

Visual C++™ 技术内幕

(第二版)

[美] David J. Kruglinski

王图邨 译

清华大学出版社

(京)新登字 158 号

Visual C++ 技术内幕(第二版)

Inside Visual C++, Second Edition, Version 1.5

David J. Kruglinski

Copyright 1994 by Microsoft Corporation.

Original English Language Edition Copyright © 1994 by David J. Kruglinski.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文版由 Microsoft Press 授权清华大学出版社独家出版。

中华人民共和国国家版权局著作权合同登记章 图字: 01-95-302 号

未经出版者书面允许,不得以任何方式复制或抄袭本书的内容。

本书封面贴有 Microsoft Press 激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Visual C++ 技术内幕/(美) Kruglinski; 王国印译. 2 版. —北京:清华大学出版社, 1995

ISBN 7-302-02015-9

I. V… II. ①K… ②王… III. C 语言-视窗(软件)-操作系统 IV. TP316

中国版本图书馆 CIP 数据核字(95)第 19583 号

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

印刷者: 清华大学印刷厂

发行者: 新华书店总店北京科技发行所

开本: 787×1092 1/16 印张: 37.5 字数: 890 千字

版次: 1996 年 5 月第 2 版 1996 年 5 月第 1 次印刷

书号: ISBN 7-302-02015-9/TP·932

印数: 0001—5000

定价: 73.00 元

译者前言

本书的第一版主要是针对 Visual C++ 1.0 的,而这一版主要是针对 Visual C++ 1.5 的。由于 Visual C++ 1.5 相对于 Visual C++ 1.0 来说又作了不少改进,增加了不少新的特性,Visual C++ 1.5 所包含的 MFC 2.0 也在 Visual C++ 1.0 所包含的 MFC 1.5 的基础上作了许多改进和补充,因此,作者在本版中所作的修改几乎遍布全书的各个角落。尤其值得一提的是,本书的作者重新编写了第二十四章和第二十五章,由此可见,作者对这两部分内容的重视。的确,随着社会对数据库应用需求的日益升温,ODBC 越来越显示出了其重要性和实用性;随着 OLE 2.0 的推出,它以其卓越的性能,引起了人们的广泛重视,人们渴望掌握它的热情也日益高涨。在这种情形下,本书第二版的面世,对于广大程序员来说,无疑是雪中送炭。

本版基本上保持了第一版的结构和风格,除针对 Visual C++ 1.5 对第一版作了相应的补充和修改外,还纠正了第一版中所出现的一些错误。另外,在本版中,译者还纠正了在上一版中出现的一些翻译错误和印刷错误。随着 Visual C++ 新的版本的不断推出,作者还将出第三版、第四版,到时,译者将不遗余力继续将它的续版的译本奉献给大家。

译者

1995 年 5 月 1 日于北京

前言

一种全新类型的软件产品的诞生并不是常有的事,而“应用框架”(application framework)就属于一种全新类型的软件产品。Visual C++包含了迄今为止功能最强大的基于 Windows 的应用框架,这一点是毫无疑问的。同时,这一产品又是真实可靠的,因为它来自 Windows 本身的开发者 Microsoft。尽管 Microsoft Visual C++ 应用框架同读者使用过的任何软件产品都有很大的差别,但它所依赖的一些基本元素却是读者早已非常熟悉的,包括 C++、适合于 C 语言的 Windows 软件开发包(SDK)以及 Microsoft C/C++ 7.0 版所提供的最初的 Microsoft 基本类库 1.0 版。

Microsoft 基本类库 2.5 版(为了简单起见,常将它简称为“类库”)是 Visual C++ 1.5 的非常重要的一部分,它构成了应用框架的核心。类库中包含了 C++ 类以及全程函数,并且提供了相应的源代码。Visual C++ 的其它一些组成元素,包括 AppWizard、ClassWizard、App Studio、Visual 工作平台、编译器和连接器,所有这些都为用户构造自己的应用提供了必要的工具。

