

Visual C++ 高级编程技术 —— 开发实例剖析

陈建春 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

Visual C++ 高级编程技术

—— 开发实例剖析

陈建春 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

本书是作者继《Microsoft Visual C++ 图形系统开发技术基础》后推出的又一本力作。本书结合具体的开发实例全面介绍和分析了各种 MFC 辅助开发工具的使用方法,包括与图形有关的 MFC 类、面向对象的程序设计基础和 C++ 类的组织、文档和视图的实现、鼠标交互绘图、视图的无级放缩和回溯、图形的选中、图形的删除和文档逆向操作、文档的存储组织、对话框的使用、剪裁板的使用、多文档多视图程序设计、控制条的使用、位图的使用、图形的打印和打印预显、DLL 的开发和使用、OLE 的开发和使用、ActiveX 控件的开发和使用、帮助系统组织等各种开发技术。在讲述中,以一个基本矢量图形系统的开发为主线贯穿全书,并附有完整的实现代码。不论是初学者还是高级技术人员,都会从本书中获益。本书的代码适用于 VC++ 5.0 及其以上版本。

本书适用于从事计算机开发的广大技术人员及大专院校的师生阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

Visual C++ 高级编程技术:开发实例剖析/陈建春编著. - 北京:电子工业出版社,1999.9

ISBN 7-5053-5377-2

I. V… II. 陈… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 14548 号

书 名: Visual C++ 高级编程技术——开发实例剖析

编 著 者: 陈建春

责任编辑: 杜振民 祖振升

排版制作: 电子工业出版社计算机排版室

印 刷 者: 民族印刷厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 46 字数: 1148 千字

版 次: 1999 年 9 月第 1 版 1999 年 9 月第 1 次印刷

书 号: ISBN 7-5053-5377-2
TP·2704

印 数: 4000 册 定价: 80.00 元 (含光盘)

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;

若书店售缺,请与本社发行部联系调换。电话 68279077

原书缺页

- **缺乏有效地指导**

在大学就读的学习者,只是学了基本知识,打了一些继续学习的基础,离独立开发应用程序还有相当大的距离。而在工作岗位上的学习者则全靠自己摸索和自学。许多读者反映,目前虽然关于 VC++ 和 MFC 程序设计方面的书籍很多,但大部分都只介绍 VC++ 本身的各种编程工具的用法,有的只对使用 VC++ 的一些工具(如 AppStudio,Classwiard 等)和 MFC 的标准功能做一些使用说明,而较少有全面而具体的分析 MFC 程序框架结构方面的书籍,更少有结合具体的开发实例对 MFC 编程技术进行系统介绍的。

编写本书的思路

编写本书的思路,在于以开发一个完整的矢量图形系统为主线,一步步引导读者,去掌握面向对象程序设计中的精髓,全面分析 MFC 程序框架的组成和结构,实例介绍使用 MFC 程序设计的各种技术。使读者从数据组织、MFC 程序设计技术、VC++ 开发工具应用、MFC 程序框架本质等各个方面,综合掌握 MFC 程序设计的技术精华。期望对读者能有所启发,从程序设计思想和设计技术等各个方面全面提高编程能力,提高组织和开发 Windows 应用程序的分析能力和实践能力。

本书的特点

作者多年从事 GIS 和 MIS 系统的开发工作,本书的大部分内容是实际工作的提炼和总结,所以本书着意突出以下特点。

- **较强的连贯性和完整的代码**

本书从一个利用 AppWizard 生成的程序框架出发,并依此为基础,就如何分析和组织一个系统的数据结构到绘制图形元素、用鼠标交互绘制图形、对话框的使用、图形放缩移动、滚动视图、鼠标选中图形、图形删除和恢复、图形的打印、图形数据的存储、文档的存储组织、剪裁板剪裁和拷贝图形元素、多文档多视图的使用、多线程使用、DLL 开发和使用、OLE 使用、ActiveX 控件开发和使用、上下文帮助功能的组织等各个方面,结合实际的代码进行了讲解。通过对实例的剖析,让读者在一个较完整的矢量图形系统的开发过程中,掌握系统的设计思路和各种开发技术、开发方法。作者以实际的代码向读者展示程序的具体实现,力图避免枯燥无味的讲解。

