



“九五”国家重点电子出版物规划项目·希望计算机知识普及教室系列
新世纪编程高手系列丛书 (1)

Windows 2000 编程利器

Visual C++ 编程高手

北京希望电脑公司 总策划
编程高手工作室 编写



本书光盘包括：
本书所有程序的源代码



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn



“九五”国家重点电子出版物规划项目·希望计算机知识普及系列
新世纪编程高手系列丛书(1)

TP312.0
159

Windows 2000 编程利器

Visual C++ 编程高手

北京希望电脑公司 总策划
编程高手工作室 编写



本书光盘包括：
本书所有程序的源代码

北京希望电子出版社
Beijing Hope Electronic Press
www.bhpe.com.cn

内 容 提 要

本书是在《Visual C++ 6.0 编程高手》基础上的新推出的 2000 版，编者在 1 年多的时间里收集、整理、加工、调试了大量的 Visual C++ 实用程序，并汇集成书。本书共计 36 章 206 个程序。这些程序覆盖了 Visual C++ 编程的主要应用：用户界面设计、多媒体（图形、图像、动画和声音）、网络（ActiveX 组件、Internet、和数据库）以及杂类等五大部分。其中，用户界面设计部分包括：按钮、编辑框、静态控件、组合框、列表框、树视、列视、工具条、状态条、菜单、其他控件、对话框、文档视窗模型、文件和文件夹、Shell、高级用户界面。多媒体部分包括 GDI、位图、打印、OpenGL、DirectX、动画、声音等。网络部分包括 ActiveX 控件、Internet、IE 编程、ISAPI、数据库等。杂类包括字符串处理、宏、剪贴板、日期和时间、DLL、调试、C++ 和 MFC 类、综合示例、工具和杂项。绝大部分程序包含源代码、项目文件和可执行文件，可以直接为你所用。

由于篇幅所限，本书主要介绍各个程序的功能及设计思想，而其完整源程序和演示程序包含在与本书配套的光盘的相关目录中。

本书内容新、丰富、实用性和指导性强。本书不但是对从事 Visual C++ 编程的新手还是老手，以及其他编程语言，比如 VB、Delphi 的程序员，都具有重要的参考价值，“拿来就用”而且也可作为大专院校相关专业师生的自学、教学参考书和社会相关领域培训班推荐教材。

本书配套光盘内容包括：本书所有程序的源代码。所有代码已在 Windows 98 和 Windows 2000 下调试成功。

- 系 列 书： 新世纪编程高手系列丛书（1）
书 名： Windows 2000 编程利器 — Visual C++ 编程高手
总 策 划： 北京希望电脑公司
文 本 著 者： 编程高手工作室
责 任 编 辑： 龙启铭
CD 制 作 者： 希望多媒体创作中心
CD 测 试 者： 希望多媒体测试部
出 版、发 行 者： 北京希望电子出版社
地 址： 北京中关村大街 26 号，100080
网 址： www.bhp.com.cn E-mail: lwm@hope.com.cn
电 话： 010-62562329,62541992,62637101,62637102
010-62633308,62633309（图书发行，技术支持）
010-62613322-215（门市） 010-62531267（编辑部）
- 经 销： 各地新华书店、软件连锁店
排 版： 希望图书输出中心
CD 生 产 者： 中新联光盘有限责任公司
文 本 印 刷 者： 北京媛明印刷厂
开 本 / 规 格： 787×1092 毫米 1/16 开本 22.875 印张 530 千字
版 次 / 印 次： 2000 年 11 月第 1 版 2000 年 11 月第 1 次印刷
印 数： 0001~5000 册
本 版 号： ISBN7-900049-66-5/TP·66
定 价： 39.00 元（1CD，含配套书）

说明：凡我社光盘配套图书若有缺页、倒页、脱页、自然破损，本社发行部负责调换

前 言

本书及其配套光盘汇集了 99 年以后来自世界各地的 Visual C++ 编程好手愿意无偿公开的源程序。其中既有短至几行却非常关键的代码，更有一个类的完整实现，还有非常实用、大型的完整应用程序。无论对于 Visual C++ 新手还是老手，以及其他编程语言，比如 VB、Delphi 的程序员，都具有很好的参考价值。这样，如果你正在用 Visual C++ 做开发，请一定要先浏览本书，也许你现在需要解决的问题，他人已帮助你解决，无需费时费力了。如果你能从本书中找到一个程序被你所用，则购买本书就非常值得了。而事实上，全书共包含大约 200 多个程序，这样，找到一个程序可用的可能性非常高！没有什么可犹豫的，赶紧行动吧。不过，请一定注意，在使用这些程序时，请尊重程序作者的本人的劳动成果，给出程序的出处和作者姓名。

