

VISUAL C++ 4

编程大全

[美] Chane Cullens 等著

石耀宇 潘朋 李蓓 译

熊桂喜 审校



浙江科学技术出版社
西蒙与舒斯特国际出版公司

QUE®

VISUAL C++4 编程大全

[美] Chane Cullens 等著

石耀宇 潘朋 李蓓 译

熊桂喜 审校

浙江科学技术出版社
西蒙与舒斯特国际出版公司

Chane Cullens, Mark Davidson, Paul Robichaux, Chris Corry, Steve Potts and Kate Gregory: USING Visual C++ 4

Authorized translation from the English language edition published by Que® Corporation

Copyright © 1997 by Que® Corporation

All rights reserved. For Sale in Mainland China Only

本书中文简体版由浙江科学技术出版社和美国西蒙与舒斯特国际出版公司合作出版, 未经出版者书面许可, 本书任何部分均不得以任何方式复制或抄袭。本书封面贴有西蒙与舒斯特防伪标签, 无标签者不得销售。

版权所有, 翻印必究。

浙江省版权局著作权合同登记号图字: 11-1997-26

VISUAL C++ 4 编程大全

[美]Chane Cullens 等著

石耀宇 潘朋 李蓓 译

熊桂喜 审校

*

浙江科学技术出版社 出版
西蒙与舒斯特国际出版公司

杭州富春印务有限公司印刷

浙江省新华书店发行

*

开本 787×1092 1/16 印张 34 字数 814 000

1998年9月第一版

1998年9月第一次印刷

ISBN 7-5341-1142-0/TP·46

定价: 56.50元

责任编辑: 陈亚猜

封面设计: 金 晖

译者序

为了更好地为广大 Windows 程序员提供帮助,Microsoft 公司推出了更加完善的 Visual C++ 的开发环境,重新设计了程序设计环境,创建了 Developer Studio,更广泛地使用了 AppWizard(应用向导),增强了 ClassWizard(类向导)的集成度,并添加了能与预先建立的解决方案更容易集成的 Component Gallery(构件库)。值得说明的是,4.0 版的 AppWizard 提供了直接基于对话框生成应用程序的选项,使创建基于对话框的应用程序异常容易,且在编辑过程中还会产生一些特殊的效果。这些实用程序使得创建高级 Windows 程序更加简单。Microsoft 公司和其他一些公司一起扩展了 Developer Studio 的功能,使它比任何时候都更加容易地完成创建和调试改进的 GUI 应用程序的任务。例如,如果出现了调试问题,就会进入 Nu Mega 的 BoundsChecker 中;如果要寻找另外的预建立构件,就可进入 Visual Component 电子表格构件或者图表构件中。Visual C++ 作为一种面向对象的可视化设计工具,越来越广泛地应用于开发各种应用系统,它提供了非常新颖的可视化开发工具,使程序员得以抛弃传统的开发方式。

目前,Visual C++ 以其功能强大而广为运用,它不但可开发 Windows 3.x 和 DOS 的产品,而且可开发 Windows95 和 WindowsNT 的高档产品,并可向 UNIX 系统移植。其中开发多媒体产品也是 Visual C++ 的一个强项。

早期的 C 和 SDK 在开发 GUI 界面的面向对象的多媒体应用软件时,工作量大,开发周期长,修改也不方便,而且要附加许多诸如 MDK、VFWDK 等外挂开发工具,因此效率较低。基于 Windows 95 和 Windows NT 的 Visual C++ 则集成了 MDK、VFWDK 等诸多开发工具的头文件和库函数,特别是 Visual C++ 4.x 以上的编译器 MSDEV,它为开发者提供了一个集成和共享的环境,使开发者可以方便地调用 Visual C++ 所有的头文件和库函数,以及界面、菜单等的设计工具,还有调试、跟踪工具等,从而大大提高了编程效率。

