

Windows 95

特色编程

李真文 编著



北京航空航天大学出版社

Windows 95 特色编程

李真文 编著

北京航空航天大学出版社

内 容 提 要

Windows 95 日益普及,掌握编写支持 Designed for Windows 95 徽标的应用软件的开发人员亦是身价百倍,本书试图使您成为这方面的人才。

本书从公共控件、OLE 2.0、在线帮助和安装向导四个方面揭示了 Windows 95 应用软件的特色,并基于 Visual C++ 4.0(包括 Visual Basic 4.0)全部实现了这些特色。

本书讨论的主题不仅适合于 Windows 95 操作系统,同样适合于与 Windows 95 同期推出的 Windows NT 3.51、Windows with Win32s 1.3 及其更高版本。同时,对于开发环境采用 Borland C++ 5.0 和 Delphi 2.0 的读者,也有“它山之石,可以攻玉”的参考价值。本书的例程软盘共 5 张,购买方法见书末。

图书在版编目(CIP)数据

Windows95 特色编程/李真文编著.-北京:北京航空航天大学出版社,1997.1
ISBN 7-81012-683-0

I. W... II. 李... III. 窗口软件, Windows 95-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(96)第 19250 号

- 书名:Windows 95 特色编程
- 作者:李真文
- 责编:赵延永
- 责校:陈 坤
- 出版:北京航空航天大学出版社
- 印刷:北京朝阳科普印刷厂
- 发行:新华书店总店北京发行所
- 经售:全国各地书店
- 开本:787×1092 1/16
- 印张:33.25
- 字数:851.2 千字
- 印数:5000 册
- 版次:1997 年 1 月第一版
- 印次:1997 年 1 月第一次印刷
- 书号:ISBN 7-81012-683-0/TP·230
- 定价:45.00 元

前 言

随着计算机硬件性价比不断提高,我国普通计算机用户购买高档微机(如奔腾)已成为现实。因此,以较高硬件配置为基础的 Windows 95 操作系统正以更加友好的用户界面、更高的性能以及强大的通信联网能力不断占领国内台式机操作系统市场,越来越多的用户接受了 Windows 95。而我们的软件发展还不能与硬件升级同步,在 486、586 等高档微机上仍然运行那些在 286、386 上就能运行的 DOS 和简单的 Windows 软件,远不能发挥高档微机的潜能,也就是出现了所谓“大马拉小车”的怪现象。适时掌握从 DOS、Windows 3. x 平台升级到 Windows 95 的技术,编写支持 Designed for Windows 95 徽标的应用程序,是每个应用软件开发者都面临的机遇和挑战。可以这样说,掌握了 Windows 95 编程,就意味着在工作中多一份成功和希望。

何谓 Windows 95 特色? 什么样的应用软件才能称为 Designed for Windows 95? 笔者参考了有关 Windows 95 的最新资料,总结出了对于国内大多数应用软件开发者来讲,Windows 95 应用软件的基本特色应包括公共控件、OLE 2. 0、在线帮助和安装向导四个方面。当然,Windows 95 应用软件特色还包括即插即用、Win32s、外壳、多线程、通信与联网等等,但考虑到这些属于“高级话题”,本书不作介绍,感兴趣的读者可以参考相关文献。

说到“编程”,人们常常会与那些逻辑性强、晦涩难懂的一行行程序代码联系在一起,从而使很多人望而却步。事实上,读者们不必紧张。本书所指的“编程”,按段永朝先生在文献[1]中的观点,实乃“信息制作”,就是在真正的“编程”师傅制作的平台 Visual C++ (Visual Basic) 上进行一些操作,然后辅以少量的代码,就可以完成大多数的设计任务。也许您会问:“真是这样简单吗?”是的,软件发展到今天,不仅仅是用户界面更加友好,功能更加强大,更重要的是软件的生产效率更高。如今 RAD(快速应用程序开发)工具纷至沓来,从 Microsoft 的 Visual Basic 和 Visual C++、Borland 的 Borland C++ 和 Delphi,到 Powersoft 的 PowerBuilder……这些工具使用起来简单方便,只要掌握每个工具特定的“制作信息”的方法,就可以完成许多复杂的设计任务。正因为如此,本书没有对读者的背景知识作过多的假设,使用过 Windows 95,精通 C++ 即可(临时抱佛脚也行)。当然,有一个前提是必需的:敢于尝试、不怕失败、对软件有相当的悟性。这些实际上也是成功人士所必须具备的基本素质。如果通过本书的学习,使读者在这方面有所增强,那将使笔者倍感欣慰。