本书不但介绍了类库中的类,还详细说明了如何使用这些类和工具来创建基于 Windows 的应用。如果读者手头已有 Visual C++,那么《Visual C++ 技术内幕》一书将为读者提供产品文档中不曾包含的许多非常有用的技术、观点、例子以及理论。如果读者正打算购买 Visual C++ 的话,那么本书足以使您了解该产品的全部功能。

本书的适用对象

经过了一段不太令人满意的使用 Windows 的 SDK 进行编程的尝试之后,我开始学习使用类库 1.0 版。我深深地体会到 C++ 和 Windows 结合得非常自然,使用 C++ 来学习基于 Windows 的程序设计实际上还要容易一些。那么我们为什么不用 C++ 的方式来教人们进行基于 Windows 的程序设计?为什么不假设读者已经具备了编程经验,可以直接跳过使用 SDK 的程序员所必须学习和掌握的一些令人讨厌的琐碎细节呢?

我的编辑们也同意我的这些观点,不过他们还说:“不要忽视了那些熟读了所有 Petzold 所著书籍的熟练 Windows 程序员”。我回答他们说“好的”,因此,我力求使本书适合于所有程序员,不管他们是否具有 Windows 编程经验。

再一个就是有关 C++ 的基本知识问题。外面已经有许多关于 C++ 的书籍,大概所有的人都知道有一种编程语言叫“C++”。“每个人都至少买了一本关于 C++ 的书”,我想这句话恐怕一点不假。或许读者也确实读过有关 C++ 的书,也尝试过一些例子,然而不久就对 C++ 失去了兴趣,重新又返回到了 C。Visual C++ 应用框架能使读者真正对 C++ 非常精通,本书将有助于读者做到这一点。

作为本书的作者,我可以假设读者不懂 Windows,但是很难假设读者不懂 C++。例如,只要给出了椭圆函数的函数原型,任何程序员都很容易写一个过程在屏幕上画出一个

椭圆,该过程可以是一个 C 函数,也可以是 C++ 的成员函数。但是如果程序员不懂得 C++ 的类和对象,那么他或她就会遇着麻烦。为此,我在附录 A 中对 C++ 作了一个回顾。如果读者还未接触过 C++, 建议最好还是首先阅读附录 A, 同时手中还需要有一些其它的关于 C++ 的书,读者最后也许不得不仔细阅读这些书。

也许读者已经注意到了,我讲了这么多都是针对程序员而言的。的确,本书的读者最好是一名程序员,至少应该是正在学习程序设计的学生。近年来各种编译器、工具以及操作系统都变得越来越复杂,几乎不可能仅通过一本书就能使读者从一个门外汉变成熟练的 Windows 程序设计者。本书至少要求读者具备 C 语言的知识,因为甚至连附录 A 都要读者能看懂 C 代码。

对了,我差点忘了,本书还要求读者懂得如何运行基于 Windows 的应用。如果读者连这最起码的知识都不具备,如何谈得上设计和编写基于 Windows 的应用呢? 如果读者希望寻找一个具体应用来作为学习的开始,那么我建议最好从使用 Microsoft Word for Windows 开始。该程序不但是一个非常好的基于 Windows 应用的例子,同时也是一个非常不错的字处理器,读者可以用它来编写一些帮助文件,我就是用它来编写本书的。

如何使用本书

当读者开始学习使用 Visual C++ 的时候,最好首先从头到尾阅读此书,它会给您提供最好的学习指导。然后读者就可以将此书作为一本参考手册,通过目录及索引来经常查阅一些内容。由于许多应用框架元素之间是彼此紧密相关的,许多概念也都是彼此联系的,因此本书不可能象百科全书那样包罗万象。在使用本书时,至少还需要一本《类库参考手册》,以供随时参考。

本书的组织

正如本书的目录中所列出的那样,本书主要包括四大部分:

第一部分: Windows、Visual C++ 和应用框架基础

在这部分中,既有理论阐述,又有实际应用,我力求使这两者达到一种平衡。在这部分中,我对现代 Windows 作了一番总结与回顾,并介绍了 Visual C++ 的基本组成,进而逐步将应用框架以及文档—视图结构(document—view architecture)展现在读者面前。同时,在这部分中还给出了一个通过类库中的类来构造的简单的“Hello world!”应用,该程序只需 30 行代码。