本书共分 5 个部分,读者完全可以根据自己的需要选择阅读,也可根据自己的编程需要参考其中的程序。

由于译者水平有限,书中疏漏之处在所难免,欢迎读者批评指正。

译者

作者序

随着 Windows 95 的成功,用户对每个应用程序的图形用户接口(GUI)的复杂性期望更高了。作为一名程序员,你必须为这种改进的 GUI 付出的代价,即所需要的程序设计工作量将显著增加。

Windows 95 和 Windows NT 程序要比与其相似的 Windows 3.1 程序庞大并且复杂得多。由于它们更加错综复杂,因此程序员在设计 Windows 95 的各种应用程序时,必须花更多的时间用于学习。为了帮助 Windows 程序员,Microsoft 公司已经推出了更加完善的 Visual C++ 开发环境。在推出 Visual C++ 4 时,Microsoft 公司重新设计了程序设计环境,并创建了 Developer Studio,更广泛地使用了 AppWizard(应用向导),增强了 ClassWizard(类向导)的集成度,并添加了能与预先建立的解决方案更容易集成的 Component Gallery(构件库)。这些实用程序使得创建高级 Windows 程序更加简单。Microsoft 公司和其他一些公司一起扩展了 Developer Studio 的功能,使它比任何时候都更加容易地完成创建和调试改进的 GUI 应用程序的任务。例如,如果出现了调试问题,就会进入 Nu-Mega 的 BoundsChecker 中;如果要寻找另外的预建立构件,就可进入 Visual Component 电子表格构件或者图表构件中。使用公式如下:

Visual C++ + Reuse(重用) + Leverage(协调) = World Class Application(世界级应用程序)

1. 这本书适合你吗

本书适用的读者是那些已熟悉了 C++、Windows 以及 Windows 基本程序设计知识的人。已了解了 Microsoft Foundation Classes(MFC,微软基本类库)的预备知识则更好,但这并不是必需的。本书包含许多使用 C++和 MFC 的程序设计的题目,如果你只用 Visual C++ 产品的 C 编译器部分,那么本书就不适合你!因为在任何一本书中,都不可能包含 Visual C++ 的每个方面和 MFC 库中的大量有用信息。

2. 硬件与软件要求

正如书名指出的那样,本书是为正在使用 Visual C++ 4 的程序员编写的。为了使用 Visual C++ 4,必须先运行 Windows 95、Microsoft Windows NT 3.51 或更新的操作系统。因此,一定要有一台功能很强的计算机,它的最低要求是带有 16M RAM 的 486 微处理器、400M 的硬盘、SVGA 图形适配器和鼠标。

3. 本书的主要内容

本书共 18 章。下面简要介绍一下各章的内容。

第一部分“Visual C++ 的各种工具”,由第 1 章至第 4 章组成。这一部分提供了使用 Visual C++ Developer Studio、AppWizard、ClassWizard 和调试工具时需要的基础知识,并介绍了

集成的各种 AppStudio 实用程序。另外,本书在这一部分还介绍了 MFC 库的各种类。

第 1 章“Visual C++ 入门”,能帮你迅速掌握 Visual C++ 开始编程工作。你将会学习到如何安装编译器,如何使用编辑器、工具箱、Help 系统以及集成的联机帮助系统。借助于 AppWizard,即可着手编写程序。

第 2 章“使用 Visual C++ Developer Studio”,讨论新的 Developer Studio 的各种选项。这些选项可用于管理文件、编辑文本、浏览信息、管理多种工程文件、浏览程序的声明和类定义部分、引导调试过程、调用各种程序设计工具以及协调各个构件(包括编译器、连接器、资源编译器以及管理 Developer Studio 环境)的运行。

第 3 章“使用 AppStudio”,这一章的主要内容是使用集成资源编辑器创建并编辑各种程序资源,包括菜单、对话框、图标、位图和字符串表。这一章还展示了如何用资源编辑器修改一个现有的菜单资源、添加一个新的对话框资源以及创建一种新形式的窗体视图。

第 4 章“调试 Visual C++ 应用程序”,这一章介绍了 Visual C++ Developer Studio 的调试器,并讨论了如何设置断点以及如何使用条件断点和无条件断点。另外,在这一章还可以看到如何使用函数调用堆栈,在 C++ 语句中混合进汇编程序代码,以及管理新的显示窗口。