- **全面翔实的代码注释**

本书中的内容都是从实现的思路、实现的方法和具体实现等几个方面进行讲解。作者给出了以标准格式写出并经过全面调试的代码,代码中的操作都给出了完整的中文注释,使读者不但知道实现的方法,而且能够轻松的知道怎么实现。

- **适合不同读者的需要**

初学者能够通过本书循序渐进地学会从编程语言的应用到系统实际实现的方法,熟练者也可以从本书中得到从程序设计思路到具体功能实现等很多方面的东西。

本书概要

本书第 1 章全面分析和介绍了利用 Visual C++ 程序设计必须或辅助使用的几个开发工具,介绍和分析了利用 MFC AppWizard 生成一个 Windows 的框架程序,利用 ClassWizard、

ClassView、AppStudio 等开发工具组织和开发程序的方法;第 2 章从文件、应用程序类等各个方面分析了 MFC 应用程序框架的组成和结构;第 3 章介绍了各种与图形设计有关的 MFC 类和与绘图有关的一些基本操作;第 4 章介绍了面向对象程序设计和 C++ 类的基本知识,并具体组织了一个矢量图形系统中各种图形元素类;第 5 章介绍了文档和视图,实现了一个基本图形系统的文档和视图;第 6 章介绍用鼠标器进行交互图形绘制的方法并实现了各种图形元素的交互绘制功能;第 7 章介绍了对话框的使用技术,并利用对话框实现了矢量文本的标注和字体的设置等功能;第 8 章介绍了增强视图功能的方法,实现了图形系统的无级放缩、移动、图形屏幕回溯等有关的各种操作,并实现了滚动视图;第 9 章介绍了用鼠标器在屏幕上选中图形元素的操作实现,介绍了文档的删除和恢复,实现了文档的逆向操作,介绍了系统的操作逆向化;第 10 章介绍了文档的存储和读取操作,结合基本图形系统讨论了串行化和文件流操作等两种文档存取的方法;第 11 章介绍了剪裁板的使用方法,并用剪裁板为图形系统实现了位图的剪裁和读取、图形元素的剪裁和粘贴操作;第 12 章对多文档、多视图的实现本质进行了分析,并结合图形系统的开发讨论了多文档和多视图程序设计注意的主要问题;第 13 章介绍了多线程程序设计技术的实现,并为图形系统实现了后台绘画功能;第 14 章对图形的打印和打印预显功能进行了分析,并实现了图形系统的任意坐标、任意比例打印输出功能;第 15 章对位图的使用技术进行了全面的介绍,实例介绍了几种显示位图的方法;第 16 章介绍了工具条、状态条、对话条以及加速键等的编程技术,为图形系统增加或改造了工具条、状态条、对话条、加速键等,第 17 章介绍和分析了开发和使用 DLL 的技术,为图形系统开发和使用了多种 DLL;第 18 章介绍了 OLE 编程技术,详细分析了 OLE 包容器程序的设计技术并为图形系统实现了 OLE 包容器功能,介绍了 OLE 服务器应用程序的设计技术,介绍了 OLE 的自动化设计技术;第 19 章介绍了 ActiveX 控件的开发和使用方法,详细分析和实例进行了 ActiveX 控件的开发设计和测试方法;第 20 章详细分析了 MFC 程序的上下文帮助实现技术,为图形系统实现了上下文帮助功能;第 21 章给出了本书实现的基本矢量图形系统的代码,供学习参考使用。

对读者的要求

阅读本书要求读者具有 C 和 C++ 编程的一些基本知识。受篇幅所限,本书对具体的 VC++ 系统的函数不做过多的说明,读者可以结合一些 VC++ 的参考手册阅读。本书介绍的编程技术和开发实例是在 VC++ 5.0 版本上实现的,对 VC++ 5.0 以上版本同样适用。

曾学贵教授对于本书的编排进行了悉心指导并提供了宝贵意见,梁青槐、张海燕、郭文军、郑顺义、万传风、郑屹、刘建峰等同志参加了本书的编写工作,杜薇、李洁等同志参加本书的录入工作,在此一并感谢。由于作者水平有限,书中难免有错误和疏漏之处,敬请各位专家、老师和读者给予批评指正。

