

微软编程技术经典之作



OLE 2

对象嵌入与链接技术

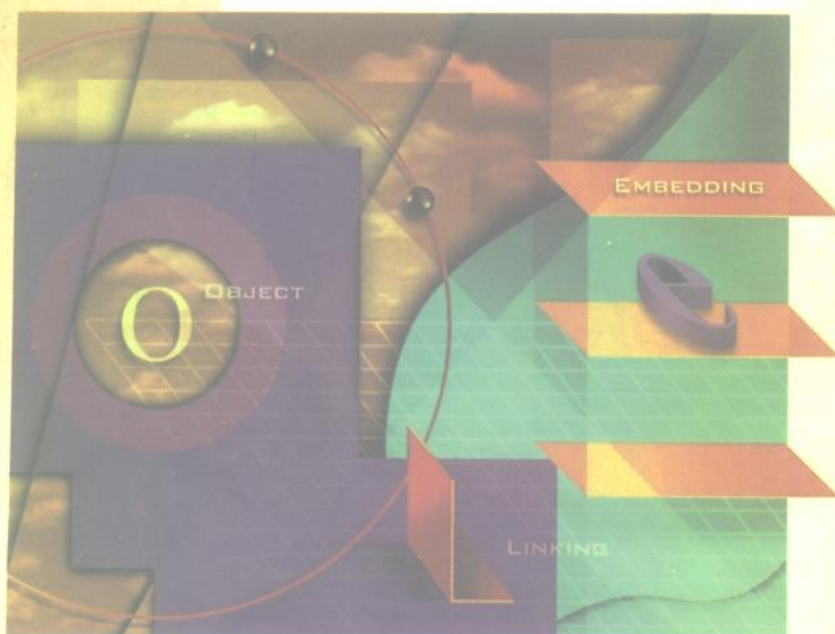
高级编程技术

INSIDE OLE 2

[美] Kraig Brockschmidt 著

张起 郑坚力 等 译 熊桂喜 陈清 审校
丁力 金耀初

- 使用 Windows 对象构造强功能面向对象应用程序的最快速途径
- 随附软盘, 具体展示如何实现对象和在应用程序中集成 OLE 2 特性



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



384953

OLE 2
对象链接与嵌入技术
高级编程技术

[美] Kraig Brockschmidt 著
张 起 郑坚力
丁 力 金耀初 等 译
熊桂喜 陈 清 校

电子工业出版社

内 容 提 要

对象链接和嵌入(简称 OLE)2.0 版,为应用程序间的集成和交互提供了一种前所未有的高级手段。在 Windows 环境下,有了 OLE 2,能更充分地利用各种成熟的软件,生成既有文字,又有图形、电子表格,甚至还有声音、图象的构件文档。操作者不用离开文档,就可充分享受各种软件的各种功能,所有应用程序之间的交互、传输等工作,均由 OLE 自动化自动生成。

本书是微软 OLE 2 系列丛书中的一本,完整地介绍了 OLE 2 的功能以及在应用程序中使用这些功能的方法。全书共分四个部分,分别介绍于 OLE 对象模式的结构、面向对象的系统性能、复合文档和现场激活等较高级的 OLE 2 编程技术。

本书的读者对象为在 Windows 环境下编程的技术人员、研究人员和计算机爱好者。

本书英文版由美国 Microsoft Press 出版,中文版于 1994 年 1 月经 Microsoft Corporation 授权电子工业出版社在中国独家出版。未经出版者书面许可,不得以任何形式或手段复制或抄袭本书内容。

Copyright © 1994 by Microsoft Corporation.