在 RAD 工具异彩纷呈的今天,应该选取哪一个工具来开始在 Windows 95 环境下进行“信息制作”呢? 笔者比较了 for Windows 95 的主要开发工具,包括 Microsoft 的 Visual Basic 和 Visual C++、Borland 的 Borland C++ 和 Delphi,发现 Visual C++ 4. 0 对 Windows 95 特征实现得最充分,随软件有关 Windows 95 的文档和例程最丰富,因而选择它作为本书的主要开发环境。同时,在有关 OLE 2. 0 技术中,也辅助地采用了 Visual Basic 4. 0 来演示 OLE 有关技术的不依赖于语言的特性。

在写作过程中,本书的责任编辑、北航出版社的赵延永让笔者考虑 Borland C++ 的读者。

在这里,我想谈一下我使用 Borland International 公司产品的经历。我曾是 Borland 忠实的信徒,其 Turbo C 让我领教了集成开发环境的魅力,而 Borland C++ 的 OWL 又给了我 Windows OOP(面向对象程序设计)全新概念,但是,我最终放弃了 Borland C++。那是在 1994 年年底,在我即将完成研究生学业、在 Borland C++ 3.1 上编写的软件“科学曲线 OLE 服务器”^[2]也即将完成时,我得到了 Borland C++ 4.0 版本,当我看到 OWL 2.0 已提供 OLE 类封装时,非常高兴,决定把在 Borland C++ 3.1 上所编写的近 7 000 行的程序放在 4.0 版上,利用对 OLE 的类封装来优化我所编写的有关 OLE 的程序。遗憾的是,这些程序放在 Borland C++ 4.0 上编译根本通不过,因为 OWL 类库基本上又重新构造了一遍,此时我有一种“被抛弃”的感觉。后来我逐渐得到了 Visual C++ 1.0、1.5、2.0 和 4.0 版本,并在这些平台上编写了一些软件,包括给广州天一自动化系统工程公司编写的“天一多媒体电脑联网报警系统”,以及后来的“实时 CASE 工具”,发现 Visual C++ 的 MFC 高低版本兼容性很好,类层次结构合理,集成环境开发界面优越,例程很多,在线文档丰富……总之,Visual C++ 成了我最喜欢的工具。而且在报纸、杂志上也能看到美国的一些 ISV(独立软件开发商)提供 MFC 类库组件,这时我庆幸及早转向了 Visual C++。

当然,这里绝没有让 Borland C++ 用户放弃 Borland C++ 而追随 Microsoft Visual C++。事实上,工具仅是个人偏好问题,只要自己用得熟练,那就是好工具。当然,有一点还是很重要的,不但会使自己喜欢的工具,还要留意与之竞争的同类工具,并能从同类工具中借鉴自己需要的东西,这才是高明的工具用户。所以,本书对于 Borland International 公司的用户来讲,用于了解 Visual C++ 和 Visual Basic 对 Windows 95 特色的实现也具有一定的参考价值。

编程是实践性很强的活动,只有在电脑上构思、输入代码、调试,再构思、再输入代码、再调试……经过多次反复,才能掌握编程要点。为了充分理解本书的内容,除了掌握每章开始的概念、原理介绍外,需要上机完成每章给出的例程,并设法自己做一遍,调试通过才行。当然,为了提高效率,以及弥补书中没有全部给出例程源代码的缺陷(全部给出是不可能的,那样会使篇幅急剧膨胀),笔者用 InstallShield SE 制作了本书例程的安装向导,共 5 张 3.5 英寸盘,欢迎读者选购。

