



西南交大松岗电脑系列丛书

Visual C++ 2.X MFC Windows 程序设计

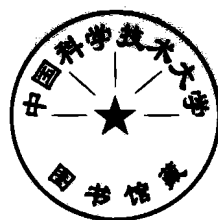
吴锡修 林正敏 编著 方旭明 改编 王倩 审校



西南交通大学出版社

Visual C++ 2.X MFC Windows 程序设计

吴锡修 编著
林正敏
方旭明 改编
王 倩 审校



西南交通大学出版社

出 版 声 明

本书繁体字版本原名为《Visual C++ 2.X MFC 视窗程式设计》，由台湾松岗电脑图书资料股份有限公司出版，版权归松岗公司所有。

本书中文简体字版本由松岗公司授权西南交通大学出版社出版、发行。本书封底皆贴有防伪标签，无标签者即为盗版出版物，不得销售。

Visual C++ 2.X MFC Windows 程序设计

吴锡修 林正敏 编著 方旭明 改编 王 倩 审校

责任编辑 王 明 封面设计 郑 宏

*

西南交通大学出版社出版发行

(成都 二环路北一段 610031)

郫县印刷厂印刷

*

开本:787×1092 1/16 印张:23.875

字数:576千字 印数:1-5000册

1996年8月第1版 1996年8月第1次印刷

ISBN 7-81022-930-3/T·172

定价:38.00元

Trademarks

All brand names and product names used in this book are trademarks, registered trademarks, or trade names of their respective holders. Unalis is not associated with any product or vendor mentioned in this book.

版权声明

本书中所提到的商标,均属于其合法注册之公司所有。

书中图表有关术语对照表

本书原版(中文繁体字版)由台湾松岗电脑图书资料股份有限公司出版。由于海峡两岸对计算机科技术语的用语不同,因此,在改编本书原版、出版中文简体字版时,对正文中的术语对照相应的中文简体版软件进行了改写。但由于书中的图形均采用扫描制版方式,故其中文字仍为繁体字,且专业术语亦未改写。为方便读者查对,现将图表中有关术语与文中所用术语对照如下:

台湾版术语

程式
檔案
檢視
視窗
函式
訊息
回圈
專案
位元圖
圖樣
物件
視域
作業
滑鼠
功能表
復式盒
行
列

大陆版术语

程序
文件
浏览
窗口
函数
消息
循环
项目
位图
图标
对象
视口
操作
鼠标
菜单
组合框
列
行

序

自从 Windows 推出后,由于画面细腻、操作方便而大受欢迎,Windows 应用程序也跟着水涨船高,许多知名的 DOS 应用程序纷纷移植到 Windows 环境。早期以 SDK 设计 Windows 应用程序,真的可用筚路蓝缕来形容,除了 Dialog Editor 能较方便地产生对话框资源外,其余都得程序员胼手胝足地去完成。而且您必须熟知 Windows 内部的运行情况,学习起来相当吃力,也因此,不论 Microsoft 或 Borland 公司后续推出的开发系统都强调简单易学、功能完善。

Visual C++ 是 Microsoft C/C++ 7.0 之后推出的新一代程序开发工具,包括下列五种核心工具:

- **MFC Library** : 提供可重复使用的类函数库。
- **Workbench** : 集编辑器、调试器、浏览器于一身的集成环境。
- **AppWizard** : 可自动产生窗口基本框架程序。
- **ClassWizard** : 类管理程序,并可自动产生消息处理函数框架。
- **AppStudio** : 可建立、编辑应用程序的各项资源。

它不仅继承 C++ 特性,同时具备 Visual Programming 语言及程序产生器的概念,使得 Windows 程序设计更直观,对系统开发者而言,Visual C++ 无疑是最佳的 Windows 系统开发工具。

本书针对 MFC Windows 程序设计,以深入浅出方式带您一窥 Visual C++ 惊人的奥秘,介绍目前流行的 OOD、Visual Programming、Event Driven、FrameWork 等新概念。