这些程序覆盖了 Visual C++ 编程的主要应用：用户界面设计、多媒体（图形、图像、动画和声音）、网络（ActiveX 组件、Internet 和数据库）以及杂类。其中，用户界面设计部分包括：按钮、编辑框、静态控件、组合框、列表框、树视、列视、工具条、状态条、菜单、其他控件、对话框、文档视窗模型、文件和文件夹、Shell、高级用户界面。多媒体部分包括 GDI、位图、打印、OpenGL、DirectX、动画、声音等。网络部分包括 ActiveX 控件、Internet、IE 编程、ISAPI、数据库等。杂类包括字符串处理、宏、剪贴板、日期和时间、DLL、调试、C++ 和 MFC 类、综合示例、工具和杂项。绝大部分程序包含源代码、项目文件和可执行文件，直接为你所用。

全书按编程主要涉及的方面进行分类，每一方面包含若干个程序。其中，由于篇幅所限，本书主要介绍程序的功能和界面；而其完整源程序和演示程序包含在光盘相关目录中。另外，全部程序在 Visual C++ 6.0、Windows 95/98/NT/2000 上编译通过，运行成功。如果你在使用过程中有问题，请 email 到 code.expert@263.net，并请详细说明 bug 的细节，以便有针对性解决你的问题。

本书由编程高手工作室编写，参加本书编写以及光盘开发的人员包括王福建、郭月强、焦祝军、卢绍青、马照亭、钱秀槟、任志杰、孙俊、陶冶、王海滨、尹宝林、袁帅、邹秋玲、杨艳、吴宏志、任玉霞、解建伟、陶榕、孔刚等。在此，对所有为本书和光盘出版付出辛勤劳动的人们表示衷心的感谢。

JSSOP 2007

目 录

第一部分 用户界面设计

第1章 按钮控件..... 3	程序4 属性列表框..... 48
程序1 带热点的按钮..... 3	程序5 Visual Studio 风格的列表框..... 50
程序2 高级按钮控件..... 4	第6章 TREEVIEW 控件..... 54
程序3 弹性按钮..... 9	程序1 定制的TreeView 控件..... 54
程序4 带选项列表的按钮..... 11	程序2 用来显示磁盘文件目录的 TreeView 控件..... 55
程序5 可重叠的按钮控件..... 12	程序3 在TreeView 控件之间实现拖放.... 58
程序6 由静态文本类继承的按钮..... 12	程序4 带有TreeView 控件的属性页..... 59
程序7 任意形状的按钮..... 13	程序5 IE 高级选项风格的TreeView 控件.. 61
第2章 编辑控件..... 15	第7章 工具条..... 64
程序1 自动填充的编辑框..... 15	程序1 可定制的工具条和菜单..... 64
程序2 货币格式的编辑框..... 15	程序2 在多文档界面中切换的工具条.... 67
程序3 对语法敏感的编辑框..... 17	第8章 状态条..... 72
程序4 包含提示文本和编辑框的 复合编辑控件..... 20	程序1 在状态条中播放 AVI 文件..... 72
程序5 高级编辑控件..... 22	程序2 扩展型的状态条..... 73
程序6 数字滚选编辑框..... 23	第9章 菜单控件..... 77
第3章 静态控件..... 25	程序1 高级弹出菜单..... 77
程序1 带有水平或垂直文字的斜线..... 25	程序2 可停靠的菜单条..... 78
程序2 彩色静态文本框..... 26	程序3 为控件设计的平面弹出式菜单..... 79
程序3 静态直方图..... 29	第10章 其他公用控件..... 83
程序4 跳跃的静态文本..... 30	程序1 数据测绘的绘图类..... 83
程序5 显示设备无关位图 (DIB) 的 静态控件..... 31	程序2 Macintosh 按钮..... 85
第4章 组合框..... 32	程序3 数字钟..... 85
程序1 多列组合框..... 32	程序4 完全用户定制的标签控件..... 86
程序2 可定制的颜色选择器..... 33	程序5 形状可改变的进度控件..... 88
程序3 边框变换的组合框..... 34	程序6 Smart Grid—表格控件..... 89
程序4 类似 MS Office 风格的组合框..... 38	程序7 静态发光二极管控件..... 93
程序5 只读组合框..... 39	程序8 液晶显示控件..... 94
第5章 列表控件..... 41	第11章 对话框..... 96
程序1 双列表选择管理器..... 41	程序1 可“钉住”的对话框基类..... 96
程序2 彩色文本列表框..... 43	程序2 扩展 MFC 文件通用对话框..... 97
程序3 多级列表框..... 44	程序3 Corel PhotoHouse 风格的 颜色选择对话框..... 101