陈建春

1999 年 5 月于北方交大

关于作者

陈建春, 30 岁, 北方交通大学教师、博士。长期从事工程 CAD 系统、GIS 系统、MIS 系统的研究和开发工作。组织《新建铁路施工指挥系统》、《铁路 GIS 系统》、《通用多媒体 GIS 系统》等软件系统的研究和开发工作。出版过《Microsoft Visual C++ 图形系统开发技术基础》等著作。

E_mail: chenjianchun@263.net

目 录

第 1 章 Visual C++ 编程时必须或辅助使用的几个开发工具	(1)
1.1 使用 MFC AppWizard	(2)
1.1.1 MFC AppWizard 的启动	(2)
1.1.2 利用 AppWizard 生成一个基于 MFC 的应用程序	(2)
1.2 使用 ClassView	(7)
1.2.1 启动 ClassView	(7)
1.2.2 编辑定位	(8)
1.2.3 增加类	(8)
1.2.4 增加类的成员	(10)
1.3 使用 ResourceView	(12)
1.3.1 ResourceView 的调用	(12)
1.3.2 ResourceView 工作分析	(12)
1.4 使用 FileView	(16)
1.4.1 打开一个工作项目	(16)
1.4.2 FileView 相关的操作功能	(17)
1.5 使用 ClassWizard	(18)
1.5.1 Windows 消息	(18)
1.5.2 消息处理函数	(20)
1.5.3 ClassWizard 的作用和启动	(22)
1.5.4 ClassWizard 管理消息处理函数	(22)
1.5.5 ClassWizard 管理可重载的虚函数	(26)
1.5.6 ClassWizard 管理映射变量	(28)
1.5.7 管理 Automation 和 ActiveX 事件	(32)
1.5.8 重新构造一个开发项目的 ClassWizard 工作信息	(32)
1.6 使用各设计工具的几点建议	(33)
第 2 章 基于 MFC 的一般应用程序框架分析	(35)
2.1 程序类和文件	(35)
2.2 程序框架中各个类的作用分析	(37)
2.3 程序的工作步骤分析	(38)
2.4 系统产生的标准菜单执行分析	(41)
2.4.1 标准菜单的执行分析	(41)
2.4.2 修改标准菜单完成的功能	(43)
2.5 程序框架中各个类对象的相互调用关系	(43)
2.6 包含详尽注释的主要源程序清单	(45)

第 3 章 与图形操作有关的类及主要操作	(71)
3.1 绘图类	(71)
3.1.1 绘图类简介	(71)
3.1.2 几个绘图类的主要差别	(72)
3.1.3 绘图类的引用方法	(73)
3.2 绘图对象类	(74)
3.2.1 CGdiObject 类	(74)
3.2.2 CPen 类	(74)
3.2.3 CBrush 类	(76)
3.2.4 CFont 类	(79)
3.2.5 CRgn 类	(82)
3.2.6 CPalette 类	(87)
3.2.7 CBitmap 类	(89)
3.3 绘图类的主要操作功能	(90)
3.3.1 坐标映像方式	(90)
3.3.2 选用绘图类对象	(96)
3.3.3 主要绘图属性设置函数	(97)
第 4 章 面向对象的程序设计和 C++ 类组织	(99)
4.1 面向对象程序设计和 C++ 类的基础知识	(99)
4.1.1 面向对象程序设计的初步知识	(99)
4.1.2 类和对象	(101)
4.1.3 函数重载	(105)
4.1.4 类的构造函数和析构函数	(105)
4.1.5 类的继承	(106)
4.1.6 虚函数和多态性	(110)
4.2 分析建立一个基本矢量图形系统 Draw 的基本类	(114)
4.2.1 建立一个图形基本类	(114)
4.2.2 通过继承建立直线类	(115)
4.2.3 通过继承建立连续直线或封闭多边形类	(116)
4.2.4 通过继承建立圆类	(117)
4.2.5 通过多级继承建立圆弧类	(118)
4.2.6 通过继承建立标注文本类	(118)
4.2.7 建立图形参数类	(119)
4.3 基本矢量图形系统类的组织和实现列表	(121)
第 5 章 实现文档和视图	(127)
5.1 组织矢量图形系统 Draw 的文档	(127)
5.1.1 组织面向对象的文档存储管理机制	(127)
5.1.2 利用 MFC 模板定义管理图形元素对象指针的对象	(127)
5.1.3 实现 Draw 系统的文档	(128)