在内容安排方面,强调循序渐进及完整性。第一章综合性地介绍 Windows 基本特性及 Windows 应用程序的基本概念,并简要地比较 SDK 程序与使用 Visual C++ MFC 程序的差异。第二章着重在 Visual C++ 集成环境的介绍,并以实例引导方式说明如何使用集成环境快速建立应用程序框架。第三章则介绍 Visual C++ MFC 程序的基本框架,并实际剖析 AppWizard 所产生的框架程序内容。第四章之后则以框架程序为基石,逐步扩充应用程序的各项功能,内容涵盖 Windows 程序的菜单、图形/文本输出、工具栏、状态栏、工具箱、后缓和自制对话框、文件管理和打印、剪贴板应用以及帮助文件的撰写与集成。

另外,对于所使用的函数皆以函数原型来详细地说明函数功用及参数用法,以免除一面看书还要一面查询函数用法的困扰。

最后要感谢技术学院蔡新民教授的指导、家人的支持及松岗公司发行部的协助,若本书能燃起您 Windows 程序设计之火苗,那将是笔者最衷心的期望。

吴锡修 林正敏

1995.06.16 于台中

目 录

第一章 从 DOS 到 Windows

1.1 认识 Windows	1
1.2 窗口的基本组件与鼠标的操作	3
1.3 Windows 与 DOS 环境的比较	5
1.4 Windows 与应用程序间的关系	9
1.5 Windows 应用程序的动作流程	10
1.6 Windows 的消息系统	11
1.7 从 C/SDK 到 Visual C++/MFC	13
1.8 易学、易用的 Visual C++	17
1.9 Windows 应用程序变量与函数命名规则	18
1.9.1 函数的命名规则	18
1.9.2 变量的命名规则	19
1.10 关于本书	19
1.11 自我评测	21

第二章 应用程序开发流程与相关工具

2.1 初识 Visual C++ 2.X	23
2.2 Visual C++ 集成环境	25
2.1.1 工作平台 (Visual Workbench)	25
2.1.2 应用程序资源工作室 (AppStudio)	26
2.1.3 Framework 的核心工具——AppWizard	28
2.1.4 类总管——ClassWizard	29
2.2 Visual Workbench 的基本操作	29
2.3 应用程序开发流程	31
2.3.1 使用 AppWizard 建立应用程序框架	32
2.3.1.1 设定应用程序窗口类型	33
2.3.1.2 设定数据库支持能力	33
2.3.1.3 设定 OLE 及 OLE Automation 支持能力	33
2.3.1.4 设定应用程序基本功能	35
2.3.1.5 项目文件格式及函数库程序链接方式	37
2.3.1.6 修改类名称及相关项目名称	37
2.3.2 扩展应用程序	41

2.3.3 编译、链接、执行与调试.....	42
2.3.4 管理程序代码.....	44
2.4 自我评测.....	44

第三章 基本框架程序剖析

3.1 应用程序的基本框架.....	46
3.2 应用程序对象.....	48
3.3 文件模板.....	51
3.4 主框架窗口对象.....	55
3.5 文件与文件框架窗口.....	60
3.6 视口对象.....	63
3.7 SDI 程序与 MDI 程序的差异	66
3.7.1 主框架窗口类的差异.....	66
3.7.2 SDI 与 MDI 文件模板的差异	67
3.7.3 窗口应用程序组成对象建立顺序的差异.....	68
3.8 自我评测.....	68

第四章 基本文本及绘图输出

4.1 Windows 的屏幕输出控制	69
4.1.1 Device Context 与 Display Context 的概念.....	69
4.1.2 Windows 的坐标系统	70
4.2 文本输出控制.....	74
4.3 基本绘图函数.....	78
4.3.1 描点.....	78
4.3.2 绘线.....	79
4.3.3 绘椭圆、圆	81
4.3.4 绘矩形、圆角四边形	82
4.3.5 绘多边形.....	84
4.3.6 Pie 图形与 Chort 图形	84
4.4 画面维护消息——WM_PAINT 的处理	86
4.5 自我评测.....	87

第五章 输入消息及其处理函数

5.1 Windows 的输入消息	89
5.1.1 键盘消息.....	89
5.1.2 鼠标消息.....	92
5.1.3 字符消息.....	94
5.1.4 定时器消息.....	98
5.1.5 滚动条消息.....	99
5.1.6 菜单消息	101