第二部分“使用 MFC”,它由第 5 章和第 6 章组成。这一部分介绍了支持多种可视程序设计控件的 MFC 库(Microsoft Foundation Class Library)中的类,如命令按钮、编辑框和列表框。

第 5 章“介绍 MFC”,展示了 AppWizard 创建的代码以及它是如何精确地工作的,还介绍了组成 MFC 库的各种类,并介绍了如何给程序添加自定义功能。

第 6 章“使用 MFC”,利用第 5 章中产生的 MFC 应用程序框架构造一个真正的 Windows 应用程序:一个简单的文本文件浏览程序。该程序展示了如何让 MFC 应用程序读取文本文件并显示,如何处理滚动及处理具有支持打印预览的打印。

第三部分“对象链接和嵌入(OLE)”由第 7 章至第 9 章组成。这一部分介绍了有关利用 OLE 2 技术编程的内容、OLE 的构件以及如何编写能够实现 OLE 的程序。

第 7 章“介绍 OLE”讨论 OLE 2 的特点。本章介绍组成 OLE 的各种技术,以及 OLE 在 Microsoft 构件软件计划中的作用。

第 8 章“OLE 和 MFC”给出 OLE 容器(container)和服务器的应用程序的代码。你可以学会如何使用 AppWizard 给程序添加 OLE,同时还会发现如何修改 AppWizard 创建的程序代码的方法。

第 9 章“OLE 自动化”集中说明 OLE 的一部分——OLE 自动化。OLE 自动化提供一种从应用程序外部控制该应用程序的对象的方法。你将学会怎样用 OLE 自动化对其他应用程序进行控制,并学会如何将程序功能显示给其他服务器。

第四部分“自定义控件”由第 10 章和第 11 章组成。这一部分介绍如何用 Visual C++ 创建 OLE 自定义控件。

第 10 章“使用 ControlWizard”,指导你使用最新的 Visual C++ 向导生成 OLE 自定义控件程序框架。

第 11 章“创建 OLE 控件”,一步步介绍向第 10 章中生成的自定义控件程序框架内添加功能时编写 OLE 自定义控件的过程。

第五部分“高级 Windows 程序设计”包括第 12 章至第 18 章。这一部分展示了如何利用 Visual C++ 的高级特性,其中包括异常处理、模板、存储器管理、数据库应用以及高级调试技术。

第 12 章“针对交叉平台的开发”,说明从 Win16 到 Win32 的飞跃,以及如何在 Macintosh 和 UNIX 操作系统上为 Win32 编写可移植的应用程序。

第 13 章“异常处理”集中介绍 Microsoft 公司已经向 Visual C++ 添加的新的 C++ 异常处理能力。

第 14 章“掌握模板”集中讲述 Microsoft 公司向 Visual C++ 添加的新 C++ 模板的能力。

第 15 章“内存管理技术”阐述如何对新的 Win32 平面存储模型进行程序设计。阅读本章后,你将能在 C 内存地址分配和 C++ 内存地址分配之间作出选择,并了解 OLE 是如何使用存储器的。

第 16 章“高级调试技术”描述一些在 Visual C++ 中可以使用的集成调试程序的高级特性,同时提供有关正确设计应用程序方面的一些技巧和为避免使用高级调试技术的要求。

第 17 章“在 Visual C++ 中编写 ODBC 和 DAO 应用程序”告诉你如何访问在 Visual C++ 和 MFC 库中内带的开放数据库连接标准(ODBC)。

第 18 章“运行时刻类型标识(RTTI)”展示 RTTI 如何能够提供多种重要服务。这些服务能使你的程序更具生命力以及良好的可移植性。

我们希望本书能帮助你在工作中用 Visual C++ 建立基于 MFC 的 Windows 应用程序。努力进行你的程序设计吧,祝你好运!