程序 4	用来选择和建立文件夹的对话框...	102
程序 5	文件预览对话框.....	104
程序 6	多文件选择对话框.....	105
程序 7	一个检验对话框中的所有控件 变化的简便方法	108
程序 8	堆叠的对话框.....	111
程序 9	标签对话框类.....	113
程序 10	用通用文件对话框作为 无模式的视图	115
第 12 章	文档/视窗模型.....	116
程序 1	一个定制的 MDI 客户区.....	116
程序 2	在多文档窗口中增加 “More Windows”对话框	117
程序 3	MDI 程序中的文档选择器	120
程序 4	MDI 程序的自动标签条	121
程序 5	一个文档对应两个视图.....	124
第 13 章	文件和文件夹.....	126

程序 1	CFileInfoArray—文件信息 和比较.....	126
程序 2	建立多级目录	134
程序 3	CStringFile 类.....	137
第 14 章	SHELL 编程.....	138
程序 1	系统条动画图标类	138
程序 2	命令行参数菜单扩展	138
第 15 章	高级用户界面	141
程序 1	Outlook 98 风格的条形控件.....	141
程序 2	动态子窗口的重新配置	147
程序 3	建立定制窗口标题的框架	147
程序 4	信息条	150
程序 5	增强的列表控件	151
程序 6	可视化的框架	156
程序 7	皮肤生成器	159
程序 8	α 混合透明窗口	161

第二部分 多媒体

第 16 章	GDI.....	165
程序 1	自动字体处理类.....	165
程序 2	设置 LOGFONT 结构的简单方法.....	165
程序 3	字体选择组合框.....	167
程序 4	JTDraw—一个绘图的例子.....	168
程序 5	旋转图片—高级内存设备环境.....	169
第 17 章	位图.....	174
程序 1	设备无关位图包.....	174
程序 2	向 CBitmap 中读取位图文件的 快速方法	176
程序 3	真彩滚轮.....	177
程序 4	在 24 位颜色位图中直接访问像素..	178
程序 5	高品质的图像旋转.....	180
程序 6	半透明位图.....	180
程序 7	静态透明图像类.....	182
程序 8	观察 PCX 文件.....	183
程序 9	一个 TWAIN 的 C++ 包	186
第 18 章	打印.....	189

程序 1	定制通用打印对话框	189
程序 2	设置默认打印机的函数	190
程序 3	Windows 文件的一般 打印/预览程序.....	194
程序 4	不需 MFC 支持的打印预览	195
程序 5	非文档/视结构的打印预览	199
程序 6	打印文本函数的封装类	200
第 19 章	OPENGL 编程.....	205
程序 1	三维字体	205
程序 2	OpenGL 的图像旋转缩放处理.....	207
程序 3	简单 OpenGL 类	209
程序 4	将一个 OpenGL 图像 发送到剪贴板.....	210
程序 5	使用 24 位位图作为 OpenGL 的图像	212
第 20 章	DIRECTX	214
程序 1	用 MFC 进行简单的 直接写屏 (DirectDraw)	214

程序 2 高级的直接写屏类..... 215
 第 21 章 动画和声音..... 216
 程序 1 AVI 图像捕获包 216
 程序 2 录制、播放、存储声音文件的

多线程类..... 217
 程序 3 简单的混音控制器包 219
 程序 4 透明的动画控件 220

第三部分 网络和数据库

第 22 章 ACTIVEX 控件 225
 程序 1 ActiveX 超级链接控件..... 225
 程序 2 自动填充控件..... 227
 第 23 章 INTERNET..... 229
 程序 1 从 IE 中向列表控件中
 拖拽链接地址 229
 程序 2 建立 Internet 快捷方式..... 231
 程序 3 创建拨号连接对象..... 232
 程序 4 简单的邮件客户端程序..... 233
 程序 5 简单聊天程序..... 236
 程序 6 CFtpGet—下载 FTP 文件类..... 240
 程序 7 Socket 传输程序 242
 程序 8 HTTP 服务监视器 243
 第 24 章 IE 编程 244
 程序 1 清除历史纪录..... 244
 程序 2 删除 Internet 的暂存文件..... 244
 程序 3 怎样将 CHtmlView 的内容
 添加到剪贴板中 246
 程序 4 用 ATL HTML 控件实现浏览器的