5.2 使用 ClassWizard 建立消息处理函数框架	102
5.3 自我评测	105

第六章 再谈 Device Context 的设定

6.1 解读 Device Context 黑匣子	106
6.2 文本属性控制	108
6.2.1 文本前景及背景颜色控制	108
6.2.2 字型、字体的变化	110
6.3 绘图效果控制	116
6.3.1 更换画笔、画刷	116
6.3.2 绘图效果的控制	121
6.4 自我评测	124

第七章 菜单制作、维护

7.1 利用 AppStudio 设计应用程序菜单	125
7.2 再谈菜单消息处理	127
7.3 菜单选项状态的设定与标记的使用	130
7.3.1 菜单选项的状态设定	131
7.3.2 标记的使用	132
7.4 修改原先设计的菜单	135
7.4.1 增加新菜单选项	135
7.4.2 修改原选项的内容	137
7.4.3 删除菜单选项	138
7.4.4 更换整个菜单	138
7.4.5 制作图形式菜单选项	140
7.5 制作浮动式菜单	142
7.6 自我评测	145

第八章 工具栏、状态栏与工具箱的设计

8.1 工具栏的设计	147
8.1.1 建立工具栏	147
8.1.2 动态更换工具栏中的命令按钮	152
8.1.3 改变工具栏的命令按钮风格	154
8.1.4 在工具栏中嵌入组合框控制项	156
8.2 状态栏	161
8.2.1 建立状态栏	161
8.2.2 更换状态栏指示区面板风格	163
8.2.3 在状态栏显示信息	164
8.3 设计工具箱	165
8.3.1 建立工具箱类	166

8.3.2 建立工具箱按钮图形位图资源	167
8.3.3 改写画面维护函数——DoPaint()	167
8.3.4 改写命令按钮识别函数——HitText()	169
8.3.5 改写 CalcFixedLayout() 函数	271
8.3.6 建立工具箱对象	172
8.3.7 更换工具箱命令按钮风格	173
8.4 自我评测	176

第九章 后缓对话框的应用

9.1 认识对话框	177
9.2 文件存取对话框	178
9.3 颜色设定对话框	184
9.4 字型设定对话框	187
9.5 打印/打印机设定对话框	191
9.6 字符串搜索/替换对话框	194
9.7 自我评测	198

第十章 消息框与自制对话框

10.1 消息框	200
10.2 模式对话框的设计	202
10.3 非模式对话框的设计	208
10.4 对话框的常用函数	211
10.5 设计图形按钮	217
10.6 集成 Visual Basic 的控制项	218
10.7 自我评测	222

第十一章 文件存取

11.1 文件基本操作流程	223
11.1.1 打开文件	225
11.1.2 存取数据	229
11.1.3 关闭文件	230
11.1.4 取得文件状态	231
11.2 Serialization	232
11.3 CArchive	234
11.4 使用 Collections 管理数据对象	236
11.4.1 表集合	237
11.4.2 数组集合	240
11.4.3 映射集合	243
11.4.4 如何使用集合类	245
11.4.4.1 直接使用后缓集合类	245

11.4.4.2 继承和扩展后缓集合类	250
11.4.4.3 使用模板建立新的集合类	250
11.5 自我评测	251

第十二章 增强打印能力

12.1 AppWizard 如何制作缺省打印功能	252
12.2 打印结构	253
12.3 打印信息	257
12.3.1 CPrintInfo 公有成员函数	257
12.3.2 CPrintInfo 公有数据成员	258
12.3.3 CPrintInfo 对象的初始状态	258
12.4 细说分页打印	259
12.5 预览功能	262
12.5.1 预览流程	263
12.5.2 设定预览页数	266
12.6 屏幕 DC 与打印机 DC 的比较	267
12.7 自我评测	269

第十三章 剪贴板

13.1 剪贴板的基本概念	270
13.1.1 剪贴板的数据存储方式	270
13.1.2 剪贴板的操作	271
13.2 剪贴板的数据格式	272
13.3 常用的剪辑函数	273
13.3.1 打开剪贴板	273
13.3.2 关闭剪贴板	274
13.3.3 清除剪贴板	274
13.3.4 复制数据到剪贴板	274
13.3.5 由剪贴板取数据	274
13.3.6 检查剪贴板的数据格式	275
13.4 剪辑数据的流程	275
13.4.1 数据复制流程	275
13.4.2 数据粘贴流程	276
13.5 存储器管理简介	276
13.5.1 使用存储器的三部曲	277
13.5.2 配置存储空间	277
13.5.3 重配存储空间	278
13.5.4 锁定/解锁存储空间	278
13.5.5 归还存储空间	279
13.5.6 最常用的组合全局存储器属性——GHND	279