第二部分: 类库中的视类

在假定读者已经熟悉使用 Windows 的 SDK 来进行编程的前提下,类库文档材料简捷而又完整地介绍了应用框架的基本元素。在本部分中我们将把读者的注意力集中在应用框架的主要构成元素——“视”(view)上,“视”实际上是一个窗口。在这里我们将从 C++ 和类库中的类的角度来学习和掌握 SDK 程序员早已熟知的有关窗口的一些特性。对于熟练的 Windows 程序员来说这部分也是非常值得一读的,因为类库视环境支持一些新的特性,如对话框数据交换、图形按钮以及 Visual Basic 控制等。在这部分中读者将更多地使用 Visual C++ 工具,而这些工具极大地减少了 SDK 程序员在编程时无法避免的

一些繁杂工作。

第三部分：文档—视结构

文档—视结构构成了应用框架的核心。在这部分中读者将会了解什么是“文档”(document)(在这里“文档”并不是指文字处理工作中通常所说的文档,它有更进一步的含义),也将学习如何将文档同第二部分中所介绍的视联系起来。一旦读者动手编写出了自己的文档类,就会禁不住对类库简化文件 I/O 及打印的方式感到非常的惊讶。

另外,读者在这部分中也会接触到命令消息处理(command message processing)、工具条(tool bar)、状态条(status bar)、切分框架(splitter frame)以及上下文相关帮助(context-sensitive help)。同时,读者还将学习 Windows 的多文档接口(MDI),这是类库应用的一个非常重要的特性。

读者须特别注意第三章 3.9 节“加速创建过程”这部分内容,因为它将有助于节省读者对书中例子的处理时间。

第四部分：其它高级话题

这部分介绍了许多类库直接支持的非常有用的 Windows 程序设计技术。本部分首先介绍了几个基于 Windows 的应用,这些应用都绕过了文档—视结构,还介绍了一个与设备无关的位图类。在第二十四章读者将看到一个应用 Microsoft 开放数据库连通性(Microsoft Open Database Connectivity)(ODBC)程序接口的例子。本书的第二十五章介绍了 OLE,第二十六章介绍了类库风格的动态连接库(DLLs)。

对 Windows 的进一步探讨：“致 SDK 程序员”这一副标题的目的

本书并没有给出最近一些新的关于 Windows 的专门书籍中所提供的一些技巧和隐含特性。这些书籍大都是针对使用 C 语言的 SDK 程序员来写的。为了能有效地使用这些书籍,读者必须理解和掌握 SDK 应用程序接口(API)以及它和类库之间的关系。

书中“致 SDK 程序员”这一副标题可帮助读者将类库同 Windows 的 SDK 联系起来。这些特殊格式的方框中所包含的内容能够帮助熟练的 Windows C 程序员将新的类库概念同他们所熟知的 SDK 准则联系起来。如果读者还不熟悉 SDK 编程,那么第一遍读此书时可跳过这些注释,但当第二遍读此书时则应该仔细阅读这些内容,这有助于帮助读者理解关于 Windows 的一些主要文献,加速对 MFC 库的掌握。

如果读者接触过其它的应用框架

读者或许也已经注意到了其它的应用框架产品(其中最著名的当数用于 Apple Macintosh 上的 MacApp)。类库和这些产品有相似之处,同时也存在着不少差异,因此,不要想当然地去理解名字相似的一些术语或类。

对硬件的要求

开发 Windows 应用比一般任务对机器的要求要高一些,这一点读者应该明白。为了节省您的时间,最好能有一台具有 8 兆或更多随机存取存储器(RAM)的 80486 或 Pentium 机器,额外的 RAM 将用来作为磁盘缓冲存储器(disk cache)。RAM 磁盘可大大加

快编译和连接的速度,第三章将会告诉读者如何来设置和安排这些额外的 RAM。

至于磁盘空间,仅 Visual C++ 的程序本身就要占 50MB,另外每个工程都至少需要 4MB 的磁盘空间(包括预编译头文件、映射文件和浏览器数据库),况且在实际使用时免不了要建立多个工程,因此,至少需要一个 200MB 硬盘,如果能有一个 500MB 硬盘就更理想了。另外,读者还需要一个 CD-ROM 驱动器,因为 Visual C++ 1.5 版是装在 CD-ROM 上的,况且读者还可能需要直接从 CD-ROM 上阅读例程序和文档。