在本书的写作过程中,得到了北京航空航天大学出版社的大力支持;《中国计算机用户》杂志社的唐秀艳编辑一直极力鼓励笔者将编写具有 Windows 95 特色的应用程序的心得写出来,并提供在该杂志上发表若干篇文章的宝贵机会;西南政法大学法律系的曾丽同学在本书写作初期也表示了相当的关切,其给笔者的来信使笔者学会了如何把本书写得更风趣一些;高其林先生在写作初期给笔者提供了全封闭的周末写作环境,让笔者能够抛开一切专心编程和写作。另外,应该提到本书列出的参考文献的所有作者,是他们的辛勤劳动铺就了笔者进入 Windows 95 的阶梯。对上述以及其他许多曾给予笔者学习和生活方便的人们,在此一并表示衷心的感谢!

李真文

1996 年 12 月

目 录

第一部分 Windows 95 与 Visual C++4.0

第1章 Windows 95 特色总览	(2)
1.1 引言	(2)
1.2 用户界面	(2)
1.2.1 工作台	(3)
1.2.2 任务条	(3)
1.2.3 我的电脑	(4)
1.2.4 网上邻居	(4)
1.2.5 回收站	(4)
1.2.6 Windows 资源管理器	(4)
1.2.7 快捷键	(5)
1.2.8 属性表	(6)
1.3 体系结构	(6)
1.3.1 抢占式多任务操作系统	(6)
1.3.2 VxD	(7)
1.3.3 32位文件系统体系结构	(8)
1.3.4 内存管理	(8)
1.4 即插即用	(9)
1.5 系统管理	(11)
1.5.1 注册表	(11)
1.5.2 用户管理	(12)
1.5.3 系统管理	(12)
1.5.4 网络管理	(13)
1.6 稳健性	(13)
1.6.1 局部重新引导	(13)
1.6.2 进程结束时 VxD 的线程清除机制	(14)
1.6.3 每个线程状态的跟踪	(15)
1.6.4 虚拟设备驱动程序参数的有效性检查	(15)
1.7 设备支持	(15)
1.8 打印	(16)
1.8.1 32位打印子系统	(17)
1.8.2 对 MS-DOS 应用程序打印支持的改进	(17)
1.8.3 支持延迟打印	(17)
1.8.4 支持图像颜色匹配	(18)
1.8.5 安装和配置打印机	(18)
1.9 联网	(21)

1.10	通信	(22)
1.11	Microsoft Exchange	(22)
1.11.1	Windows 信报子系统:MAPI 1.0	(23)
1.11.2	Microsoft Exchange	(24)
1.11.3	从桌面计算机发送和接收传真	(25)
1.12	Microsoft Network	(26)
1.13	移动式计算	(26)
1.14	多媒体	(27)
1.15	国际化支持	(28)
1.16	小结	(29)
第2章 Microsoft Visual C++4.0 使用琐记		(30)
2.1	引言	(30)
2.2	安装 Microsoft Visual C++4.0	(30)
2.2.1	设置在线书籍在硬盘上的路径	(30)
2.2.2	设置例程在硬盘上的路径	(31)
2.3	AppWizard	(32)
2.3.1	使用 AppWizard 创建应用程序	(33)
2.3.2	AppWizard 的中文支持	(33)
2.4	ControlWizard	(34)
2.5	工程工作空间	(35)
2.5.1	低版本的工程文件转换	(35)
2.5.2	一切都是“树”	(35)
2.6	ClassView	(36)
2.7	ResourceView	(37)
2.7.1	资源及资源编辑器	(38)
2.7.2	在资源编辑器中激活 ClassWizard	(42)
2.8	FileView	(43)
2.9	InfoView	(44)
2.9.1	F1 键	(45)
2.9.2	索引	(45)
2.9.3	全文搜索	(45)
2.9.4	例程	(46)
2.10	ClassWizard	(46)
2.10.1	ClassWizard 属性表	(47)
2.10.2	使用 ClassWizard 增加函数	(48)
2.10.3	使用 ClassWizard 增加变量	(48)
2.10.4	使用 ClassWizard 增加代码	(49)
2.11	WizardBar	(49)
2.12	组件廊	(50)
2.12.1	在工程中插入组件	(50)
2.12.2	创建自己的组件	(52)
2.12.3	与他人共享组件	(52)
2.13	使用调试器	(53)