5.1.4	实现 Draw 系统文档的管理功能	(129)
5.2	实现矢量图形系统 Draw 的视图	(133)
5.2.1	为 Draw 系统建立一个适合各种映像方式的坐标系	(133)
5.2.2	实现各图形元素类的绘制函数	(138)
5.2.3	视图类的 OnDraw 函数	(147)
5.2.4	实现视图	(148)
第 6 章	使用鼠标器交互作图	(150)
6.1	用鼠标器作图要解决的主要问题	(150)
6.1.1	建立鼠标消息处理函数捕捉鼠标器消息	(150)
6.1.2	捕捉所有的鼠标输入	(151)
6.1.3	实现在屏幕上拖动图形	(152)
6.1.4	保存图形数据到文档	(152)
6.1.5	将图形以实际的形态重画	(152)
6.2	加入一个绘图菜单	(153)
6.2.1	增加绘图菜单资源	(153)
6.2.2	增加菜单的消息处理函数	(154)
6.3	实现各图形元素的交互绘制功能	(156)
6.3.1	绘制直线	(156)
6.3.2	绘制连续直线和多边形区域	(159)
6.3.3	绘制圆和圆形区域	(163)
6.3.4	绘制圆弧	(166)
第 7 章	使用对话框	(174)
7.1	对话框程序设计概况	(174)
7.1.1	对话框程序的组成	(174)
7.1.2	对话框的类型	(174)
7.2	对话框编程的主要技术	(175)
7.2.1	创建对话框资源	(175)
7.2.2	创建对话框类	(176)
7.2.3	对话框类中的常用函数	(178)
7.2.4	对话框类对象的定义和使用	(180)
7.2.5	控制无模式对话框	(182)
7.2.6	对话框中使用各种控制	(183)
7.2.7	对话框中的消息操作	(184)
7.3	用无模式对话框完成 Draw 系统的交互文本标注	(186)
7.3.1	交互标注文本的操作过程	(186)
7.3.2	定义对话框资源	(186)
7.3.3	实现对话框类	(187)
7.3.4	修改视图类实现文本交互标注	(195)
7.4	用模式对话框实现 Draw 系统中字体参数的设置	(199)

7.4.1	建立对话框资源	(199)
7.4.2	实现对话框类	(200)
第 8 章	增强视图的操作功能	(203)
8.1	增加一个图形操作菜单	(203)
8.1.1	增加菜单资源	(203)
8.1.2	增加消息处理函数	(204)
8.2	图形重画	(205)
8.2.1	全屏重画	(205)
8.2.2	局部重画	(206)
8.3	图形放大和移动	(207)
8.3.1	图形放大或移动后图形参数的计算方法	(207)
8.3.2	图形放大和移动的实现方法	(208)
8.4	重画上屏和重画首屏	(211)
8.5	显示全图	(215)
8.5.1	操作目的和实现思路	(215)
8.5.2	各种图形元素的矩形边界	(215)
8.5.3	实现重画全图功能	(220)
8.6	提高图形重画速度的基本方法	(223)
8.6.1	基本原理及方法	(223)
8.6.2	提高图形重画速度的实现方法	(224)
8.6.3	修改图形元素的绘制函数	(225)
8.7	实现滚动视图	(226)
8.7.1	实现滚动视图的方法	(226)
8.7.2	必要的参数设置	(227)
8.8	为 Draw 系统定制自身的图形滚动功能	(228)
8.8.1	确定滚动步长	(228)
8.8.2	产生一个滚动条	(229)
8.8.3	实现滚动功能	(229)
8.8.4	加快图形滚动的速度	(234)
8.8.5	用键盘滚动图形	(236)
第 9 章	文档的删除和恢复	(238)
9.1	用鼠标在屏幕上选中图形元素	(238)
9.1.1	各种图形元素的选中区域	(238)
9.1.2	各种图形元素的选中判断	(243)
9.1.3	选中图形的特殊显示	(247)
9.1.4	实现图形选中功能	(247)
9.1.5	放弃选中的图形元素	(253)
9.1.6	图形元素显示和选中的顺序	(254)
9.2	图形元素的删除	(254)