建议用户考虑配置一个 SVGA 大屏幕显示器。在大屏幕显示器上可以同时显示 Visual 工作平台、帮助窗口以及正在被调试的基于 Windows 的程序。

附加 CD-ROM 光盘的使用

在本书封底的内面还附带着一张附加光盘,本书中所有例程序的源文件及其 make 文件都在该光盘中。但附加盘中并没有包含可执行文件,因此,读者必须创建感兴趣的例程序的可执行文件。为了安装附加光盘中的文件,请将该光盘插入 CD-ROM 驱动器中,再运行 Setup 程序,然后只需按照屏幕提示进行操作即可。

对于传统的用 C 语言编写的 Windows SDK 程序,源程序会告诉人们一切,但对于类库应用框架来说,事情就没有这么简单了。在类库应用框架中,大部分的 C++ 源代码都是由 AppWizard 产生的,而资源则是由 App Studio 生成的。前面几章的例子大都是一步一步地教读者如何使用有关工具来生成自己的源代码文件,建议读者最好能按照规定的步骤去试几个例子,在这些例子中需要用户自己动手编写的代码是很少的。中间几章的例子可以使用附加盘所提供的源代码,但也必须仔细阅读创建过程,以便对 App Studio 及两个 Wizard 的作用有更深的体会。在最后几章中并没有列出例子的所有源代码,读者只好从附加盘上去查找这些例子的完整源代码了。

技术说明及例程序

读者可以查阅 Visual C++ 的帮助文件,其中包含了 46 个非常有用的技术说明。这些技术说明包含了许多在该帮助文件中并没有进行讨论的有关 MFC 库的一些高级特性,本书按照序号对相应的技术说明提供了进一步的参考内容。读者可以通过两种方式阅读到帮助文件中的这些技术说明:可以执行 WINHELP,然后选择 \MSVC\HELP\MFCNOTES.HLP,或者在 Visual C++ 程序管理组中在 MFC 技术说明图标上连续按两下鼠标。

读者可以在 \MSVC\MFC\SAMPLES 子目录下发现 30 个非常有用的 MFC 库例程序。 \MSVC\HELP\MFC\SAMP.HLP 帮助文件(可以直接通过 Visual C++ 程序管理组中的 MFC 例程序帮助图标访问该帮助文件)给出了有关这些例程序的说明,这些例程序说明了一些关于 MFC 库的更高级的特性,本书对这些例程序也零星作了一些说明。

Visual C++ 1.5 版

本书的第一版是针对 Visual C++ 1.0 版而写的,而本版是专门针对 Visual C++ 1.5 版而写的,其间作了大量的改动,所作的修改分散在全书的各个章节。同样,针对

AppWizard和 ClassWizard 所作的一些改进,本书在一些文字说明和屏幕图示上也作了相应的修改。在 MFC 2.5 中,增加了一些新的 ODBC 类,并且为了支持 OLE 2.0,还对 OLE 类做了相应的修改,因此,我重新编写了本书的二十四和二十五章。

有关 ODBC 和 OLE 2.0 的内容是非常复杂的,有专门的书籍分别对它们进行介绍。针对这两部分内容,本书将只就它们和 Visual C++ 有关的话题进行讨论。第二十四章只介绍了 C++ 数据库程序设计,而没有完整地讨论有关客户—服务器计算环境问题。第二十五章则重点论述了 OLE 2.0 的基本理论和 OLE 的自动化,没有涉及 OLE 2.0 的其它一些特性,如实地(in-place)编辑、连接以及拖和放等。