镶嵌窗口..... 247
 第 25 章 ISAPI..... 248
 程序 1 浏览 IIS 目录结构 248
 程序 2 ISAPI 扩展 DLL 中的 Cookies 248
 第 26 章 数据库..... 249
 程序 1 从 Visual C++中用 Access
 自动生成报表..... 249
 程序 2 在 Visual C++中访问 Borland
 数据库引擎(BDE)..... 252
 程序 3 浏览 ODBC 的数据源名 (DSN) 258
 程序 4 CByteArrayFile—串行化对象
 到数据字段..... 258
 程序 5 CODBCdynamic— 一个从 ODBC 数据源
 中动态读取数据的类..... 260
 程序 6 DAO 组合框..... 262
 程序 7 ODBC 的日期时间..... 264
 程序 8 交互式的 SQL 工具 (使用 ADO) ... 265
 程序 9 交互式的 SQL 工具 (使用 MFC) ... 265
 程序 10 在 DLL 中使用 DAO..... 265

第四部分 杂类

第 27 章 字符串..... 271
 程序 1 在 Windows95/98 上处理
 UNICODE 字符串..... 271
 程序 2 进行资源读取和字符串
 格式化的 CString 扩展类 271
 程序 3 不依赖系统的 CString 类..... 273
 程序 4 CUnicodeString 类 276
 程序 5 简单字符串类..... 276
 第 28 章 宏..... 280

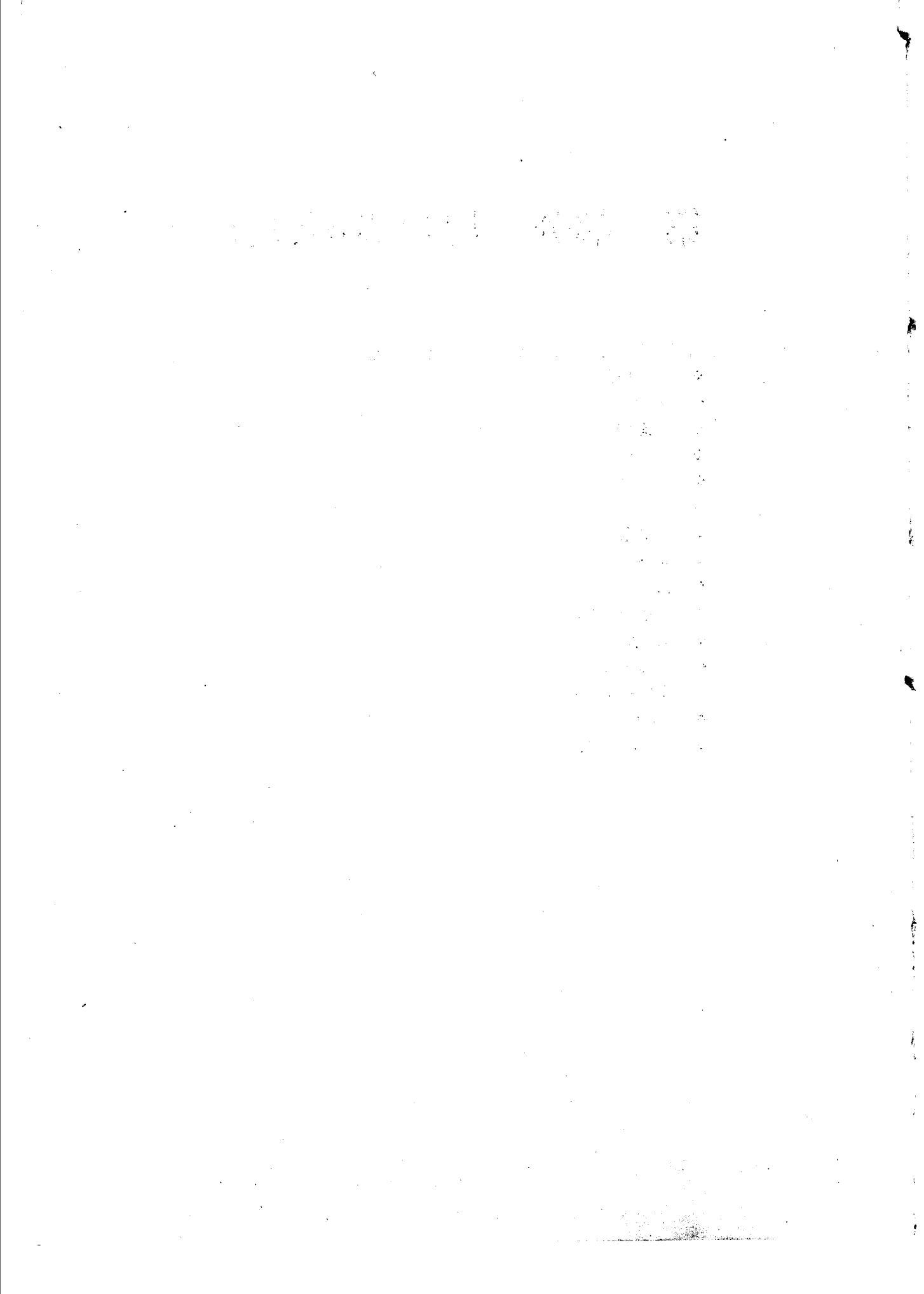
程序 1 C++操作符宏 280
 程序 2 注释块宏 280
 程序 3 定制的应用向导—
 “关于”对话框..... 280
 程序 4 程序行计数器插件 282
 程序 5 SmartHelp 帮助文件插件..... 283
 程序 6 WM_COMMAND 用户消息宏..... 283
 程序 7 去除替换字符和重复的注释 285
 第 29 章 剪贴板..... 287

