访问 .....                    | 319 | 12.10.1 应用程序删除失败的可能<br>原因 .....    | 355 |
| <b>第十一章 分发成品软件</b>                        |     | 12.11 同时使用安装向导和安装工<br>具包 .....     | 356 |
| 11.1 Visual Basic 应用程序的                   |     |                                    |     |

# 第一章 工程与程序编译

工程是用来建造应用程序的文件和对象的集合。在工程的所有部件被汇集在一起并完成代码编写之后,便可以编译工程,创建一个可执行文件。

## 1.1 工程简介

在开发应用程序时,要使用工程来管理构成应用程序的所有不同的文件。

### 1. 工程的组成

一个工程包括:

(1) 跟踪所有部件的工程文件 (.vbp)。

(2) 每个窗体的文件 (.frm)。

(3) 每个窗体的二进制数据文件 (.frx),它含有窗体上控件的属性数据。对含有二进制属性(例如图片或图标)的任何 .frm 文件都是不可编辑的,这些文件都是自动产生的。

(4) 每个类模块的一个文件 (.cls),该文件是可选项。

(5) 每个标准模块的一个文件 (.bas),该文件是可选项。

(6) 一个或多个包含 ActiveX 控件的文件 (.ocx),该文件是可选项。

(7) 单个资源文件 (.res),该文件是可选项。

工程文件就是与该工程有关的全部文件和对象的清单,也是所设置的环境选项方面的信息,所有这些文件和对象也可供其他工程共享。当完成工程的全部文件之后,即可将此工程转换成可执行文件 (.exe);从“文件”菜单上选取“制作 project.exe”命令。

### 2. 工程资源管理器

当创建、添加或从工程中删除可编辑文件时,Visual Basic 会反映工程资源管理器窗口中发生的变化,该窗口包含此工程的当前文件的列表。图 1.1 所示的工程资源管理器窗口中列出可以纳入 Visual Basic 6.0 工程的一些文件类型。

### 3. 工程文件

每次保存工程,Visual Basic 都要更新工程文件 (.vbp)。工程文件包含文件列表,它与出现在工程资源管理器窗口的文件列表相同,工程文件还引用工程中所使用的 ActiveX 控件和可插入对象。

通过双击一个现存工程的图标,或从“文件”菜单中选定“打开工程”命令,或拖动该文件并放入工程资源管理器窗口,可以打开这个现存工程文件。

### 4. 工程结构

以下介绍的是可以纳入工程的不同类型的文件和对象:

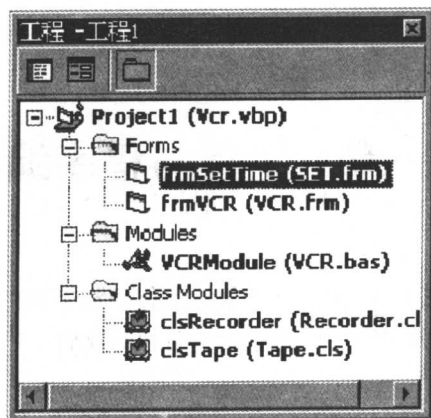


图 1.1 工程资源管理器窗口

(1) 窗体模块。窗体模块(具有 .frm 文件扩展名)包含窗体及其控件的正文描述,包括它们的属性设置。其中也含有窗体级的常数、变量和外部过程的声明,事件过程和一般过程。

(2) 类模块。类模块(具有 .cls 文件扩展名)与窗体模块类似,只是没有可见的用户界面。可以使用类模块创建含有方法和属性代码的自己的对象。

(3) 标准模块。标准模块(具有 .bas 文件扩展名)可以包含类型、常数、变量、外部过程和公共过程的公共的或模块级的声明。

(4) 资源文件。资源文件(具有 .res 文件扩展名)包含着无需重新编辑代码便可以改变的位图、字符串和其他数据。例如,如果计划用一种外语将应用程序本地化,可以将用户界面的全部正文串和位图存放在资源文件里,然后将资源文件本地化,而不是将整个应用程序本地化。一个工程最多包含一个资源文件。

(5) ActiveX 文档。ActiveX 文档(.dob)类似于窗体,但是在互联网资源管理器之类的互联网浏览器中是可以显示的。Visual Basic 6.0 的专业版和企业版能够创建 ActiveX 文档。

(6) 用户控件和属性页模块。用户控件(.ctl)和属性页(.pag)模块也类似于窗体,但它们被用于创建 ActiveX 控件及与其关联的用来显示设计时属性的属性页。Visual Basic 的专业版和企业版能够创建 ActiveX 控件。