目 录

前言	18
第一部分 Windows、Visual C++ 和应用框架基础	
第一章 Microsoft Windows 和 Visual C++	3
1.1 Windows 的编程模式	3
1.1.1 消息处理	3
1.1.2 Windows 的图形设备接口 (GDI)	4
1.1.3 基于资源的程序设计	4
1.1.4 内存管理	4
1.1.5 动态连接库 (DLLs)	5
1.1.6 Windows NT	5
1.2 Visual C++ 的组成	5
1.2.1 Visual 工作平台和创建过程	6
1.2.2 App Studio 资源编辑器	8
1.2.3 C/C++ 编译器	8
1.2.4 连接器	8
1.2.5 资源编译器	9
1.2.6 调试器	9
1.2.7 AppWizard	9
1.2.8 ClassWizard	10
1.2.9 源程序浏览器	10
1.2.10 联机帮助	10
1.2.11 Windows 诊断工具	11
1.2.12 Microsoft 基本类库 2.5 版	11
第二章 Microsoft 基本类库应用框架	12
2.1 为什么要使用应用框架	12
2.1.1 学习曲线	14
2.2 什么是应用框架	14
2.2.1 应用框架和 MFC 库	14
2.2.2 应用框架例子	15
2.3 MFC 库消息映射	17
2.3.1 文档和视	18
第二部分 MFC 库中的视觉	
第三章 从“Hello, world!”入手来学习 AppWizard	23
3.1 什么是视	23
3.2 单文档接口 (SDI) 和多文档接口 (MDI)	24

3.3	“不做任何事情”的应用	24
3.4	CEx03aView 视类	27
3.5	在视窗内绘图——Windows 的图形设备接口	27
3.5.1	OnDraw 成员函数	27
3.5.2	Windows 设备环境	28
3.5.3	往 EX03A 程序中加入绘制代码	28
3.6	提前了解一下 App Studio, 同时介绍一下资源	29
3.6.1	EX03A.RC 的内容	29
3.6.2	运行 App Studio	30
3.7	Windows 的 Debug 内核和 DBWIN	32
3.8	需要使用调试器吗	34
3.8.1	允许诊断宏	34
3.9	加速创建过程	34
3.10	创建新的 MAK 文件	39
第四章	基本事件控制——使用 ClassWizard	41
4.1	获取用户输入——消息映射函数	41
4.1.1	消息映射	41
4.1.2	保存视的状态-类数据成员	42
4.1.3	初始化视类数据成员	42
4.1.4	使矩形区域无效	43
4.1.5	Windows 的用户区域	43
4.2	EX04A 例程序	43
4.3	对 EX04A 使用 ClassWizard	46
4.3.1	同时使用 AppWizard 和 ClassWizard	46
4.4	EX04B——通过鼠标来拖动圆	49
4.4.1	EX04B 程序的构成元素	51
4.5	滚动视窗	53
4.5.1	窗口比人们所见到的要大	54
4.5.2	滚动条	54
4.5.3	滚动方式的选择	54
4.5.4	EX04C 滚动例程序	54
4.5.5	EX04C 程序中的元素	57
4.6	其它 Windows 消息	58
4.6.1	WM_CREATE 消息	58
4.6.2	WM_CLOSE 消息	58
4.6.3	WM_QUERYENDSESSION 消息	59
4.6.4	WM_DESTROY 消息	59
4.6.5	WM_NCDestroy 消息	59
第五章	图形设备接口(GDI)	60
5.1	设备环境类	60
5.1.1	显示设备环境类 CClientDC 和 CWindowDC	60
5.1.2	构造和析构 CDC 对象	61