9.2.1	实现可恢复的图形元素删除方法	(254)
9.2.2	即时从屏幕上删除图形	(255)
9.2.3	各种图形元素的删除实现	(256)
9.2.4	实现图形元素删除功能	(259)
9.3	实现 Draw 系统文档操作逆向化	(260)
9.3.1	逆向操作的含义	(260)
9.3.2	记录文档操作过程	(261)
9.3.3	实现文档逆向操作	(268)
9.3.4	实现操作过程的完全逆向化	(270)
第 10 章	文档的存储和读取	(271)
10.1	利用 CFile 对象存储和读取文件	(271)
10.1.1	构造和删除一个 CFile 对象	(271)
10.1.2	CFile 类的主要操作	(273)
10.1.3	CFile 的派生类	(274)
10.2	利用串行化存取文档	(274)
10.2.1	CArchive 类及 CArchive 对象的构造	(274)
10.2.2	Serialize 函数	(276)
10.2.3	CObArray 类的 serialize 函数	(278)
10.3	用串行化实现 Draw 系统的文档存储和读取功能	(279)
10.3.1	实现各个图形元素类的串行化	(279)
10.3.2	MFC 框架程序的文档串行化机制	(284)
10.3.3	实现文档的存储和读取	(284)
10.3.4	SetModifiedFlag 函数	(287)
10.4	用 CFile 对象实现 Draw 系统文档的存储和读取	(289)
10.4.1	在各个图形类中增加一个存取数据成员的函数	(289)
10.4.2	实现文档的存储和读取	(293)
第 11 章	使用剪裁板	(299)
11.1	剪裁板使用方法	(299)
11.1.1	打开、关闭、清空剪裁板	(299)
11.1.2	存取剪裁板	(300)
11.1.3	注册和检查剪裁板格式	(302)
11.2	用剪裁板实现 Draw 系统剪裁和粘贴图像的功能	(304)
11.2.1	增加操作菜单	(305)
11.2.2	剪裁视图到剪裁板	(305)
11.2.3	插入剪裁板的位图到当前视图	(306)
11.3	用剪裁板实现 Draw 系统中矢量图形元素的剪裁、移动和复制	(308)
11.3.1	实现各图形元素数据的字符串表示	(308)
11.3.2	将要剪裁的图形元素数据写入剪裁板	(313)
11.3.3	从剪裁板中读出数据成员形成图形元素对象	(317)

11.3.4	实现 Draw 系统的 Cut、Copy 和 Paste 操作功能	(321)
第 12 章	多视图、多文档编程	(326)
12.1	单文档和多文档应用程序比较	(326)
12.1.1	用 MFC AppWizard 产生程序框架比较	(326)
12.1.2	程序框架类比较	(326)
12.1.3	标准菜单比较	(327)
12.2	单文档应用程序分析	(328)
12.2.1	建立文档模板对象	(328)
12.2.2	打开文档操作	(329)
12.2.3	关闭文档操作	(332)
12.2.4	保存文档操作	(333)
12.3	多文档应用程序分析	(335)
12.3.1	建立文档模板对象	(335)
12.3.2	打开文档	(336)
12.3.3	关闭和保存文档	(338)
12.3.4	创建视图	(338)
12.4	多文档、多视图编程注意事项	(340)
12.4.1	全局变量的使用	(340)
12.4.2	共享资源的使用	(341)
12.4.3	多文档、多视图应用程序文档和视图类数据成员的组织	(342)
第 13 章	多线程编程	(344)
13.1	管理线程的技术	(344)
13.1.1	设置开发环境参数	(344)
13.1.2	启动及终止线程	(344)
13.1.3	管理线程	(347)
13.2	线程中使用 MFC 类对象	(348)
13.2.1	线程中共享数据及对象	(348)
13.2.2	线程中使用主线程的视图类和文档类对象	(348)
13.2.3	使用 CDC 类对象	(350)
13.2.4	使用 CGdiObject 类对象	(351)
13.3	线程的同步化方法	(351)
13.3.1	线程同步化概念	(351)
13.3.2	使用互斥信号灯同步化线程	(352)
13.3.3	其他同步化方法	(353)
13.3.4	其他同步化类型	(354)
13.4	实现 Draw 系统的图形后台绘制	(355)
13.4.1	实现一个图形绘制线程	(355)
13.4.2	实现同步化	(357)
第 14 章	图形打印和打印预览	(360)