作者简介

Chane Cullens 是 Bristol Technology's Wind/U 的产品经理。Chane 有着 12 年多的 C++ 编程经验,其中不但包括 Windows 和 Motif 中的 C++ 程序,还包括一些可移植的 API。他还是《Cross Platform Development With Visual C++》一书的作者,并在《Dr. Dobb's Journal》杂志上发表过多篇 C++ 编程技术的文章。可在 Internet(因特网)上以 Chane@Bristol.Com 地址与 Chane 交流。

Mark Davidson 曾开发过 CP/M 系统上的 Turbo Pascal,并在 Turbo C 1.0 之后一直在使用 Borland C++ 编程。自 Windows 2.0 推出以来,他就一直在撰写有关 Windows 应用的文章。目前,他在为索尼电子出版公司(Sony Electronic Publishing)工作,他的工作是开发 Windows 及 Macintosh 的各种应用程序。

Chris Corry 是位于美国弗吉尼亚州的美国清晰传真管理系统高级技术中心的负责人。尽管他的专业兴趣非常广泛,但他还是把主要精力用在研究面向对象的程序设计技术、软件构件标准 OLE 和 OpenDOC 以及分布式对象技术。他是《Killer Borland C++4》和《Special Edition Using Visual C++2》的作者。这两本书由 Que 公司出版。他的另一本书《OS/2 Unleashed》,由 Sams 公司出版。

Paul Robichaux 自 1986 年以来,一直是一位软件研究人员。1986 年,他就是 Internet 的一个用户。目前,他是 Intergraph Corporation 的软件顾问。他撰写了一些关于 Windows NT 和 Windows 95 应用的文章。他还在业余时间著书以及写一些有关 Macintosh 应用的文章。你可以按 Internet 地址 Perobich@ingr.com 发 e-mail 与他交流。

Steve Potts 是在乔治亚技术学院计算机科学系获得学位的。12 年来,他一直从事设计并编写软件系统的工作。他是一位基于 Windows 技术的顾问,并在美国亚特兰大的乔治亚市拥有一家计算机教育公司 NoBoredom Classes。

Kate Gregory 主要从事 Visual C++ 研究工作(专门研究 Internet 应用程序),讲授 Internet 和 C++ 程序设计课程,并编写这些方面的书籍。在加拿大郊区, Kate 和她的丈夫 Brian 用 Visual C++ 编写 Internet 应用程序,他们的新项目是一个新闻阅读器(newsreader)程序。她还在讲授 Internet 和 C++ 程序设计课程,本书是 Kate 为 Que 公司写的第二本书。她在互联网上的地址是 Kate@gregcons.com。

目 录

第一部分 Visual C++ 的各种工具

第 1 章 Visual C++ 入门	3
1.1 Visual C++ 简介	3
1.2 新版本的主要特点	4
1.3 系统需求	5
1.4 安装 Visual C++	5
1.4.1 运行 Setup 程序	6
1.4.2 使用 Custom 安装	8
1.5 Visual C++ 介绍	10
1.5.1 启动 Developer Studio	10
1.5.2 使用 Editor	11
1.5.3 使用 Search 选项	11
1.5.4 使用工具栏	12
1.5.5 获取联机帮助	13
1.6 使用项目文件	14
1.7 利用 AppWizard 创建应用程序	15
1.8 配置选项	16
1.9 小结	17
第 2 章 使用 Visual C++ Developer Studio	19
2.1 装入 Developer Studio	19
2.2 菜单系统	19
2.3 File 菜单	21
2.3.1 New 命令 (Ctrl+N)	21
2.3.2 Open 命令 (Ctrl+O)	22
2.3.3 Close 命令 (Ctrl+F4)	23
2.3.4 Open Workspace 命令	23
2.3.5 Close Workspace 命令	23
2.3.6 Save 命令 (Ctrl+S)	23
2.3.7 Save As 命令	23
2.3.8 Save All 命令	23
2.3.9 Find in Files 命令	24
2.3.10 Page Setup	24
2.3.11 Print 命令 (Ctrl+P)	25
2.3.12 Exit 命令 (Alt+F4)	25
2.4 Edit 菜单	25
2.4.1 Undo 命令 (Ctrl+Z)	26
2.4.2 Redo 命令 (Ctrl+Y)	26
2.4.3 Cut 命令 (Ctrl+X)	26