5.1.3	设备环境的状态	62
5.1.4	CPaintDC 类	62
5.2	GDI 对象	62
5.2.1	GDI 对象的构造与析构	63
5.2.2	跟踪 GDI 对象	63
5.2.3	库存的 GDI 对象	64
5.2.4	GDI 选择的有效期	64
5.2.5	持久的用于显示的设备环境——登记窗口类	65
5.3	Windows 的颜色映射	66
5.3.1	标准视频图形矩阵(VGA)显示卡	66
5.3.2	256 色显示卡	67
5.3.3	24 位颜色显示卡	68
5.4	映射方式	68
5.4.1	MM_TEXT 映射方式	68
5.4.2	“固定比例”的映射方式	69
5.4.3	“比例可变”的映射方式	70
5.4.4	坐标变换	71
5.5	字体	73
5.5.1	字体是 GDI 对象	73
5.5.2	选择字体	73
5.5.3	打印字体	73
5.5.4	显示字体	74
5.5.5	窗口中的点——逻辑 Twips	74
5.5.6	计算字符高度	75
5.6	EX05A 例程序	76
5.6.1	EX05A 程序的组成元素	79
5.7	EX05B 程序	79
5.8	EX05C 例程序——再次使用 CScrollView	83
5.8.1	EX05C 程序的一些元素	86
5.8.2	CScrollView 的 SetScaleToFitSize 状态	86
第六章	有模式对话框	87
6.1	有模式和无模式对话框	87
6.2	系统模式对话框	87
6.3	资源和控制	88
6.4	有模式对话框的程序设计	88
6.5	包罗万象的对话框——EX06A 例程序	89
6.5.1	ClassWizard 和对话框类	95
6.5.2	将对话框与视图连接	99
6.5.3	深入理解 EX06A 应用	100
6.6	增强对话框函数	101
6.6.1	获得 OnOK 退出控制	101
6.6.2	OnCancel 处理	103

6.6.3 和滚动条控制挂上钩	103
6.7 确认控制: CWnd 指针和控制 ID	105
6.8 设置对话框的背景及其中控制的颜色	106
6.9 在对话框窗口内绘图	106
6.10 利用其它的控制特性	107
第七章 无模式对话框和 COMMDLG 对话框类	108
7.1 无模式对话框	108
7.1.1 创建无模式对话框	108
7.1.2 用户定义的消息	109
7.1.3 对话框的归属	109
7.1.4 无模式对话框的例子——EX07A	109
7.2 CFormView 类——另外一个可供选择的无模式对话框	115
7.3 COMMDLG 对话框	115
7.3.1 直接使用 CFileDialog 类	116
7.3.2 在程序运行的过程中加进对话框控制	116
7.3.3 从 COMMDLG 类进行派生	116
7.3.4 使用 CFileDialog 的一个例子——EX07B	116
第八章 Visual Basic 控制	122
8.1 Windows 的标准控制和普通常规控制	122
8.2 C++ 的类和 Visual Basic 控制	123
8.2.1 方法和成员函数	123
8.2.2 属性和数据成员	123
8.2.3 Visual Basic 控制事件和 Windows 控制通知	124
8.3 Visual Basic 事件登记	124
8.4 CVBControl 类	124
8.5 EX08A 例程序	125
8.5.1 Visual Basic GRID 控制	126
8.5.2 创建 EX08A 例程序	126
8.6 利用其它的 Visual Basic 控制	135
8.7 Visual Basic 的图象属性	135
8.8 编写和利用 Visual Basic 控制的有利和不利因素	135
第九章 全新的 Windows 内存管理	136
9.1 内存模式综述	136
9.2 16 位 Windows	137
9.2.1 Intel 段结构	137
9.2.2 DGROUP 段和多事件程序	138
9.2.3 内存模式——一个代码段还是多个代码段	139
9.2.4 近程函数调用	139
9.2.5 内存模式——一个数据段还是多个	140
9.2.6 局部堆	142
9.2.7 全局堆	142
9.2.8 程序的 vtable 在内存中的位置	144

9.2.9 直接使用 Windows 的内存申请函数	144
9.2.10 80386/80486 的虚内存管理器	144
9.3 检测内存疏漏(Leak)	145
9.4 内存用尽时该如何处理	146
第十章 位图	147
10.1 GDI 位图和与设备无关的位图(DIB)	147
10.2 使用 GDI 位图	147
10.3 彩色位图和单色位图	148
10.4 从资源中装入 GDI 位图	148
10.5 显示映射方式的作用	149
10.6 对位进行伸缩处理	149
10.6.1 EX10A 例程序	150
10.7 利用位图改进屏幕显示	152
10.7.1 EX10B 例程序	153
10.8 GDI 位图的其它应用	157
第十一章 位图按钮、计时器和空状态处理	158
11.1 位图按钮	158
11.1.1 EX11A 例程序	158
11.1.2 对位图按钮的进一步讨论	162
11.2 计时器的使用及控制的产生	163
11.2.1 计时器	163
11.2.2 产生控制	163
11.2.3 EX11B 例程序	164
11.3 空状态处理	167
11.3.1 EX11C 程序	168