14.1	视图打印实现过程分析	(360)
14.1.1	视图打印的机理	(360)
14.1.2	视图打印过程分析	(361)
14.2	实现 Draw 系统的多坐标系无级打印和打印预显功能	(369)
14.2.1	进行分页	(369)
14.2.2	打印标题、注脚等其他信息	(372)
14.2.3	任意比例分页打印的具体实现	(372)
14.2.4	打印预显	(377)
第 15 章	显示位图	(378)
15.1	位图文件的格式	(378)
15.2	CBitmap 类及 CBitmap 类对象的初始化方法	(381)
15.2.1	CBitmap 类简介	(381)
15.2.2	初始化 CBitmap 类对象的方法	(382)
15.3	显示位图的步骤和主要的位操作函数	(384)
15.3.1	显示位图的步骤	(384)
15.3.2	位操作函数	(385)
15.4	显示位图的函数实现	(387)
15.4.1	显示装载在资源文件中的位图	(387)
15.4.2	通过 HBITMAP 句柄装载显示位图文件	(389)
15.4.3	用画点法选择装入位图数据显示位图	(394)
第 16 章	使用工具条、状态条、对话条及加速键	(403)
16.1	使用工具条	(403)
16.1.1	工具条的实现过程	(403)
16.1.2	为 Draw 系统创建一个绘图工具条	(407)
16.2	使用状态条	(410)
16.2.1	状态条的实现过程	(410)
16.2.2	为 Draw 系统在状态条中增加坐标跟踪提示	(413)
16.3	使用对话条	(414)
16.3.1	对话条的实现过程	(415)
16.3.2	在 Draw 系统中实现设置绘图参数的对话条	(418)
16.4	加速键等的实现方法	(421)
16.4.1	使用 CCmdUI 消息处理函数	(422)
16.4.2	修改命令交互对象的命令描述	(425)
16.4.3	设置加速键	(427)
第 17 章	使用和开发 DLL	(430)
17.1	DLL 的使用	(430)
17.1.1	DLL 的特点	(430)
17.1.2	Windows 中的 DLL	(430)
17.1.3	MFC 库中的 DLL	(431)

17.1.4	DLL 的文件组成	(431)
17.1.5	使用 DLL 的方法	(432)
17.2	VC++ 5.0 能够开发的基于 MFC 的 DLL 种类	(434)
17.2.1	通常形式的静态使用 MFC 的 DLL	(435)
17.2.2	通常形式的动态使用 MFC 的 DLL	(436)
17.2.3	扩展 DLL	(437)
17.3	调整开发平台设置来开发和调试 DLL	(441)
17.3.1	管理多个程序项目	(441)
17.3.2	调整 DLL 开发项目和使用 DLL 项目的路径设置	(443)
17.4	在 Draw 系统中开发和使用 DLL	(445)
17.4.1	利用通常形式的静态 DLL 实现标注字体设置对话框	(446)
17.4.2	利用通常形式的动态 DLL 实现标注字体设置对话框	(450)
17.4.3	利用扩展 DLL 实现标注字体设置对话框	(451)
第 18 章	OLE 编程	(455)
18.1	OLE 使用技术	(455)
18.1.1	OLE 主要术语	(455)
18.1.2	OLE 服务器应用程序调用的方式和实现	(458)
18.1.3	MFC 中的主要 OLE 类	(459)
18.1.4	MFC AppWizard 创建的 OLE 程序框架	(462)
18.2	OLE 容器应用程序设计	(463)
18.2.1	程序框架组成	(463)
18.2.2	OLE 条目的文档组织	(465)
18.2.3	增加和删除 OLE 条目	(469)
18.2.4	OLE 条目绘制过程	(471)
18.2.5	OLE 条目的选中和拖动	(473)
18.2.6	OLE 条目的可视原位编辑	(475)
18.2.7	OLE 条目的存取	(476)
18.3	在 Draw 系统中实现 OLE 容器功能	(479)
18.3.1	给 OLE 条目类增加数据成员	(479)
18.3.2	绘制 OLE 条目	(480)
18.3.3	增加 OLE 条目	(482)
18.3.4	OLE 条目的选中和拖动	(486)
18.3.5	对 OLE 条目进行清除选中和删除操作	(488)
18.3.6	编辑 OLE 条目	(490)
18.3.7	保存 OLE 条目	(491)
18.4	OLE 服务器程序开发技术分析	(491)
18.4.1	程序类分析	(492)
18.4.2	交互操作界面分析	(492)
18.4.3	视图分析	(496)