2.13.1	使用多个调试窗口	(53)
2.13.2	使用 Just-in-Time 调试	(54)
2.13.3	控制和管理断点	(54)
2.13.4	调试 DLLs	(55)
2.13.5	调试 OLE 应用程序	(55)
2.13.6	调试远程程序	(56)
2.14	发行应用程序	(56)
2.14.1	构造应用程序的发行版本	(56)
2.14.2	再发行 DLL 文件	(57)
2.15	小结	(58)
第 3 章	MFC 4.0: Windows 的 C++ 应用程序框架	(59)
3.1	引言	(59)
3.2	MFC 综述	(59)
3.2.1	发展历史	(59)
3.2.2	MFC 4.0	(60)
3.2.3	客户反馈驱动的改进	(61)
3.2.4	MFC 转换工具	(61)
3.3	MFC 的移植性	(61)
3.3.1	多平台	(61)
3.3.2	申请 MFC 许可	(62)
3.3.3	工业标准的演化	(62)
3.4	MFC 4.0 的新特征	(63)
3.4.1	Windows 公共控件类	(63)
3.4.2	完全的 OLE 控件支持	(63)
3.4.3	用 DAO 进行直接数据库访问	(63)
3.4.4	新的公共对话框	(63)
3.4.5	线程同步对象	(64)
3.4.6	简单 MAPI 支持	(64)
3.4.7	WinSock 支持	(64)
3.5	MFC 3.0 引入的特征	(64)
3.5.1	增强的工具条	(64)
3.5.2	小型边框窗口	(66)
3.5.3	属性表	(66)
3.5.4	Windows 95 用户接口	(67)
3.5.5	Win32 支持	(67)
3.5.6	多线程	(68)
3.5.7	Unicode 支持	(68)
3.5.8	共享 32 位 DLLs	(69)
3.6	C++ 语言语法支持	(69)
3.6.1	C++ 模板	(69)
3.6.2	C++ 异常	(70)
3.7	数据库类	(71)
3.7.1	数据库引擎类	(71)

3.7.2	记录视	(71)
3.8	OLE 类	(72)
3.9	MFC 框架类	(72)
3.9.1	命令	(72)
3.9.2	文档和视窗	(73)
3.9.3	打印和打印预览	(73)
3.9.4	对话框数据交换和有效化	(74)
3.9.5	上下文敏感帮助	(74)
3.10	高级抽象	(74)
3.10.1	报表视窗	(75)
3.10.2	编辑视窗	(75)
3.10.3	滚动视窗	(75)
3.10.4	切分窗口	(75)
3.10.5	控制条	(75)
3.11	兼容性	(76)
3.11.1	OLE 1.0 支持	(76)
3.11.2	VBX 1.0(16 位)	(76)
3.12	Windows API 类	(76)
3.12.1	标准应用程序支持	(76)
3.12.2	边框窗口	(76)
3.12.3	图形/GDI	(77)
3.12.4	对话框	(77)
3.12.5	控件	(77)
3.13	通用目的类	(77)
3.13.1	运行时间类型信息	(77)
3.13.2	对象永久性	(78)
3.13.3	集类	(78)
3.13.4	字符串	(78)
3.13.5	文件	(78)
3.13.6	时间和日期	(78)
3.14	调试和诊断支持	(78)
3.15	小结	(79)

第二部分 公共控件

第 4 章	新用户界面的核心:公共控件	(83)
4.1	引言	(83)
4.2	初识公共控件	(83)
4.2.1	动画控件	(83)
4.2.2	拖放列表框	(84)
4.2.3	头控件	(84)
4.2.4	热键控件	(84)
4.2.5	图像列表	(85)