2.4.4	Copy 命令 (Ctrl+C)	26
2.4.5	Paste 命令 (Ctrl+V)	27
2.4.6	Delete 命令 (Del)	27
2.4.7	Select All 命令	27
2.4.8	Find 命令 (Ctrl+F3)	27
2.4.9	Replace 命令	29
2.4.10	Go To 命令 (Ctrl+G)	29
2.4.11	InfoViewer Bookmarks 命令	30
2.4.12	Bookmarks 命令	30
2.4.13	Breakpoints 命令 (Ctrl+B)	31
2.4.14	Properties 命令 (Alt+Enter)	32
2.5	View 菜单	32
2.5.1	ClassWizard 命令 (Ctrl+W)	32
2.5.2	Resource Symbols 命令	32
2.5.3	Resource Includes 命令	33
2.5.4	Full Screen 命令	34
2.5.5	Toolbars 命令	34
2.5.6	InfoViewer Query Results 命令	34
2.5.7	InfoViewer History List 命令	34
2.5.8	Project Workspace 命令 (Alt+O)	35
2.5.9	InfoViewer Topic 命令 (Alt+1)	35
2.5.10	Output 命令 (Alt+2)	36
2.5.11	Watch 命令 (Alt+3)	36
2.5.12	Variables 命令 (Ctrl+4)	36
2.5.13	Registers 命令 (Alt+5)	36
2.5.14	Memory 命令 (Alt+6)	36
2.5.15	Call Stack 命令 (Alt+7)	36
2.5.16	Disassembly 命令 (Alt+8)	36
2.6	Insert 菜单	37
2.6.1	File 命令	37
2.6.2	Resource 命令 (Ctrl+R)	37
2.6.3	Resource Copy 命令	38
2.6.4	Files into Project 命令	38
2.6.5	Project 命令	39
2.6.6	Component 命令	39
2.7	Build 菜单	39
2.7.1	Complie 命令 (Ctrl+F8)	39
2.7.2	Build<your app. exe>命令 (shift+F8)	40
2.7.3	Rebuild All 命令 (Ctrl+F8)	40
2.7.4	Batch Build 命令	40
2.7.5	Stop Build 命令 (Ctrl+Break)	40
2.7.6	Update All Dependencies 命令	41
2.7.7	Debug 子菜单	41
2.7.8	Execute<your app. exe>命令 (Ctrl+F5)	41

2.7.9	Settings 命令	41
2.7.10	Configurations 命令	42
2.7.11	Subprojects 命令	42
2.7.12	Set Default Configuration 命令	42
2.8	Debug 菜单	42
2.8.1	Go 命令(F5)	43
2.8.2	Restart 命令(Shift+F5)	43
2.8.3	Stop Debugging 命令(Alt+F5)	43
2.8.4	Break 命令	43
2.8.5	Step Into 命令(F8)	43
2.8.6	Step Over 命令(F10)	43
2.8.7	Step Out 命令(Shift+F7)	43
2.8.8	Run to Cursor 命令(F7)	43
2.8.9	Step Into Specific Function 命令	44
2.8.10	Exceptions 命令	44
2.8.11	Threads 命令	44
2.8.12	Settings 命令	45
2.8.13	QuickWatch 命令(Shift+F9)	45
2.9	Tools 菜单	45
2.9.1	Browse 命令(Alt+F12)	46
2.9.2	Close Browse Info 命令	47
2.9.3	Profile 命令	47
2.9.4	Remote Connection 命令	47
2.9.5	Customize 命令	48
2.9.6	Options 命令	49
2.9.7	Record Keystrokes 命令(Ctrl+Shift+R)	50
2.9.8	Playback Recording 命令	50
2.10	Window 菜单	50
2.10.1	New Windows 命令	50
2.10.2	Split 命令	50
2.10.3	Hide 命令(Shift+Esc)	50
2.10.4	Cascade 命令	50
2.10.5	Tile Horizontally 命令	50
2.10.6	Tile Vertically 命令	50
2.10.7	Close All 命令	51
2.10.8	Windows 命令	51
2.11	Help 菜单	51
2.12	小结	51
第3章	使用 AppStudio	52
3.1	Project Workspace 资源窗口	52
3.2	资源编辑器生成的文件	54
3.3	修改菜单资源	54
3.3.1	创建 Menu03 应用程序	55
3.3.2	编辑菜单资源	55