18.4.4	文档分析	(497)
18.5	开发 OLE 混合应用程序	(500)
18.5.1	程序类分析	(500)
18.5.2	交互操作界面分析	(501)
18.5.3	文档组织分析	(503)
18.6	Automation 支持	(503)
18.6.1	术语	(503)
18.6.2	Automation 方法的定义和使用	(504)
18.6.3	Automation 属性的定义和使用	(509)
18.6.4	OLE 自动化容器获得 OLE 自动化服务器接口的方法	(515)
第 19 章	ActiveX 控件的开发及使用	(516)
19.1	ActiveX 控件特点、构成及开发技术分析	(516)
19.1.1	ActiveX 控件的特点	(516)
19.1.2	用 MFC AppWizard 产生 ActiveX 控件开发程序框架	(518)
19.1.3	ActiveX 控件的主要开发工作	(520)
19.2	开发 ActiveX 控件的 Automation 属性和方法	(521)
19.2.1	开发 Automation 属性和方法的程序框架	(521)
19.2.2	增加一个库存属性	(522)
19.2.3	增加其他 Automation 属性和方法	(524)
19.3	开发 ActiveX 控件的事件	(525)
19.3.1	开发事件的程序框架	(525)
19.3.2	增加库存事件	(526)
19.3.3	增加和激发定制事件	(528)
19.4	开发 ActiveX 控件的属性页	(530)
19.4.1	属性页开发的程序框架	(531)
19.4.2	属性页的开发	(532)
19.5	开发一个直线 ActiveX 控件实例	(534)
19.5.1	给控件增加 Automation 属性	(535)
19.5.2	给控件增加 Automation 方法	(537)
19.5.3	给控件增加事件	(538)
19.5.4	完成控件的绘制	(539)
19.5.5	激发事件	(540)
19.5.6	完成控件的属性页开发	(542)
19.5.7	ActiveX 控件的编译及注册	(546)
19.6	ActiveX 控件的测试和使用	(546)
19.6.1	使用 VC++ 提供的测试容器程序	(546)
19.6.2	在对话框中使用 ActiveX 控件	(549)
第 20 章	实现系统的帮助功能	(553)
20.1	开发帮助功能的程序框架	(553)

20.1.1	MFC 应用程序中帮助功能的程序代码组成	(553)
20.1.2	MFC 应用程序的标准帮助功能及其实现	(557)
20.2	求助主题组织	(561)
20.2.1	组织求助主题和普通显示内容	(562)
20.2.2	组织求助主题的标识	(563)
20.2.3	组织求助主题中的跳转	(566)
20.2.4	使用 Word 编辑 RTF 文件可能出现的问题	(568)
20.3	组织帮助项目文件 prj	(568)
20.3.1	求助项目文件中常用的节	(569)
20.3.2	其他一些在 HPJ 文件中使用的节	(572)
20.4	实现 Draw 系统的帮助功能	(573)
20.4.1	创建每个菜单项的求助主题页	(573)
20.4.2	创建菜单的求助目录主题页并与帮助文件的主目录主题连接	(574)
20.4.3	测试新增帮助功能	(576)
第 21 章	矢量图形系统 Draw 的源代码列表	(579)
21.1	Draw 系统的主要功能	(579)
21.2	Draw 系统的源代码文件列表	(581)
21.2.1	Draw 系统开发过程中增加或做过修改的文件	(581)
21.2.2	drawdoc.h 和 drawdoc.cpp 文件列表	(581)
21.2.3	视图类的 drawview.h 和 drawview.cpp 文件	(643)
21.2.4	主框架类的 mainfrm.h 和 mainfrm.cpp 文件	(697)
21.2.5	OLE 条目控制类的 ctrlitem.h 和 ctrlitem.cpp 文件	(702)
21.2.6	对话框类的 textwritedlg.h 和 textwritedlg.cpp 文件	(708)
附录 A	本书所附光盘的内容及技术支持	(715)
附录 B	参考书目	(716)