程序 1 ClipConvert——剪贴板数据转换器	287
程序 2 从资源管理器中复制文件名	288
第 30 章 时间和日期	290
程序 1 一周时间选择器	290
第 31 章 DLL	292
程序 1 建立非 MFC DLL 的静态链接	292
程序 2 显示已装载模块	298
程序 3 运行时动态调用 DLL	299
第 32 章 DEBUG	303
程序 1 LibDump	303
程序 2 QProfile 工具包— 一个评估代码的类	303
程序 3 Visual C++ 定制 Debug 监视器	304
第 33 章 C++ 和 MFC 类	307
程序 1 强大的函数解析	307
程序 2 CEditDist 抽象类	307
程序 3 环形缓冲区	308
程序 4 ComprLib—数据压缩类库	308
程序 5 观察者模式类	311
程序 6 CStringArray 的快速排序	313
程序 7 CArray 模板类的快速排序	314
程序 8 用名字实例化类	315
程序 9 独立于 AVL 库的存储媒体	316
程序 10 TimeStamp—替代 ColeDateTime 和 ColeDateTimeSpan	316
第 34 章 综合实例	317
程序 1 磁盘空间监测器	317
程序 2 取回密码程序	317
程序 3 Outlook 风格的数据库浏览器	318
第 35 章 常用工具	324
程序 1 命令行查找工具	324
程序 2 Win32 DEF 文件生成器	325
程序 3 文件快速分隔器	326
程序 4 资源查找器	327
程序 5 StampVer—版本资源修改器	327
程序 6 存储过程类向导	329
程序 7 系统颜色变换器	330
第 36 章 其他杂类	332
程序 1 文件被其它程序更改时提醒用户 ..	332
程序 2 CIniFile—读写 INI 文件的类	332
程序 3 建立一个自解压的可执行文件	334
程序 4 得到版本信息的 CWinApp 派生类 ..	336
程序 5 在任何从 CWnd 派生的窗口中 实现拖放	336
程序 6 综合系统日志	339
程序 7 在标题栏和菜单条之间 加入标识图	340
程序 8 MFC 窗口风格预览器	342
程序 9 图形选择缩放类	343
程序 10 简单的消息服务	346
程序 11 简单的函数式计算器	352
程序 12 非窗口类的计时器支持	354
程序 13 在内部过程通信 (IPC) 中 使用内存映射文件	355
程序 14 Win32 位图屏幕保护程序	355
程序 15 xSleep—Sleep 函数的改进	355
程序 16 在 MFC 中使用 Zip 和 UnZip	355

第一部分 用户界面设计

本部分收集的程序主要与用户界面设计有关，包括下述方面：

- ❖ 按钮控件
- ❖ 编辑框
- ❖ 静态控件
- ❖ 组合框
- ❖ 列表框
- ❖ 树视
- ❖ 工具条
- ❖ 状态条
- ❖ 菜单
- ❖ 其他公共控件
- ❖ 对话框
- ❖ 文档/视窗
- ❖ 文件和文件夹
- ❖ Shell
- ❖ 高级用户界面



第 1 章 按钮控件

程序 1 带热点的按钮

本程序定义的 `CHotButton` 类是按钮类的子类，这个按钮控件在它的右面部分有一个“热点”的区域，这使得它成为一个具有双重功能的按钮。在点击热点区域时，向程序发送 `BN_HOT_CLICKED` 的消息；而点击非热点区域时，发送的是普通的 `BN_CLICKED` 消息。参见图 1.1。
















图 1.1 带热点的按钮

`CHotButton` 类有两个键盘焦点区——左边的文字部分和右边的热点区部分。鼠标、`Tab` 键和 `Shift-Tab` 键可以操纵焦点在这两个区域之间的转换。从这方面看，它就象两个按钮一样。