3.3.3	加入成员函数	56
3.3.4	修改代码	57
3.4	加入新的对话框资源	62
3.5	创建图标和对话框资源	62
3.5.1	创建新类	64
3.5.2	用户自定义代码	65
3.6	创建窗体视图	71
3.6.1	创建 Ch3_form.exe 应用程序	71
3.6.2	窗体视图窗口中加放各种控件	72
3.6.3	增加函数和成员	73
3.6.4	定义代码	73
3.7	小结	79
第 4 章	调试 Visual C++ 应用程序	81
4.1	Visual C++ 4 新增加的调试功能	81
4.2	Debug 菜单命令回顾	82
4.3	关于调试程序	83
4.4	设置断点	84
4.4.1	位置断点	85
4.4.2	数据断点	86
4.4.3	数组断点	87
4.4.4	指针断点	88
4.4.5	地址断点或寄存器断点	88
4.4.6	条件断点	89
4.4.7	消息断点	89
4.5	调试程序实例	90
4.5.1	使用无条件断点	90
4.5.2	使用带表达式的断点	92
4.5.3	查看调用栈	93
4.6	查看混合的源代码和汇编代码	94
4.7	管理 Watch 窗口	95
4.7.1	增加和删除表达式	95
4.7.2	变量的展开与收缩	95
4.8	小结	96
第二部分 使用 MFC		
第 5 章	介绍 MFC	99
5.1	MFC 概述	99
5.1.1	MFC 是如何提供的	100
5.1.2	MFC 类的本质	101
5.2	用 AppWizard 生成应用程序	102
5.3	AppWizard 到底生成了什么	110
5.3.1	Make 文件(FileView.mak)	111
5.3.2	主头文件(FileView.h)	114
5.3.3	应用程序类的源文件(FileView.cpp)	116

5.3.4	主窗口头文件(mainfrm.h)	121
5.3.5	主框架源文件(MainFrm.cpp)	123
5.3.6	文档类源文件(FileVdoc.h)	127
5.3.7	视图类源代码(FileVWw.h 和 FileVWw.cpp)	130
5.3.8	子窗口类(CChildFrame)	135
5.3.9	MFC 文件(#include)	137
5.3.10	资源标识(resource.h)	138
5.3.11	缺省资源文件(FileView.rc)	138
5.3.12	资源目录(RES)	149
5.3.13	用于 ClassWizard 的信息文件(FileView.clw)	149
5.3.14	建立联机帮助文件(Makehelp.bat)	149
5.3.15	帮助项目文件(FileView.hpj)	150
5.3.16	帮助目录(HLP)	150
5.4	进一步完成 FileView	151
5.5	小结	151
第 6 章	使用 MFC	152
6.1	规划 FileView 应用程序	152
6.1.1	你想让 FileView 干什么	152
6.1.2	文档和视图究竟是什么	153
6.2	使 FileView 的文档类读文件	153
6.2.1	理解缺省的串资源	154
6.2.2	将文本文件读入内存并建立文档	155
6.3	让 FileView 视图类显示文本	160
6.3.1	提示用户打开文件	163
6.3.2	让视图能够滚动	163
6.4	改进打印预览	171
6.5	小结	177
第三部分 对象链接与嵌入(OLE)		
第 7 章	介绍 OLE	181
7.1	OLE 2 是什么	181
7.1.1	以文档为中心的计算	181
7.1.2	构件软件	182
7.1.3	为什么 OLE 如此重要	182
7.1.4	OLE 的历史	182
7.1.5	OLE 2 为用户带来的好处	183
7.1.6	OLE 2 为开发人员带来的好处	183
7.1.7	OLE 2 的不足	184
7.2	OLE 技术定义	184
7.2.1	Windows 对象:COM	186
7.2.2	结构化存储	187
7.2.3	复合文件	187
7.2.4	拖放支持	188
7.2.5	一致性数据传送	189