OLE 2 对象链接与嵌入技术 高级编程技术

[美]Kraig Brockschmidt 著

张 起 郑坚力

丁 力 金耀初 等 译

熊桂喜 陈 清 校

特约编辑:沈 翔

责任编辑:王世忠

*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

电子工业出版社计算机排版室排版

北京科技印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:52.5 字数:1274.6 千字

1995 年 10 月第 1 版 1995 年 10 月第 1 次印刷

印数:5000 册 定价 98.00 元(含两张软盘)

ISBN 7-5053-2677-5/TP·831

译者序

本书是美国微软公司有关 OLE 2 技术的一部重要著作，经电子工业出版社引入中国发行其中文版。相信该书在中国的出版能够为广大软件技术人员提供一本内容丰富的参考材料。

本书是集体智慧的结晶，由张起、郑坚力、丁力和金耀初四位同志组织翻译工作。参加本书翻译工作的还有：王忆强、李伟、华国强、王勇、王智利以及董岳、沈飞、黎伟国等同志；杜海燕、常红、马兴和王雪瑞四位小姐为本书译稿的抄写付出了大量的工作，在此表示感谢。

北京航空航天大学的熊桂喜、陈清两位老师，为本书的审校工作付出了艰辛的劳动。没有熊桂喜和陈清老师的帮助，本书的出版工作将会困难重重。

微软公司北京测试中心的韩雪灵小姐在百忙之余参加了本书的审校工作，并提出了许多宝贵意见，在此表示深深的谢意。

由于译者的水平有限，加之 OLE 2 技术的深奥，本书也许还有许多不尽人意之处，欢迎读者批评指正。

译者

1995.8

目录

前言	1
读者对象	2
有关汇编语言的要求	3
书中的代码格式	3
致谢	3
路线图 (Road Map)	4

第一部分 Windows 对象

第一章	OLE 2 概述	9
	Windows 对象：构件对象模式 (Component Object Model)	10
	对象和接口	12
	结构化存储与复合文件	15
	标准数据传输及通告	17
	通知单 (Notification)	18
	数据对象和剪贴板	18
	数据对象与拖放	19
	数据对象和复合文档	19
	数据对象和 DDE	20
	复合文档：对象嵌入	20
	复合文档：对象链接和标记	21
	复合文档：定位激活 (直观编辑)	23
	自动化	23
第二章	约定、C++ 及示例代码	25
	C 还是非 C (拟莎士比亚语)	25
	用户定义型：C++ 类	26
	访问权限 (Access Rights)	30
	单一继承性	31
	虚拟函数与抽象基类	34
	多重继承性	36
	示例代码	38
	包含文件：INC 目录	38

库: LIB 目录	39
BUILD 目录	40
三个 Amigos: BtnCur, GizmoBar 及 StatStrip	40
类库: CLASSLIB 目录	41
接口模板: INTERFAC 目录	43
章节源文件: CHAPxx 目录	44
Cosmo: 一个图形编辑器	44
Patron: 一个页面载体	46

第三章	对象和接口	51
	生命的首要问题, 宇宙和对象 (拟道格拉斯·埃德姆语)	52
	Windows 对象与 C++ 对象的比较	55
	让我们旅行去	56
	Windows 对象与 C++ 对象的其它不同之处	59
	类定义	59
	对象例示	59
	对象引用	60
	对象毁灭	61
	C 和 C++ 中的一个简单的对象: RECTEnumerator	61
	RECTEnumerator 和 IEnumRECT 接口	62
	创建 RECTEnumerator 对象	84
	使用 IEnumRECT 指针	84
	引用计数	85
	我的最优化王国	87
	调用 - 使用 - 释放	88
	IUnknown, 万物之根	89
	QueryInterface 和继承 (Inheritance) 的比较	91
	QueryInterface 特性和接口的生命周期	91
	一些数据类型和调用约定	93
	STDMETHOD 和相关内容	93
	HRESULT 和 SCODE	94
	全局唯一标识符: GUID、IID、CLSID	96
	OLE 2 接口和 API 函数	96
	客户接口 (Custom Interface)	97
	接口与 API 函数的比较	98
	什么是 Windows 对象 (再次讨论)	98
	小结	99