第三部分 文档-视图结构

第十二章 菜单和键盘加速键	175
12.1 主框架窗口和文档类	175
12.2 Windows 的菜单	176
12.3 键盘加速键	177
12.4 命令处理	177
12.4.1 派生类中的命令消息控制	178
12.4.2 更新命令用户接口(UI)消息	178
12.4.3 对话框发送的命令	179
12.5 应用框架的内置菜单项	179
12.6 菜单项的允许/禁止	180
12.7 CEditView 类	181
12.8 EX12A 例程序	181
12.9 CMenu 类	186
12.10 扩展命令处理	187
12.11 EX12B 例程序	188

第十三章 工具条和状态条	193
13.1 控制条和应用框架	193
13.2 工具条	193
13.2.1 工具条位图	194
13.2.2 按钮的状态	194
13.2.3 工具条和命令消息	195
13.2.4 工具条更新命令 UI 消息	195
13.3 寻找主框架窗口	196
13.4 EX13A 工具条例程序	197
13.5 状态条	202
13.5.1 状态条的定义	202
13.5.2 信息行	202
13.5.3 状态指示器	203
13.5.4 获得对状态条的控制	203
13.6 EX13B 状态条例程序	204
第十四章 可重用基类	209
14.1 为什么可重用基类难以设计	209
14.2 CPersistentFrame 类	209
14.3 CFrameWnd 类和 ActivateFrame 成员函数	210
14.4 Windows 的 INI 文件	210
14.5 使用 CString 类	212
14.6 极大窗口的位置	214
14.7 静态数据成员	214
14.8 缺省窗口矩形	214
14.9 EX14A 例程序	215
14.10 MDI 应用中的持续框架	220
第十五章 文档与视图的分隔	221
15.1 文档-视图之间的相互作用函数	221
15.1.1 CView 类的 GetDocument 函数	221
15.1.2 CDocument 类的 UpdateAllViews 函数	222
15.1.3 CView 类的 OnUpdate 函数	222
15.1.4 CView 的 OnInitialUpdate 函数	223
15.2 最简单的文档-视图应用	223
15.3 CFormView 类	224
15.4 COject 类	225
15.5 诊断信息转储	225
15.5.1 TRACE 宏	225
15.5.2 afxDump 对象	226
15.5.3 信息转储环境及 COject 类	226
15.5.4 未被删除对象的自动信息转储	227
15.6 EX15A 例程序	229
15.7 更加高级的文档-视图之间的相互作用	235

15.8	CDocument 的 Delete Contents 函数	236
15.9	CObList 集合类	236
15.9.1	对先入先出(FIFO)表使用 CObList 类	237
15.9.2	在 CObList 中进行迭代——POSITION 变量	238
15.9.3	信息转储环境及集合类	239
15.10	EX15B 例程序	240
15.11	CEx15bApp	241
15.12	CMainFrame	242
15.13	CStudentDoc	242
15.13.1	ClassWizard 和 CStudentDoc	245
15.13.2	数据成员	245
15.13.3	构造函数和析构函数	245
15.13.4	GetList	245
15.13.5	DeleteContents	246
15.13.6	Dump	246
15.14	CStudentView	246
15.14.1	ClassWizard 和 CStudentView	253
15.14.2	数据成员	254
15.14.3	OnInitialUpdate	254
15.14.4	OnUpdate	254
15.14.5	工具条按钮命令消息控制函数	254
15.14.6	工具条按钮的更新命令 UI 消息控制函数	254
15.14.7	protected 型虚函数	255
15.15	资源要求	255
15.15.1	符号	255
15.15.2	Edit 菜单	255
15.15.3	IDD_STUDENT 对话框	255
15.15.4	工具条	256
15.15.5	测试 EX15B 应用	256
15.16	留给读者的两个练习	257
第十六章	读和写文档——单文档接口	258
16.1	什么是序列化	258
16.1.1	磁盘文件和归档(Archives)	258
16.1.2	使类可序列化	259
16.1.3	编写序列化函数	259
16.1.4	从归档中装入——内嵌对象和指针	260
16.1.5	使集合序列化	263
16.1.6	序列化函数和应用框架	263
16.2	SDI 应用	264
16.2.1	Windows 应用对象	264
16.2.2	文档模板类	264
16.2.3	文档模板资源	265