(7) 部件。除文件和模块以外,还有几个其他类型的部件可以添加到工程中。

1) ActiveX 控件。ActiveX 控件(具有 .ocx 文件扩展名)是可选的控件,它可以被添加到工具箱中并在窗体里使用。当安装 Visual Basic 时,Visual Basic 包含的含有控件的那些文件被复制到一个公共目录(Windows 95 下面的子目录 \ Windows \ System)中。从多种源可以取得附加的 ActiveX 控件。也可以使用 Visual Basic 6.0 专业版和企业版创建自己的控件。

2) 可插入的对象。可插入的对象,例如 Excel 的工作表对象,是可用作建造集成方案时建造块的部件。一个集成方案可以包含由不同的应用程序创建的不同格式的数据,例如工作表、位图和正文。

3) 引用。也可以添加能被应用程序使用的外部 ActiveX 部件的引用。通过访问“工

程”菜单上的“引用”菜单项,使用“引用”对话框可指定引用。

4) ActiveX 设计器。ActiveX 设计器是类的设计工具,从类出发可以创建对象。窗体的设计界面是缺省的设计器。从其他的源可取得附加的设计器。

(8) 标准控件。标准控件是由 Visual Basic 6.0 提供的。标准控件,例如命令按钮或框架控件,总是包含在工具箱里,与可从工具箱里删除添加的 ActiveX 控件和可插入对象不同。

## 1.2 创建、打开和保存工程

“文件”菜单上的四个命令允许创建、打开和保存工程:

(1) “新建工程”。关闭当前工程,提示用户保存所有修改过的文件。可以从“新建工程”对话框选定一个工程类。然后 Visual Basic 6.0 创建一个带有单个新文件的新工程。

(2) “打开工程”。关闭当前工程,提示用户保存所有改动。随后,Visual Basic 6.0 打开一个现有工程,包括其工程文件(.vbp)中所列的窗体、模块和 ActiveX 控件。

(3) “保存工程”。更新当前工程的工程文件及其全部窗体、标准和类模块。

(4) “工程另存为”。更新当前工程的工程文件,用规定的文件名保存此工程文件。Visual Basic 还提示用户保存所有修改过的窗体或模块。

在工程间文件可以共享。像窗体这样的单个文件,可以是多个工程的组成部分。注意,在一个工程中的窗体或模块所做的改变,将会传播到共享这个模块的所有工程。

## 1.3 添加、删除和保存文件

在工程中使用多个文件和使用工程本身类似。要向工程中添加文件:

(1) 选定“工程”、“添加 filetype”(这里, filetype 是文件的类型),出现“添加窗体”对话框(图 1.2)。

(2) 选定一个现存的文件或一个新的文件类型,并选取“打开”。

在工程中添加文件时,是简单地将对于该现存文件的引用纳入工程,而不是添加该文件的复制件。因此,如果更改文件并保存它,这个更改会影响包含此文件的任何工程。若想改变文件而不影响其他工程,应在“工程资源管理器”里选定该文件,从“文件”菜单选取“filename 另存为”,然后以一个新的文件名保存此文件。

注意,可以从 Windows 的“资源管理器”、“文件管理器”或“网上邻居”拖动文件并放入“工程”窗口,将它们添加到一个工程。也可以拖动 .ocx 文件并放入工具箱,以添加新的控件。

要从工程中删除文件:

(1) 在“工程资源管理器”中选定该文件。

(2) 从“工程”菜单中选取“删除 filename”,此文件将从工程里删除掉,但是仍存在于磁盘上。

如果从工程里删除了文件,在保存此工程时 Visual Basic 更新此工程文件中的这个信

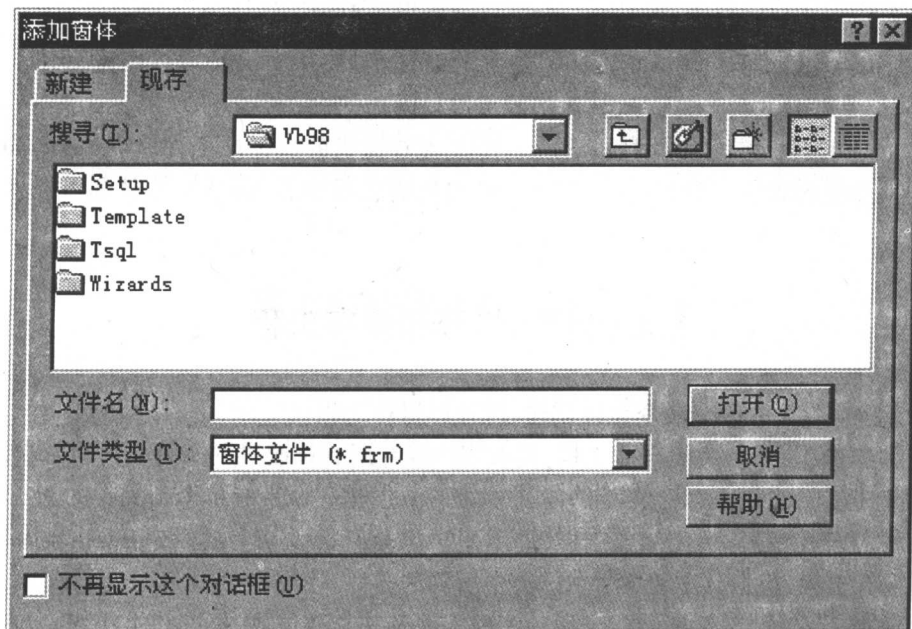


图 1.2 “添加窗体”对话框

息。

但是,如果在 Visual Basic 之外删除一个文件,Visual Basic 不能更新此工程文件,因此,当打开此工程时,Visual Basic 将显示一个错误信息,警告一个文件丢失。

要只保存文件而不保存工程:

- (1) 在“工程资源管理器”里选定此文件。
- (2) 从“文件”菜单中选取“保存 filename”。

也可从其他文件将现存的文本插入到一个代码模块中。在添加常数清单或添加可能保存在文本文件中的代码段时,这个功能很有用。要将文本文件插入代码:

- (1) 从“工程”窗口中选定要插入代码的窗体或模块。
- (2) 选取“查看代码”按钮,将光标移动到代码编辑器中要插入代码的地方。
- (3) 从“编辑”菜单中选取“插入文件”。
- (4) 选定欲插入的文本文件名,选取“打开”。

## 1.4 在工程中添加控件

工具箱里可用的控件集可被每个工程单独定义。任何给定的控件,在将其添加到工程的窗体之前,必须先位于工具箱里。

### 1.4.1 在工程中添加 ActiveX 控件

将 ActiveX 控件和可插入对象添加到工具箱中之后就能将它们添加到工程中。要将控件添加到工程的工具箱中:

(1) 从“工程”菜单上选取“部件”，出现“部件”对话框，如图 1.3 所示。列于此对话框中的项目包含全部登记的 ActiveX 控件、可插入对象和 ActiveX 设计器。

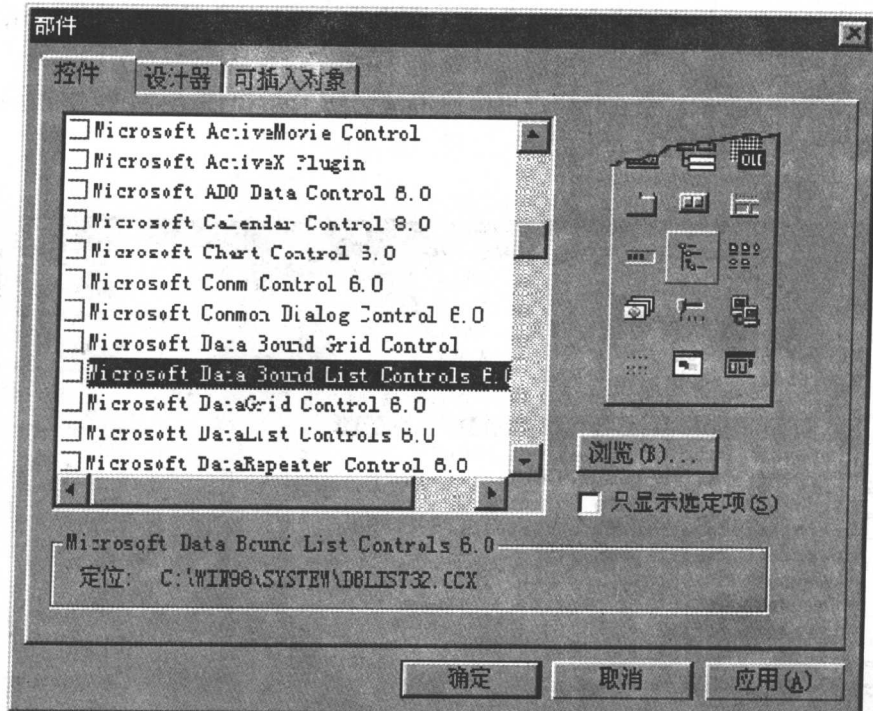


图 1.3 “部件”对话框

(2) 要向工具箱添加控件(具有 .ocx 文件扩展名)或可插入对象,选定控件名左面的复选框。为查看具有 .ocx 文件扩展名的控件,选定“控件”选项卡。为查看可插入对象,例如 Microsoft Excel 图表,选定“可插入对象”选项卡。