第四章	构件对象（构件对象模型）	101
	令人着迷的东西究竟在哪儿	102
	Windows 对象的新增应用程序	104
	扩大信息队列	104
	确认建库版本	105
	调用 CoInitialize 或 OleInitialize	106
	调用 CoUninitialize 或 OleUninitialize	107
	内存管理和分配程序对象	107
	来自类标识符的构件对象：构件用户	120
	#include<initguid.h>和预编译过的头文件	128
	例示一个构件对象	129
	管理对象及调用 CoFreeUnusedLibraries	132
	实现一个构件对象和一个服务器	133
	注册 CLSID	154
	实现分类工厂（Class Factory）	156
	展示分类工厂	157
	DLL 服务器	157
	EXE 服务器	158
	提供一个卸载机制	160
	DLL 服务器	160
	EXE 服务器	162
	Cosmo 的 Polyline——一个 DLL 对象	164
	对象复用性	168
	示例一：对象包含	169
	示例二：对象聚集（Aggregation）	170
	小结	176

第二部分 面向对象的系统性能—文件和数据传输

第五章	结构化存储和复合文件	179
	动机的形成	181
	使人紧张不安的 Patron 文件	181
	去除咖啡因的方法	183
	使你精力充沛却不会紧张不安：复合文件	185
	复合文件的特征	186
	流、存储及位锁存对象	186
	单元的命名	187
	存取模式	188

交易式存储 (Transacted Storage)	189
增量式存取 (Incremental Access)	190
共享单元	191
复合文件对象和接口	192
存储对象和 IStorage 接口	192
存储的复合文件实现	193
流对象和 IStream 接口	195
流的复合文件实现	197
LockBytes 对象和 ILockbytes 接口	197
LockBytes 的复合文件实现	197
Stat 成员函数及 STATSTG	198
实际使用的复合文件	200
简单存储: Cosmo	200
用 STGM - CONVERT “从帽子上拔去野兔毛”	207
流及文件	208
复杂的复合文件: Patron	210
根存储和临时文件	229
管理子存储	230
多级认可	231
File Save As 操作	233
Low-Memory Save As 操作	234
作为内存结构的流	236
OLE 2 的其它技术与结构化存储	236
IPersistStorage、IPersistStream 和 IPersistFile 接口	237
具有 IPersistStorage 的一种强化协议	239
有关构件用户和 IPersistStorage: 构件 Cosmo	241
有关构件对象与 IPersistStorage: Polyline	244
复合文件的压缩	250
小结	258
第六章 使用数据对象的一致数据传送	261
什么是数据对象	262
新的和改进的超结构	263
数据对象和 IDataObject 接口	267
FORMATETC 枚举器和格式排序	270
构件数据对象	276
某些 CDataObject 功能部件	287
实现 IDataObject	287
一个 (构件) 数据对象用户	288

数据对象的通告和通知单	301
建立咨询连接	307
发送通知单作为一个数据对象	310
对远程通知单的特殊考虑	314
在通告接收器内部	315
IDataObject 作为一个面向对象数据传递的标准	317
视图对象和 IViewObject 接口	319
IViewObject:: Draw	320
对一个特定设备的呈送	322
绘制进入一个元文件 (Metafile)	322
中止长时间的重画	323
其它 IViewObject 成员函数	324
IViewObject 和通知单	324
来自 OLE2.DLL 的自由存取 (Freeload)	325
IDataObject 和 DDE	334
小结	335
第七章	
使用数据对象的剪贴板格式	337
OLE 2 剪贴板协议	338
但我想要做的只是拷贝一些简单的数据!	340
一个数据传送构件对象	342
如果已拥有一个数据对象... Component(构件) Cosmo	353
如果已经有了扩展的剪贴板处理(Clipboard - handling)代码	355
简单的数据源和用户: Cosmo	355
启动/关闭	356
Copy (复制) /Cut (剪切)	357
使能 Edit (编辑) /Paste 粘贴	358
Paste (粘贴)	359
选择性粘贴和一个实用的 Patron	361
Paste Special (选择性粘贴) 对话框和 OLE2UI 库	363
“房客”创建, 粘贴	367
保存及装载“房客”	370
Copy 和 Cut	371
小结	372
第八章	
使用数据对象的拖放操作	373
源和目标: 拖放传送模型	374
一个循序渐进的拖放实现: Cosmo	379
设计并实现放目标用户反馈	394