16.2.4	SDI 文档的多视	267
16.2.5	创建空文档——CWinApp 的 OnFileNew 函数	267
16.2.6	文档类的 OnNewDocument 函数	267
16.2.7	将 File Open 与序列化代码相连接——OnFileOpen	268
16.2.8	文档类的 DeleteContents 函数	268
16.2.9	将 File Save 和 File Save As 与序列化代码相连接	268
16.2.10	文件的 IsModified 标记	269
16.3	EX16A——序列化 SDI 例程序	269
16.4	CStudent	270
16.5	CEx16aApp	271
16.6	CFrameWnd	275
16.7	CStudentDoc	278
16.7.1	Serialize	278
16.7.2	OnOpenDocument	278
16.7.3	OnUpdateFileSave	278
16.7.4	CStudentView	279
16.7.5	AppWizard 和 EX16A	279
16.7.6	测试 EX16A 应用	279
16.7.7	File Manager 中的文档相关	280
第十七章	对文档的读和写——MDI	282
17.1	MDI 应用	282
17.1.1	一个典型的具有 MFC 库风格的 MDI 应用	283
17.1.2	MDI 应用对象	284
17.1.3	MDI 文档模板类	285
17.1.4	MDI 框架窗口和子窗口	285
17.1.5	主框架和文档模板资源	286
17.1.6	创建空文档——CWinApp::OnFileNew 函数	287
17.1.7	为现存文档创建新的视	287
17.1.8	装入和存入文档	287
17.1.9	多文档模板	288
17.2	拖和放——程序和文档	288
17.2.1	程序登记	288
17.2.2	允许拖和放	289
17.2.3	允许内嵌启动	289
17.2.4	程序启动参数	289
17.2.5	响应 DDE 消息	290
17.3	EX17A 例程序	290
17.4	CEx17aApp	291
17.5	CMainFrame	295
17.5.1	测试 EX17A 应用	297
17.6	EX17B 例程序	298
第十八章	打印和打印预览	301

18.1	Windows 的打印	301
18.1.1	标准打印机对话框	301
18.1.2	交互选择打印页码	301
18.1.3	显示页和打印页	303
18.2	打印预览	303
18.3	有关打印机的程序设计	303
18.3.1	打印机设备环境和 CView::OnDraw 函数	304
18.3.2	CView 类::OnPrint 函数	304
18.3.3	准备设备环境——CView 类::OnPrepareDC 函数	304
18.3.4	打印工作的开始和结束	305
18.4	EX18A 例程序——一个所见即所得的打印程序	305
18.5	EX18B 例程序——多页打印程序	311
18.6	留给读者的一些练习	316
第十九章	切分窗口和多视	317
19.1	切分窗口	317
19.2	多视选择	317
19.3	动态和静态切分窗口	318
19.4	EX19A——具有单个视类的 SDI 动态切分例子	318
19.4.1	CMainFrame	318
19.4.2	资源需求	319
19.4.3	测试 EX19A 应用	320
19.5	EX19B——双视类 SDI 静态切分例子	320
19.5.1	CHexView	320
19.5.2	CMainFrame	321
19.5.3	测试 EX19B 应用	322
19.6	EX19C——多视类 MDI 例子	322
19.6.1	CEx19cApp	322
19.6.2	CMainFrame	323
19.6.3	资源要求	325
19.6.4	测试 EX19C 应用	325
第二十章	上下文相关帮助	326
20.1	Windows 的 WINHELP 程序	326
20.1.1	复文本格式(RTF)	326
20.1.2	编写简单的帮助文件	327
20.2	应用框架与 WINHELP	331
20.2.1	调用 WINHELP	332
20.2.2	使用搜索字符串	332
20.2.3	帮助上下文别名	332
20.2.4	决定帮助上下文	333
20.2.5	进入帮助的菜单	333
20.2.6	F1 帮助	334
20.2.7	Shift-F1 帮助	334