本程序定义了 13 个内部的热点位图，调用不同的函数时，可以在按钮上显示出不同的位图。参见表 1.1。

表 1.1 热点位图

函数	位图
<code>UseXBitmap()</code>	
<code>UseCheckBitmap()</code>	
<code>UseFolderBitmap()</code>	
<code>UsePaperBitmap()</code>	
<code>UsePrinterBitmap()</code>	
<code>UseRightArrowBitmap()</code>	
<code>UseLeftArrowBitmap()</code>	
<code>UseLEDBitmap()</code>	
<code>UseEllipseBitmap()</code>	
<code>UseBinocularBitmap()</code>	
<code>UseClockBitmap()</code>	
<code>UseRedFlagBitmap()</code>	
<code>UseBannedBitmap()</code>	

当然，也可以在应用程序中定义位图，但自定义的位图必须包含大小相等的两张并排的图样，分别是活动和非活动时的图形，参见图 1.2。



图 1.2 活动和非活动图样

方法 `CHotButton::SetAppBmp(UINT nBitmapID, int cx, COLORREF crMask)` 用来安装应用程序定义的位图。参数 `nBitmapID` 用来定义位图的资源 ID 号；`cx` 用来定义单个图片（活动或非活动的）的宽度，`crMask` 定义了透明部分的颜色，在这个例子里是紫色——`RGB(255,0,255)`。

程序通过 `UseAppBitmap(int iBitmap)` 方法来选择所用的一对热点区的位图，`iBitmap` 参数是所选用的图形的索引号。

通过 `SetTooltipText(LPCSTR szText, BOOL bActivate=TRUE)` 可以为热点区设置工具条提示，`szText` 是用来显示的提示文字，`bActivate` 是设置工具条是否被激活的初始状态的，可以通过 `ActivateTooltip(BOOL bEnable = TRUE)` 来设定工具条提示是否为可用状态。

按钮上的热点区可以通过 `EnableHotSpot(BOOL bEnable = TRUE)` 来设置是否有效。当它为无效状态时，总是显示无效时的图形，工具条提示不显示，当点击热点区时，发送的是普通的 `BN_CLICKED` 消息。热点区的图形可以通过 `ShowBitmap(BOOL bShow = TRUE)` 来设置是否可见，当图像被设为隐藏状态时，这个按钮看上去和普通的按钮是一样的。

这个按钮控件也存在一些局限：

- 只支持立体的框架风格；
- 支持水平居中/左对齐/右对齐的文本对齐方式，但不支持垂直居中的方式；
- 不支持多行文本；
- 当处理 `WS_DISABLED` 消息时，文本并不变灰，但将会在今后的版本进行改进。

其演示程序参见 `01\01\demo`。

程序 2 高级按钮控件

这个名为 `KSet` 的按钮控件是为那些希望在 Windows 的标准命令按钮之外多一些选择的开发者设计的。它在提供了标准按钮具有的一切功能的同时还附加了许多新的易用功能。

这个控件有几种状态：一般状态，鼠标覆盖时，按下鼠标时以及无效时的状态，它可以包含文本和图像并且可以灵活的改变按钮的外观。参见图 1.3。值得注意的是，`KSet` 按钮控件并不是从已经存在的窗口类继承下来的子类。