13.6 简易剪贴方法	279
13.6.1 复制文本数据到剪贴板.....	279
13.6.2 识别剪贴板是否内含文本格式的数据.....	281
13.6.3 从剪贴板中取用文本数据.....	281
13.6.4 复制位图到剪贴板.....	282
13.6.5 从剪贴板取出位图.....	283
13.6.6 Metafile 的转移	285
13.6.6.1 制作元文件.....	285
13.6.6.2 执行元文件.....	286
13.6.6.3 复制元文件到剪贴板.....	286
13.6.6.4 粘贴元文件.....	288
13.7 转移多重数据项	288
13.8 延迟传送(Delay Rendering)	290
13.9 使用私有数据格式	291
13.9.1 CF_DSP...类型格式	292
13.9.2 处理 CF_OWNERDISPLAY 类型数据.....	292
13.9.3 注册新的剪贴板数据格式.....	294
13.10 自我评测	294

第十四章 画龙点睛——Windows Help 制作

14.1 WinHelp 的浏览功能	296
14.2 联机帮助文件制作流程	297
14.2.1 收集相关资料.....	297
14.2.2 规划帮助文件结构.....	298
14.2.3 编写帮助文件.....	298
14.2.4 建立帮助文件项目与编译测试.....	299
14.3 图文集成	302
14.4 精益求精——Context Sensitive Help	303
14.5 将 Help 功能加入应用程序	304
14.6 MFC 的 Help 命令处理方式	305
14.7 自我评测	306
附录 A C++ 语言快速指引	307
附录 B QuickWin 应用程序设计	324
附录 C Visual C++ MFC 类层次	342
附录 D Visual Workbench 菜单操作指引	346
附录 E 本书所使用的 MFC 函数快速指引	350
附录 F Windows 常用的数据结构	360
附录 G 中英文对照	365
附录 H 参考书籍	369

第一章 从 DOS 到 Windows

近年来,Windows 在 PC 界造成一股旋风,所到之处风行草偃,许多 DOS 知名软件纷纷投入 Windows 怀抱,推出 for Windows 版本。的确,当别人使用鼠标在声光效果一流的环境下愉快的作业,而您还在 DOS 环境下利用方向键慢慢地爬时,您就不难了解大家为何喜欢 for Windows。而从市场销售量来看,也的确如此,它不但荣获 1993 年 BYTE 读者票选的年度最佳软件产品,更是 PC Magazine 每月畅销软件排行榜之常客,可见其魅力十足。大势所驱,每位程序员都希望能撰写 Windows 应用程序,进军 Windows 世界,到底 Windows 为何有此能耐?而又要如何撰写 Windows 应用程序?

Visual C++ 是微软公司继 Microsoft C/C++ 7.0 之后推出的一套 Windows 应用程序员开发工具,它有非常方便的集成环境,提供基础类函数库,并可自动产生 Windows 应用程序的框架程序,让您能快速开发 Windows 应用程序,是目前最被看好的应用软件开发工具之一。

本章里,我们将就以下主题作进一步介绍:

- 简介 Windows 的发展及特色。
- 说明 Windows 的基本组件的作用与鼠标的操作方式。
- 说明 Windows 应用程序的执行流程。
- 剖析 Windows 应用程序的消息系统。
- 简介 Visual C++, 并比较 Visual C++ MFC 所建立的 Windows 应用程序与 SDK 程序的差异。

1.1 认识 Windows

Windows 系统是微软(Microsoft)公司在 IBM PC 或兼容机型,所推出的一个具有多任务能力(Multitasking)、支持多媒体、提供友好方便的图形化用户界面(GUI)的窗口式操作环境。由于在 Windows 下所开发的应用程序都具有一致的外观及相同的操作方式,因此,比传统 MS-DOS 环境下的程序更容易学习和使用。此外,还提供各种数据交换协议,使得不同应用软件间能轻易地进行数据交换,有效地解决了软件间的数据集成问题。