7.2.6	链接	190
7.2.7	嵌入	190
7.2.8	Visual Editing(可视编辑)	191
7.2.9	OLE Automation(OLE 自动化)	192
7.2.10	OLE 自定义控件	193
7.3	使用 OLE 2 技术	193
7.3.1	字处理器	193
7.3.2	电子数据表格	193
7.3.3	数据库	193
7.4	比较 OLE 2 和 OLE 1.0	194
7.5	OLE 2 定义的术语	194
7.5.1	对象(Object)	194
7.5.2	容器(Container)	194
7.5.3	激活(Activation)	194
7.5.4	服务器(Server)	195
7.5.5	完整服务器(Full Servers)	196
7.5.6	小型服务器(Miniservers)	196
7.5.7	整理(Marshaling)	196
7.5.8	接口(Interface)	196
7.6	Visual C++ 4 中的 OLE 2 工具	197
7.7	OLE 2 注意事项	198
7.8	OLE 用户接口	198
7.9	登记 OLE 应用程序	198
7.9.1	系统注册表(注册表)	198
7.9.2	注册表编辑器	199
7.10	小结	200
第 8 章	OLE 和 MFC	201
8.1	为什么要用 MFC 编写 OLE	201
8.2	MFC 提供了什么	201
8.3	OLE 和 AppWizard	202
8.4	MFC 对 OLE 2 的支持	204
8.4.1	OLE 基类	204
8.4.2	可视编辑类	204
8.4.3	数据传送类	205
8.4.4	标准对话框类	205
8.4.5	自定义控件类	206
8.4.6	其他 OLE 类	206
8.5	用 MFC 创建 OLE 容器	206
8.5.1	初始化应用程序	206
8.5.2	打开 OLE 对话框	207
8.5.3	显示 OLE 对象	207
8.5.4	保存和装入 OLE 信息	208
8.5.5	简单的容器样本	208
8.6	用 MFC 创建 OLE 服务器	219

8.6.1	初始化服务器	220
8.6.2	绘制服务器条目	220
8.6.3	简单的服务器样本	220
8.7	小结	223
第9章	OLE 自动化	224
9.1	介绍 OLE Automation	224
9.1.1	OLE Automation 服务器	224
9.1.2	OLE Automation 客户	225
9.1.3	OLE Automation 的优点	225
9.2	在 MFC 中创建 Automation 服务器	225
9.2.1	表述属性	226
9.2.2	表述方法	227
9.3	OLE Automation 样本程序	227
9.3.1	服务器应用程序类	227
9.3.2	服务器文档类	227
9.3.3	服务器 Automation 对象	230
9.3.4	ODL 文件	231
9.4	将 DispTest 用作 OLE Automation 客户	232
9.4.1	OLE Automation 对象	233
9.4.2	从文件中激活 OLE Automation 对象	233
9.5	OLE Automation 及展望	234
9.6	小结	235

第四部分 自定义控件

第10章	使用 Control Wizard	239
10.1	OLE 控件	239
10.1.1	什么是 OLE 控件	239
10.1.2	为什么要使用 OLE 控件	240
10.2	简单的控件	240
10.2.1	Calen.h 文件	244
10.2.2	Calen.cpp 文件	245
10.2.3	Calen.rc 文件	246
10.2.4	Calen.def 文件	249
10.2.5	CalenCtl.h 文件	249
10.2.6	CalenCtl.cpp 实现文件	251
10.2.7	CalenPpg.h 头文件	255
10.2.8	CalenPpg.cpp 文件	256
10.2.9	StdAfx.h 文件	257
10.2.10	StdAfx.cpp 文件	258
10.2.11	Resource.h 头文件	258
10.2.12	Calen.odl 类型库文件	259
10.3	为控件增加许可权支持	260
10.3.1	修改过的 CalenderCtl.h 头文件	261
10.3.2	修改过的 CalenCtl.cpp 实现文件	263