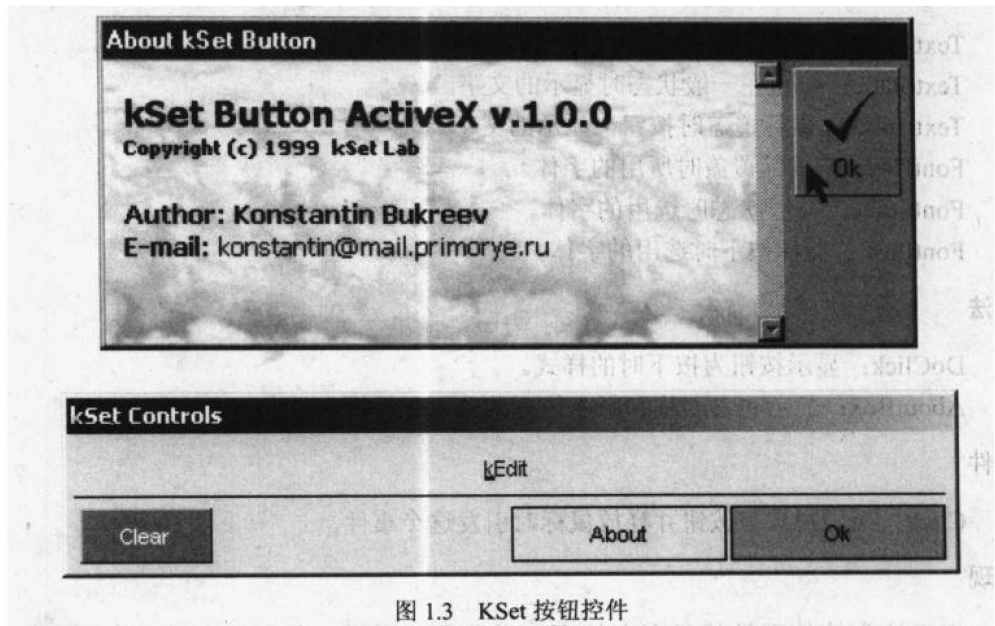


图 1.3 KSet 按钮控件

这个按钮控件提供了丰富的属性设置。可以设置它在各种状态下的外观，包括可以独立地改变按钮的图像、字体、边界颜色、阴影等。

属性

- Enabled:** 是否可以接受用户的输入。
- HoverEnabled:** 允许鼠标覆盖时的状态。
- AppearanceStatic:** 控件在无焦点时的静态外观。
- AppearanceHover:** 控件在鼠标覆盖时的外观。
- AppearancePush:** 控件在鼠标按下时的外观。
- ShowFocus:** 有焦点时的显示风格（在按钮上画一个矩形框）。
- ShowStateAsFlat:** 按钮为平的或是为有边框的外观风格。
- BorderColor:** 按钮的边界颜色。
- HighlightColor:** 当按钮为突出显示时的颜色。
- ShadowColor:** 按钮具有阴影效果时的阴影颜色。
- TextPushColor:** 按钮被按下时文字的前景色。
- TextHoverColor:** 按钮在鼠标覆盖时的文字的前景色。
- TextStaticColor:** 按钮在一般状态时的文字颜色。
- PictureAlignment:** 按钮上图片和文字的位置关系。
- PictureSpacing:** 图片和标题文字之间的间隔。
- PicturePush:** 当按钮被按下时显示的图片。
- PictureHover:** 鼠标覆盖时显示的图片。
- PictureStatic:** 一般状态时显示的图片。
- PictureDisabled:** 当控件无效时，在一般状态时显示的图片会自动地转换成无效样式的外观，无效状态可以通过设置这个属性来实现。

TextPush: 按钮在按下状态时显示的文字。
TextStatic: 按钮在一般状态时显示的文字。
TextHover: 鼠标覆盖时按钮上显示的文字。
FontHover: 鼠标覆盖时所用的字体。
FontStatic: 一般状态时选用的字体。
FontPush: 鼠标按下时选用的字体。

方法

DoClick: 显示按钮为按下时的样式。
AboutBox: 显示包含名称和版本信息的关于对话框。

事件

Click: 当用户按下按钮并释放鼠标时引发这个事件。

实现

由于这个控件不是从已存在的窗口类继承的子类，所有用来显示按钮的代码都在 `_Render` 的复合类中。这个按钮使用的是 `HFONT`、`HPEN` 一类的简单的 GDI 句柄，用一组函数来直接绘制的，这组函数的指针保存在一个静态数组中。

```
[renderbutton.h]
template <class T>
class _Render
{
    ...
    typedef void (_Render:: *PTR_FDRAW)(kDrawDC& dc, RECT& rc);
    static PTR_FDRAW& f_table (int appearance, int status)
    {
        static PTR_FDRAW& f_table (int appearance, int status)
        {
            //appearance, status
            static PTR_FDRAW table[8][4] = {...};
            ...
            return table[appearance][status];
        }
    }
    ...
};

template <class T>
void _Render<T>:: DrawIt(HDC hdc, int state, RECT& rc)
{
    PTR_FDRAW ff;

    // We choose function depending on a state and appearance of the button
    switch (state)
```

```

{
case 0:
    ff = f_table(pT->m_AppearanceStatic, state);
    ...
}

//draws a button
(this->*ff)(_kDrawDC(hdc), rc);
...
//display the text
DrawState(...);
...
//draws focus rectangle if need
DrawFocusRect(hdc, &rc_focus);
...
//display an image through method IPicture interface
pPic->Render(...);
...
}

```

这里我们使用了 user32.lib 的 TrackMouseEvent 函数来跟踪鼠标指针。

```

[kbutton.cpp]
void _kButton: : DetectTrackMouse()
{
    m_pfTrackMouseEvent=(PTRF_TrackMouseEvent)GetProcAddress
        (GetModuleHandle(_T("user32")),_T("TrackMouseEvent"));
}

```

或者在 Windows 95 中, 通过类 _kMouseMoveImpl 从第二个线程来跟踪鼠标的移动。
_kMouseMoveImpl 类从 _ApartmentThread<>继承而来, 实现了 IMouseMove 接口。

```

[mousemove.h]
class ATL_NO_VTABLE _kMouseMoveImpl :
public _IMouseMove,
public CComTearOffObjectBase<_kButton>,
public _ApartmentThread<_kMouseMoveImpl, _IMouseMove, &IID__IMouseMove>
{
    ...
}
[kbutton.h]
COM_INTERFACE_ENTRY_CACHED_TEAR_OFF(IID__IMouseMove,
    _kMouseMoveImpl, m_pUnkMouseMove.p)