Windows 的前身可追溯至 1970 年代中期,施乐公司(Xerox)的“Palo Alto Research Center (PARC)”率先以窗口作为人机界面,原始动机在突破传统用户界面亲和力不够,并且可让用户方便地观察各窗口中的信息,但当时这种方式仅限于内部使用,并未公开发表。

到了 1983 年 1 月,苹果电脑公司将窗口的概念引入 Lisa,并于次年 1 月推出 Macintosh,由于窗口界面的亲和性得到相当的回响,迫使微软公司不得不开始投注人力于窗口界面的研究与开发,以稳固微软在软件市场的地位。最早版本是 1985 年 11 月间推出的 Windows 1.02,

初期由于用户界面设计上尚不成熟,系统效率也不理想,一般评价并不高,一直到 Windows 3.0 才大放异彩,至今已更新至 Windows 3.11(3.2——改编者注)版,而目前深受大家瞩目的 Chicago 将在今年正式与大家见面,称之为“Windows 95”,而且中文版首度与英文版同步发行,这点颇值称许,也算是对大家尊重知识产权的回馈吧!

从程序员的观点,Windows 提供了一套健全的后援程序库,让设计者很轻易地就能设计出具有菜单、滚动条、对话框及友好人机界面等优点的应用软件。同时,程序员在设计软件时,可以完全不考虑键盘、鼠标、显示器、打印机、系统定时器以及 RS-232 通讯接口等设备的厂牌、机型,而以一种与设备无关(Device-independent)的方式看待,换言之,您在 Windows 下开发的程序可轻易地移植到不同硬件配备的 PC 窗口环境下执行。

此外,Windows 中所有应用程序可分别用不同的图标(Icon)来表示,您可将这些图标依程序的性质加以规划分组,置于不同的文件夹中,用户只要在图标上双击鼠标左键即可选取并执行所代表的程序。而通过“文件管理器”的 Drag-and-Drop 功能,文件复制、删除、移动……等操作变得非常轻松愉快,再者,利用“控制台”则可让您自行规划自己的窗口桌面环境。

在 Windows 95 中新的系统管理程序为“Explorer”,也就是图 1.1 桌面环境下方的“工作条”,您可以随您喜好将它置于桌面底部、顶端、左侧或右侧位置,还可利用“文件管理”重新规

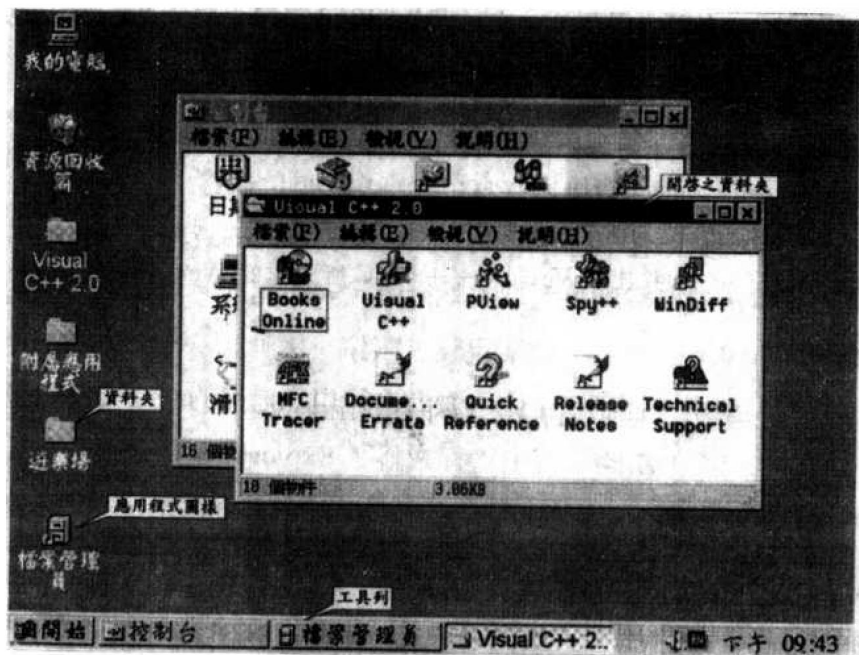


图 1.1 Windows 桌面环境

划开始菜单的内容,有了它的帮助,要在一堆重叠窗口中找寻所要的工作窗口将是轻而易举。但若您习惯于 Windows 3.X 的操作方式,怀念那朝夕相处的“程序管理器”,那么可直接修改 Windows 的 system.ini 文件[boot]段落的 shell 设定值,重新指定 progman 为系统管理程序,例如:

[boot]

'shell=Explorer.exe

←取消原先系统设定

shell=progmán.exe

←指定程序管理器为系统管理程序

keyboard.drv=keyboard.drv

事实上,您也可由“Explorer”开启“程序管理器”,如此可同时享受二种不同管理程序之便利。

1.2 窗口的基本组件与鼠标的操作

在学习如何设计 Windows 应用程序之前,首先让我们说明 Windows 操作的方式及术语,并浏览典型 Windows 程序的外观,以了解窗口基本组成元件的位置与功能;这将有助于您熟悉 Windows 应用程序的操作,并体会 Windows 应用程序每一组件所扮演的角色及功能,往后撰写 Windows 程序时,您就能掌握该赋予应用程序怎样的外观。

虽然 Windows 也可通过键盘来操作,但是您必须熟记每一命令所对应的按键组合,这对您的大脑无疑是一种虐待,而且真要使用键盘来移动光标也是非常地不便;所以最好的方式便是使用鼠标操作 Windows。一般使用鼠标可产生下列四种不同的动作:

• 移动(Move)

纯粹移动鼠标来变换光标位置,在移动过程中没有任何鼠标按键动作。

• 单击(Click)

在鼠标静止时(偏移不超过额定范围),快速按一下鼠标键,用来“选定”某对象。例如:从菜单中执行某菜单选项、从按钮组里按下某一按钮、从列表框中挑选某数据项……等等。

Click 也常配合 Ctrl 或 Shift 键使用。Ctrl+Click 常用来做多重选择,例如在文件管理的目录窗口利用 Ctrl+Click 来挑选一个以上的文件;而 Shift+Click 则是与 Click 配合,用来选择指定范围内的所有数据项,例如在文件管理器的目录窗口先以 Click 选择某一文件,再 Shift+Click 另一文件,即可选定二次单击间的所有文件。

• 双击(Double Click)

在鼠标静止时(偏移不超过额定范围),于规范的时间内快速连按两下鼠标键,意谓着“确定”和“执行”。例如:在代表应用程序的图标上快速按两次鼠标左键来启动该应用程序、于开启文件对话框的文件名上快速连按两次鼠标键以开启该文件……等等。

• 拖动(Drag)

按着鼠标键不放,移动鼠标到适当位置再放开鼠标键,常用来改变对象的大小和位置,或是指定选择区。例如:拖动窗口边框来改变窗口大小、拖动窗口标题栏来移动窗口位置、在窗口客户区中使用拖动来选定拷贝/剪贴区域……等。

在熟悉鼠标操作后,接下来我们对照图 1.2 说明窗口的基本组成元件及其功能,分述如下:

- 系统菜单图标 (System-Menu Icon)

位于窗口的左上方,用来开启系统菜单。在此位置单击鼠标左键可开启系统菜单,而若直接在此图标上双击鼠标左键,则可立即结束此应用程序。

除了利用系统菜单图标来开启系统菜单外,在窗口标题栏任一位置单击鼠标右键也同样可开启系统菜单。

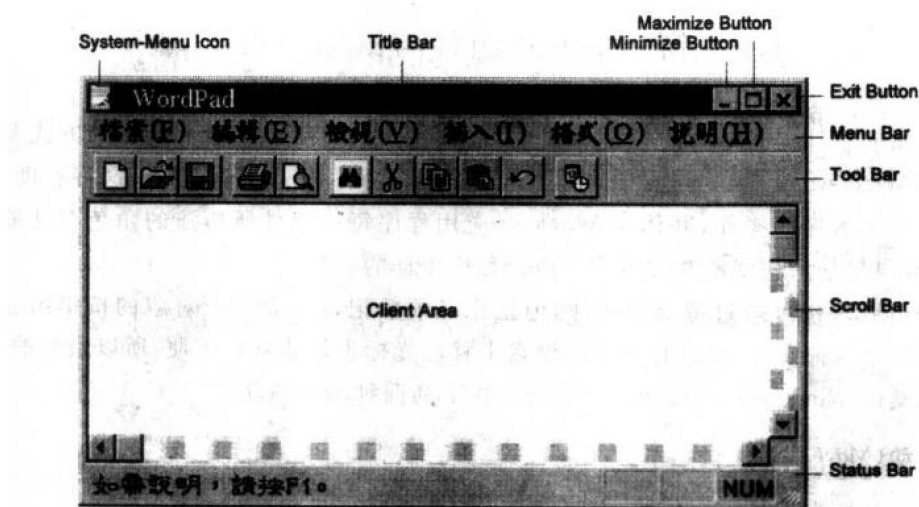


图 1.2 窗口基本组件

- 标题栏 (Title Bar)