4.2.6	列视控件	(85)
4.2.7	进度指示器	(85)
4.2.8	属性表	(87)
4.2.9	多信息编辑控件	(89)
4.2.10	状态条	(89)
4.2.11	标签控件	(90)
4.2.12	工具条	(90)
4.2.13	工具提示	(90)
4.2.14	游标	(91)
4.2.15	树视控件	(91)
4.2.16	滚动控件	(91)
4.3	SDK 支持	(93)
4.3.1	窗口类	(93)
4.3.2	公共控件风格	(93)
4.3.3	通知消息	(94)
4.4	MFC 支持	(95)
4.5	文档/视模型支持	(95)
4.6	对话框资源编辑器支持	(96)
4.7	小结	(96)
第 5 章 工具条、状态条和工具提示		(97)
5.1	引言	(97)
5.2	概念释疑	(97)
5.3	使用工具条编辑器	(98)
5.3.1	创建新工具条资源	(99)
5.3.2	把位图转换成工具条	(99)
5.3.3	创建、移动和编辑工具条按钮	(100)
5.4	编写 ToolStatusTip 例程	(100)
5.4.1	实现目标	(101)
5.4.2	实现细节	(101)
5.5	程序演示	(102)
5.6	代码分析	(104)
5.6.1	使用 CToolBar 和 CStatusBar	(104)
5.6.2	在 CMain Frame 中实现工具条、工具提示和状态条	(104)
5.6.3	组件廊产生当前日期和时间的状态条信息	(105)
5.7	信息唾手可得	(108)
5.8	小结	(111)
第 6 章 动画控件、进度指示器、滚动控件和游标控件		(112)
6.1	引言	(112)
6.2	动画控件和进度指示器	(112)
6.2.1	概念释疑	(112)
6.2.2	编程考虑	(112)
6.2.3	实现目标	(113)
6.2.4	创建基于对话框的起始应用程序 AnimateProgress	(113)

6.2.5	编辑对话框资源	(113)
6.2.6	添加代码	(114)
6.2.7	程序演示	(116)
6.2.8	信息唾手可得	(116)
6.3	滚动控件和游标控件	(118)
6.3.1	概念释疑	(118)
6.3.2	编程考虑	(119)
6.3.3	实现目标	(119)
6.3.4	创建基于对话框的起始应用程序 SliderSpin	(119)
6.3.5	编辑对话框资源	(119)
6.3.6	添加代码	(120)
6.3.7	程序演示	(124)
6.3.8	信息唾手可得	(124)
6.4	小结	(126)
第7章 快捷菜单、标签和属性表		(127)
7.1	引言	(127)
7.2	概念释疑	(127)
7.3	编程考虑	(128)
7.4	创建 SDI 起始应用程序 ContextMenuTabProperty	(129)
7.5	用组件廊增加 Pop-up Menu 组件	(130)
7.6	用组件廊增加 Property Sheet 组件	(131)
7.6.1	添加 Property Sheet 组件	(131)
7.6.2	编辑属性页资源	(134)
7.6.3	添加属性页类代码	(134)
7.6.4	添加属性表类代码	(140)
7.6.5	添加预览窗口类代码	(141)
7.6.6	添加视类代码	(141)
7.7	程序演示	(143)
7.8	小结	(144)
第8章 向导		(145)
8.1	引言	(145)
8.2	什么是向导	(145)
8.3	编程考虑	(145)
8.4	创建 SDI 起始应用程序 Wizard	(146)
8.5	用组件廊增加 Property Sheet 组件	(147)
8.5.1	添加 Property Sheet 组件	(147)
8.5.2	编辑属性页资源	(148)
8.6	在属性页类中添加代码	(150)
8.6.1	添加属性页控件变量	(150)
8.6.2	使“您是…”属性页的“上一步”按钮不能	(151)
8.6.3	使“适应性”属性页的“下一步”按钮不能	(151)
8.7	修改属性表类代码	(152)
8.8	在视类中添加代码	(152)