```

```

[kbutton.cpp]
void _kButton: : TrackMouse()
{
    if (m_pfTrackMouseEvent)
    {
        ...
    }
}

```

```

    }
    else
    {
        CComQIPtr<_IMouseMove> pI(GetControllingUnknown());
        pI->_StartThread(100);
    }
}

```

使用这种接口的类可以节省内存，因为 `kMouseMoveImpl` 只在 Windows 95 下才激发。这种情况也应用于鼠标敲击动作（例如，调用 `DoClick` 方法）。

```

[kButton.cpp]
STDMETHODIMP _kButton: : DoClick(VARIANT_BOOL bAnimate)
{
    ...
    CComQIPtr<_IAnimationClick> pI(GetControllingUnknown());
    pI->_StartThread(100);
    ...
}

```

`_ApartmentThread` 类封装了与 STA 的工作——创建线程，调度接口指针等等。

```

[apartmentthread.h]
template <class Derived, class T, const IID* piid>
class _ApartmentThread
{
    ...
    HRESULT _StartThread(int nTimeOut)
    {
        ...
        if (SUCCEEDED(CoMarshalInterThreadInterfaceInStream(*piid,
            (T*)pDerived, &m_pStream)))
            m_hThread = CreateThread(0, 0, &Apartment, (void*)this, 0, &dw);
        ...
    }
    ...
}

```

控件的属性是由 `atlctr.h` 文件中的 `IMPLEMENT_XXX` 宏来实现的。

对于属性 `FontXXX` 或 `PictureXXX`，相应的宏为 `IMPLEMENT_PICTURE_PROPERTY`、`IMPLEMENT_FONT_PROPERTY`，以及 `_HelperPictureProperty` 和 `_HelperFontProperty` 类。

事件是通过接口 `IkButtonEvents` 和自定义接口 `IkButtonEvents` 实现的。

这里还使用了 `HSLModel` 类中的 HSL 色彩模式来计算 `ShadowColor` 和 `HighlightColor`。

```

[kbutton.h]
void _kButton: : Get3DColors (COLORREF clr, COLORREF& clrShadow,
    COLORREF& clrHighlight)
{
    _HSLModel hsl(clr);
    _HSLModel _hsl = hsl;
}

```

```

BYTE lum = hsl.luminance();
hsl.luminance (lum / 3 * 2);
_hsl.luminance (lum + ((240 - lum) / 2));
clrShadow = hsl.rgb();
clrHighlight = _hsl.rgb();
}

```

对于 Visual Basic IDE, 是在 `kICategorizePropertiesImpl` 类中实现 `ICategorizeProperties` 接口的。

```

[categorizeproperties.h]
template <class T>
class ATL_NO_VTABLE _kICategorizePropertiesImpl :
public ICategorizeProperties
{
    STDMETHODCALLTYPE (MapPropertyToCategory) (DISPID dispid, PROPCAT* ppropcat )
    ...
    STDMETHODCALLTYPE (GetCategoryName) (PROPCAT propcat, LCID lcid, BSTR* pbstrName)
    ...
}

```

```

[kbutton.h]
class ATL_NO_VTABLE _kButton :
...
public _kICategorizePropertiesImpl<_kButton>,
...
{
    ...
    BEGIN_CATEGORY_PROP_NAME_MAP (_kButton)
    CATEGORY_PROP_NAME (1, _T ("Picture"))
    CATEGORY_PROP_NAME (2, _T ("Color"))
    END_CATEGORY_PROP_NAME_MAP ()
    BEGIN_PROPERTY_TO_CATEGORY_NAME_MAP (_kButton)
    PROPERTY_TO_CATEGORY (dispidTextPush, PROPCAT_Text)
    PROPERTY_TO_CATEGORY (dispidPictureDisabled, 1)
    ...
    END_PROPERTY_TO_CATEGORY_MAP ()
    ...
}

```

KSet 源代码参见 01\02\source, 其演示程序参见 01\02\demo。

程序 3 弹性按钮

弹性按钮的控件类似于一个 Macintosh 控件, 它继承了标准按钮类, 包括了普通按钮的功能, 我们一般可以把它用作复选框或单选框。这个控件的外观是由一个可以改变状态的按钮图像和可以改变的文字组成的, 当单击按钮时 (文字或左边的圆钮) 文字可以随之改变。参见图 1.4。