10.4	创建 Calen 控件	267
10.5	Test Container 应用程序	267
10.6	小结	269
第 11 章	创建 OLE 控件	271
11.1	空的外壳	271
11.2	OCX 属性	272
11.2.1	基础属性	272
11.2.2	包容属性	273
11.2.3	扩展属性	274
11.2.4	自定义属性	274
11.3	绘制日历	276
11.3.1	添加属性	276
11.3.2	填写 Get 和 Set 函数	279
11.3.3	初始化和稳定性	283
11.3.4	绘制控件	285
11.3.5	处理属性的变化	293
11.3.6	小规模外观改进	295
11.4	属性页	295
11.4.1	基础属性页	296
11.4.2	通用属性页	297
11.4.3	自定义属性页	298
11.5	方法和事件	300
11.5.1	设置日期	300
11.5.2	添加基础事件 Chick()	301
11.5.3	增加自定义事件 Select()	301
11.6	转换 VBX 控件	304
11.7	小结	305

第五部分 高级 Windows 程序设计

第 12 章	针对交叉平台的开发	309
12.1	高级程序设计的交叉平台工具	309
12.2	应用程序的结构	311
12.3	将 Win16 MFC 应用程序移植到 Win32 MFC	312
12.3.1	Windows 32 操作系统扩展(Win32s)	313
12.3.2	使用 MFC 编程的可移植性	323
12.4	用 Win32 API 和 MFC 设计的可移植程序	324
12.4.1	Intel x86、Motorola 680x0 和 RISC 之间的差异	325
12.4.2	结构封装和存储分配边界	326
12.4.3	内存分配	328
12.4.4	文件系统的不同	329
12.5	把应用程序移植到 Macintosh	332
12.6	把 Visual C++ 应用程序移植到 UNIX 上	333
12.6.1	对 Wind/U 体系结构的理解	333
12.6.2	转换文件	334

12.6.3	创建 UNIX 的 Make 文件	335
12.6.4	修改 DIBLOOK 文件	335
12.6.5	编译 DIBLOOK 文件	337
12.6.6	运行 DIBLOOK	337
12.6.7	STL 与 MFC	337
12.7	小结	338
第 13 章	异常处理	340
13.1	在不完美的世界中进行软件开发	340
13.1.1	异常处理:通往可靠性之路的里程碑	341
13.1.2	C 中的错误捕获	341
13.1.3	返回错误值	341
13.1.4	以函数参数返回错误值	343
13.1.5	全局错误变量	343
13.1.6	使用 goto 或者 setjmp/longjmp	344
13.1.7	异常处理是更好的方法	346
13.2	异常处理的三个步骤:抛出、捕获及重试	346
13.2.1	检查 try 块	348
13.2.2	捕获异常	351
13.2.3	你正在抛出什么	358
13.3	安装你自己的 terminate() 处理程序	362
13.4	看看一个虚拟的应用程序	363
13.4.1	对于 ExpSum 应用程序的介绍	363
13.4.2	ExpSum 应用程序的结构	365
13.4.3	ExpSum1:花费报告的模型 T	371
13.4.4	ExpSum2:消费报告的费拉里(Ferrari)	375
13.4.5	包装 ExpSum 应用程序	383
13.5	启用和禁止 Visual C++ 的异常处理	383
13.6	MFC 异常支持概述	384
13.7	小结	385
第 14 章	掌握模板	386
14.1	使用模板的理由	386
14.2	理解通用函数和通用类	387
14.2.1	源代码的重用和效率	388
14.2.2	类型的安全性	388
14.2.3	模板与函数类家族	389
14.3	函数模板	390
14.4	为什么模板不能对什么类型都起作用	391
14.5	构造类模板	393
14.5.1	模板语法	393
14.5.2	定义模板	394
14.5.3	实例化模板	398
14.5.4	使用静态模板成员数据和方法	400
14.5.5	常量表达式模板参数	403
14.5.6	一些语法上易犯的错误	404