8.8.1	添加编辑控件.....	(152)
8.8.2	输出述职报告.....	(153)
8.9	程序演示.....	(158)
8.10	信息唾手可得	(159)
8.11	小结	(160)
第9章	多信息编辑控件:兼谈长文件名、MAPI 和注册表.....	(161)
9.1	引言.....	(161)
9.2	多信息编辑控件.....	(161)
9.2.1	概念释疑.....	(161)
9.2.2	Visual C++ 4.0 对多信息编辑控件的支持	(163)
9.3	长文件名.....	(163)
9.3.1	文件系统.....	(163)
9.3.2	Visual C++ 4.0 对长文件名的支持	(165)
9.3.3	应用程序支持长文件名.....	(165)
9.4	MAPI	(167)
9.4.1	MAPI 的概念和体系结构	(167)
9.4.2	MAPI 特征	(167)
9.4.3	MAPI 体系结构	(168)
9.4.4	MAPI 客户应用程序	(168)
9.4.5	MAPI 子系统	(169)
9.4.6	MAPI 缓冲池	(169)
9.4.7	服务提供者.....	(170)
9.4.8	Visual C++4.0 对 MAPI 的支持	(170)
9.5	注册表.....	(171)
9.5.1	注册表简介.....	(171)
9.5.2	注册数据库的结构.....	(171)
9.5.3	预定义键.....	(172)
9.5.4	更新注册表.....	(173)
9.5.5	Visual C++4.0 对注册表的支持	(174)
9.6	编程考虑.....	(174)
9.7	编写 RichEditMail 例程	(174)
9.7.1	用 AppWizard 创建起始程序	(174)
9.7.2	重用 Wizard 例程的代码	(175)
9.7.3	RichEditMail 对注册表的支持	(175)
9.7.4	在程序中增加字符和段落格式化功能.....	(176)
9.7.5	格式化述职报告.....	(176)
9.8	程序演示.....	(180)
9.9	信息唾手可得.....	(181)
9.10	小结	(187)
第10章	列视控件	(188)
10.1	引言	(188)
10.2	预备知识	(188)
10.2.1	列视控件	(188)

10.2.2	图像列表	(189)
10.2.3	Visual C++ 4.0 支持	(189)
10.3	编程考虑	(190)
10.4	编写 List 例程	(191)
10.4.1	用 AppWizard 创建起始程序	(191)
10.4.2	制作列表项图标资源	(191)
10.4.3	初始化列视列表项	(193)
10.4.4	获取列视控件项	(196)
10.4.5	用四种视窗显示列表项	(197)
10.4.6	响应列点击通知消息	(200)
10.5	程序演示	(201)
10.6	信息唾手可得	(204)
10.6.1	结构体参考	(204)
10.6.2	通知消息参考	(207)
10.6.3	CListCtrl 成员参考	(208)
10.6.4	CImageList 成员参考	(210)
10.7	小结	(211)
第 11 章	树视控件	(212)
11.1	引言	(212)
11.2	概念释疑	(212)
11.3	编程考虑	(212)
11.4	编写 Tree 例程	(213)
11.4.1	用 AppWizard 创建起始程序	(213)
11.4.2	制作列表项字符串和图标资源	(214)
11.4.3	初始化树视列表项	(215)
11.4.4	实现拖放编辑	(219)
11.4.5	实现父项在打开和关闭时显示不同的图标的	(222)
11.5	程序演示	(223)
11.6	信息唾手可得	(223)
11.6.1	结构体参考	(224)
11.6.2	通知消息参考	(226)
11.6.3	常数参考	(227)
11.6.4	CTreeCtrl 成员参考	(228)
11.7	小结	(230)

第三部分 OLE 2.0

第 12 章	OLE:一种集成技术	(232)
12.1	引言	(232)
12.2	OLE 产生的历史背景	(232)
12.3	OLE 是一种集成技术	(233)
12.4	OLE 基石:COM	(234)
12.5	结构化存储	(235)