(3) 选取“确定”,关闭“部件”对话框。现在,所选定的全部 ActiveX 控件将出现在工具箱中。

要向“部件”对话框添加 ActiveX 控件,选取“浏览”按钮,搜索其他目录寻找具有 .ocx 扩展名的文件。在可用控件清单中添加 ActiveX 控件时,Visual Basic 自动选定该复选框。

注意,每个 ActiveX 控件有一个具有 .oca 扩展名的文件。该文件存储有高速缓存型库信息和该控件特有的其他数据。具有 .oca 扩展名的文件通常存储在与 ActiveX 控件相同的目录里,并可在需要时重建(文件大小和日期可以改变)。

#### 1.4.2 从工程中删除控件

要从工程中删除控件:

(1) 从“工程”菜单中选取“部件”,显示“部件”对话框。

(2) 清除要删除的每一个控件旁边的复选框,这些控件的图标将从工具箱内删除。

注意,若某个控件的一个实例被这个工程的任何窗体所用,则不能从工具箱中删掉这个控件。

### 1.4.3 使用其他应用程序的对象

也可以使用来自其他应用程序的对象,例如包含在 Excel 对象库中的对象,无论作为工具箱中的控件还是作为代码中的可编程对象都可使用。

要使其他应用程序的对象在代码中可以使用,但不是作为控件,应设置对那个应用程序的对象库的引用。要添加对其他应用程序的对象库的引用:

(1) 从“工程”菜单中选取“引用”,出现“引用”对话框,如图 1.4 所示。

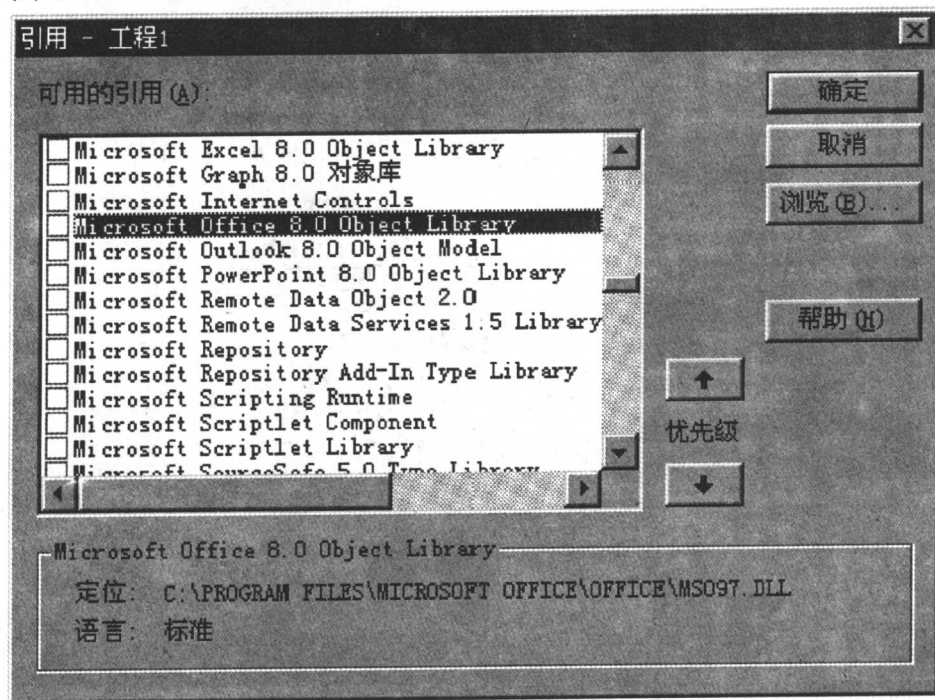


图 1.4 “引用”对话框

(2) 选定欲添入工程的每个引用旁边的复选框。

要添加未列入“引用”对话框的应用程序的引用,选取“浏览”按钮,然后选定这个应用程序。

(3) 选取“确定”,在工程中添加所选定的引用。

如果不是正在使用引用库中的任何对象,应当清除该引用的复选框,将 Visual Basic 必须解决的对象引用的数量减至最少,这样能减少工程编译占用的时间。

一旦设置了对所需对象库的引用,从“视图”菜单里选取“对象浏览器”,就可以在“对象浏览器”中找到一个特定的对象以及它的方法和属性。在代码中可以使用列在“对象浏览器”中的任何对象。

### 1.4.4 使用资源文件

资源文件允许将某一应用程序使用的全部特定版本的正文和位图汇集于一处,它可以包含常数声明、图标、屏幕文本和其他可在本地化版本之间或在各种修订版之间或特殊