位于窗口顶端,主要的功能为显示应用程序的名称或目前正在编辑的文件名称。另外,可在此区域中拖动鼠标左键来移动窗口位置,而若直接在标题栏上双击鼠标左键则可使窗口最大化或使之从最大化还原。

- 窗口边框 (Border)

为窗口边界,您可在窗口边框上拖动鼠标左键来改变窗口的大小。

- 最小化按钮 (Minimize Button)

位于窗口的右上方,单击此按钮可用来将窗口最小化。所有最小化的应用程序或文件夹会自动陈列在工作条上。

- 最大化按钮 (Maximize Button)

在最小化按钮的右边,单击此按钮可将窗口放大到和屏幕一样大。

- 关闭按钮 (Close Button)

相当于执行系统菜单中的关闭 (Close) 命令,会立即结束应用程序。

- 菜单条 (Menu Bar)

位于标题栏的下方,在菜单选项(Menu Item)上单击鼠标左键则会执行该选项命令;而若该选项为下拉式菜单(Pull-down Menu)的名称,则选择该菜单选项后会打开另一子菜单。

- 滚动条(Scroll Bar)

可分成垂直滚动条及水平滚动条两种,分别位于窗口的右边及下方,可用来滚动窗口内容。

- 工具栏(Tool Bar)

一般置于菜单条下方,主要将用户常用的菜单选项以命令按钮的方式陈列在工具栏中,以免除要经常打开菜单的不便。

- 状态栏(Status Bar)

位于窗口最下方,用来显示应用程序目前的基本状态,如光标当前位置、键盘的 Insert、Caps Lock 及 Num Lock 等按键的状态……等。

- 客户区(Client Area)

在窗口中除了上述对象所占据的区域外就是该窗口的客户区,它是应用程序最重要的输出区域。

若在客户区单击鼠标右键,则可打开编辑专用的弹出式菜单(Popup Menu),一般应用程序会将编辑操作常用的命令项置于此菜单中,以方便编辑客户区数据。

1.3 Windows 与 DOS 环境的比较

何以 Windows 会比 DOS 来得吸引人?当然不只是它外观上所呈现的美感而已,更重要地在于它的内涵,除此之外,近年来硬件和软件技术的提高也功不可没。在硬件方面,从 8088 演进到 Pentium,一次能处理的信息量,由 8 位进展到 64 位,不但 CPU 处理的速度愈来愈快,而且存储器的容量也比以前大得太多了,同时电脑的外部设备也进步得相当惊人。而软件方面,在设计观念上亦进步、成熟许多,诸如虚拟存储器、面向对象的设计(OOD)、多任务或分时操作环境……等概念的引入,都有助于大系统的设计。

只可惜 PC 在硬件方面虽然已有能力完成软件观念的梦想,但反观最重要的操作系统 DOS 发展就逊色不少,当初 DOS 5.0 以能控制 1 M 以上的 64 K 的存储器空间(HMA)为卖点,而最新的 DOS 6.2 也仅是加强其它工具程序,在存储器管理上则未有新的突破。然而,我们都知道就连 80286 CPU 在保护虚拟寻址模式中的实际寻址能力即可达 16 MB,更不用谈目前的 Pentium 个人电脑了。面对这种情况,您是否也曾有过这样的疑问:多出来的存储器空间要如何管理呢?虽然书店有不少书籍探讨 EMS、XMS 存储器的使用,但将这类问题丢给程序员毕竟不是 OS 该有的表现,正因如此,促使大家将焦点转移到存储器管理表现不俗的 Windows 身上,下面就来探讨窗口操作环境有哪些是 DOS 环境所不及的。

- 一致性的图形用户界面