12.6	别名(Moniker)	(238)
12.7	统一数据传输和 OLE 拖放	(239)
12.7.1	数据格式和传输媒体	(240)
12.7.2	剪贴板和拖放	(241)
12.7.3	OLE 拖放和数据传输类	(243)
12.8	OLE 文档	(243)
12.8.1	嵌入与链接	(244)
12.8.2	到嵌入的链接	(245)
12.8.3	在位激活/可视编辑	(245)
12.9	OLE 自动化	(246)
12.10	OLE 控件	(248)
12.11	小结	(249)
第 13 章	编写 COM 组件对象/客户程序	(250)
13.1	引言	(250)
13.2	准备知识	(250)
13.2.1	COM	(250)
13.2.2	二进制标准	(251)
13.2.3	组件对象	(252)
13.2.4	接口	(252)
13.2.5	全球唯一标识(GUID)	(255)
13.2.6	IUnknown	(255)
13.2.7	组件对象库	(256)
13.3	编程考虑	(257)
13.4	构造接口 IDrawing	(257)
13.4.1	用 C++ 实现 IDrawing	(257)
13.4.2	用组件廊组件 GUID Generator 产生 GUID	(258)
13.5	构造 COM 组件程序 SERVER.DLL	(259)
13.5.1	构造工程	(259)
13.5.2	增加对象和接口 GUIDs	(260)
13.5.3	创建组件对象类	(260)
13.5.4	创建组件对象 ID 文件	(260)
13.5.5	创建对象 REG 文件	(261)
13.5.6	增加类的接口定义	(261)
13.5.7	实现组件对象代码	(262)
13.6	构造组件客户程序 CLIENT.EXE	(264)
13.6.1	创建起始应用程序	(264)
13.6.2	构造 C++ 类 CDrawRect	(264)
13.6.3	实现 C++ 类 CDrawRect	(265)
13.6.4	绘制 C++ 对象和 COM 组件	(265)
13.7	小结	(268)
第 14 章	OLE 文档:分析服务器程序 HIERSVR	(269)
14.1	引言	(269)
14.2	HIERSVR 程序演示	(269)

14.2.1	在注册表中登记	(269)
14.2.2	链接对象	(269)
14.2.3	嵌入对象	(271)
14.2.4	剪贴板和拖放操作	(273)
14.3	在应用程序类中初始化 OLE 服务器	(273)
14.4	文档类提供的 OLE 服务器支持	(275)
14.4.1	登记剪贴板格式	(275)
14.4.2	创建或编辑嵌入对象	(275)
14.4.3	查找链接项	(276)
14.4.4	在位编辑	(276)
14.5	实现服务器对象	(277)
14.5.1	绘制 OLE 对象	(277)
14.5.2	获取 OLE 对象大小	(279)
14.5.3	谓词处理	(279)
14.5.4	获取剪贴板对象	(280)
14.5.5	显示文件数据	(281)
14.6	在视类中实现 OLE 在位编辑	(281)
14.6.1	设置绘图模式为 MM_ANISOTROPIC	(281)
14.6.2	实现 CServerView::OnDraw	(282)
14.7	在视类中实现剪贴板操作	(282)
14.7.1	拷贝	(282)
14.7.2	剪切	(283)
14.7.3	清除	(283)
14.7.4	粘贴	(284)
14.8	在视类中实现拖放操作	(285)
14.8.1	登记放目标对象	(285)
14.8.2	处理鼠标左键按下消息	(285)
14.8.3	实现 CServerView::OnDragEnter	(286)
14.8.4	实现 CServerView::OnDragOver	(287)
14.8.5	实现 CServerView::OnDragLeave	(288)
14.8.6	实现 CServerView::OnDrop	(288)
14.9	小结	(289)
第 15 章	OLE 文档:分析包容器程序 OCLIENT	(290)
15.1	引言	(290)
15.2	演示 OCLIENT	(290)
15.2.1	支持链接和嵌入对象的在位编辑	(290)
15.2.2	支持 OLE 公共对话框	(290)
15.2.3	支持剪贴板和拖放操作	(290)
15.3	OLE 公共对话框	(291)
15.3.1	“插入对象”对话框	(292)
15.3.2	“选择性粘贴”对话框	(292)
15.3.3	“更改图标”对话框	(292)
15.3.4	“转换”对话框	(293)

15.3.5	“编辑链接”或“更新链接”对话框	(294)
15.3.6	“服务器占线”或“服务器没有响应”对话框	(295)
15.3.7	“属性”对话框	(295)
15.4	在应用程序类中初始化 OLE 容器	(295)
15.5	文档类提供的 OLE 容器支持	(297)
15.5.1	登记剪贴板格式	(297)
15.5.2	重载 CMainDoc::OnShowViews	(298)
15.6	实现客户对象	(298)
15.6.1	客户对象状态	(298)
15.6.2	实现 CRectItem::OnGetItemPosition	(300)
15.6.3	实现 CRectItem::OnChangeItemPosition	(300)
15.7	在视类中实现矩形跟踪器	(301)
15.7.1	矩形跟踪器(Rectangle Tracker)	(301)
15.7.2	实现矩形跟踪器	(301)
15.8	实现视类	(304)
15.8.1	绘制 OLE 对象	(304)
15.8.2	实现 CMainView::OnInsertObject	(306)
15.8.3	实现 CMainView::OnSetFocus	(307)
15.8.4	实现 CMainView::OnSize	(307)
15.9	在位编辑	(308)
15.10	剪贴板和拖放操作	(308)
15.10.1	有关粘贴的核心实现 CMainView::DoPasteItem	(309)
15.10.2	粘贴链接	(313)
15.10.3	选择性粘贴	(313)
15.10.4	拖放操作中自动滚动滚动条	(315)
15.11	显示 OIE 对象的属性	(316)
15.11.1	获取快捷菜单	(317)
15.11.2	显示属性对话框	(317)
15.12	显示 OLE 客户对象	(317)
15.13	小结	(319)
第 16 章	OLE 自动化:用 VC 创建自动化对象	(320)
16.1	引言	(320)
16.2	自动化服务器/客户	(320)
16.3	Visual C++ 4.0 对 OLE 自动化服务器支持	(321)
16.3.1	MFC 框架支持	(321)
16.3.2	ClassWizard 支持	(321)
16.3.3	AppWizard 支持	(322)
16.4	在应用程序中使能 OLE 自动化	(322)
16.4.1	创建 OLE 自动化服务器框架程序	(322)
16.4.2	分析派遣接口名	(323)
16.5	分析 AppWizard 提供的代码	(324)
16.5.1	自动化服务器的应用程序类	(324)
16.5.2	自动化服务器的文档类	(325)

16.5.3	创建 OLE 类型库	(326)
16.6	实现 DrawCircle 的基本行为	(327)
16.7	实现自动化属性和方法	(328)
16.7.1	实现派遣接口的属性	(328)
16.7.2	实现派遣接口的方法	(330)
16.7.3	ClassWizard 产生的对象定义库	(332)
16.8	程序演示	(333)
16.9	小结	(334)
第 17 章	OLE 自动化:用 VC 和 VB 访问自动化对象	(335)
17.1	引言	(335)
17.2	OLE 自动化客户	(335)
17.3	使用类型库	(336)
17.3.1	ODL 文件语法	(336)
17.3.2	MkTypLib:创建类型库工具	(337)
17.3.3	ClassWizard 对类型库的支持	(338)
17.4	用 VC 编写自动化客户程序 SetCircle	(338)
17.4.1	创建框架程序	(338)
17.4.2	从 SliderSpin 例程中获取可重用资源和代码	(339)
17.4.3	创建 TLB 文件重建派遣类	(342)
17.4.4	引用和访问派遣类 CDrawCircle	(345)
17.4.5	程序演示	(346)
17.5	用 VB 编写自动化客户程序 SetCircle	(347)
17.5.1	构造 SetCircle	(347)
17.5.2	实现 SetCircle	(348)
17.5.3	程序演示	(350)
17.6	小结	(350)
第 18 章	OLE 控件:用 VC 创建控件	(351)
18.1	引言	(351)
18.2	OLE 控件综述	(351)
18.2.1	OLE 控件的基本组成部分	(352)
18.2.2	控件与控件容器之间的交互	(352)
18.2.3	OLE 控件的激活和非激活状态	(353)
18.2.4	序列化	(353)
18.3	名词释疑	(353)
18.3.1	事件	(353)
18.3.2	方法	(355)
18.3.3	属性	(356)
18.3.4	序列化	(357)
18.3.5	属性页	(358)
18.3.6	许可	(359)
18.4	OLE ControlWizard 支持	(359)
18.4.1	工程选项	(360)
18.4.2	控件选项	(361)