实现一个放目标对象及 IDropTarget 接口	396
IDropTarget::DragEnter	396
IDropTarget::DragOver	400
IDropTarget::DragLeave	401
IDropTarget::Drop	401
注册并调入放目标对象	403
设计并实现放源用户反馈	403
决定拾起事件 (Pick Event)	404
实现一个放源对象及 IDropSource 接口	404
调用 DoDragDrop	405
高级拖放: Patron 中的反馈与滚动	407
“房客”拾起区域和放源	408
更为高级的放目标击打测试	410
一个反馈矩形	411
滚动页面	414
小结	419

第三部分 复合文档: OLE

第九章 复合文档和嵌入载体	421
复合文档机制	422
被动态	422
装载态	423
情况 1: InProcServer	425
情况 2: InProcHandler	426
情况 3: 缺省处理器 (Default Handler)	426
装载对象: 所有情况	428
绘制对象	429
运行态	431
妈妈, 爸爸, 新的对象是从哪里来的?	432
载体应用程序的结构	434
逐步实现嵌入载体	435
在启动和停机时调用初始化函数	436
定义现场和管理现场存储	436
实现现场接口并加入现场变量	439
实现 IAdviseSink	444
实现 IOleClientSite	447
实现现场的加阴影	453
调入 InSert Object 对话框	455

	调用 OleUIInsertObject	457
	调用 OleCreate 或 OleCreateFromFile	459
	初始化对象	461
	绘制和打印对象	465
	激活对象并加入对象动词菜单	465
	鼠标双击	468
	对象动词菜单	469
	右鼠标按钮弹出式菜单	473
	从剪贴板和拖放传送来创建对象	475
	拷贝和资源 (source) 嵌入式对象	477
	关闭并删除对象	480
	保存和装载包含嵌套对象的文档	481
	处理图标表示形式 (高速缓存控制)	484
	小结	486
第十章	复合文档和嵌入对象服务器 (EXE)	487
	服务器应用程序的结构	487
	链接支持及最小服务器和最大服务器的比较	489
	版本号	489
	安装	489
	逐步实现嵌入服务器	490
	在启动 (Startup) 和停机 (Shutdown) 时调用初始化函数	491
	创建登记数据库 (Registration Database) 条目	491
	实现和注册一个分类工厂	496
	面向嵌套对象的分类工厂	501
	以 - Embedding 起动	502
	实现一个携有 IUnknown 的初始对象	504
	实现 IPersist Storage 接口	512
	实现 IDataObject 接口	518
	实现 IOleObject 接口	527
	平凡函数	530
	必要的函数	540
	可选的函数	543
	但它仍然不能工作	544
	修改服务器的用户接口	545
	发送通知单	549
	(全服务器) 加入 OLE 2 剪贴板格式	552
	(任选的) MDI 服务器用户接口和停机	553
	小结	554

第十一章	IN - PROCESS 对象处理器和服务	555
	在处理模块的结构	555
	为什么要使用处理器	557
	为什么使用在处理服务器	558
	为什么不使用在处理服务器	558
	委托给缺省处理器	559
	IOleObject	560
	IDataObject	561
	IPersistStorage (在高速缓存上)	563
	IViewObject	564
	实现一个对象处理器	565
	获得缺省处理器 IUnknown	567
	在 QueryInterface 中采用缺省处理器接口	570
	实现 IPersistStorage	572
	实现 IOleObject::GetExtent	577
	实现 IViewObject	578
	要与当地服务器齐头并进	583
	年终奖	587
	对实现一个在处理服务器的附注	587
	小结	599
第十二章	标记和链接载体	601
	谁愿意解决一下“什么是一个标记”这个问题?	602
	标记类	603
	从何处获得标记?	606
	逐步链接载体	607
	从 Insert Object (插入对象) 中使链路得以实现	608
	从剪贴板和拖放操作中使链接得以实现	611
	粘贴链路和粘贴特定命令	612
	拖放链接反馈	615
	测试链接	616
	警告	617
	警告: 潜在的难题!	618
	实现 Show Objects (示出对象) 命令	618
	管理一个文件标记, 调用 IOleObject::SetMoniker, 并实现 IOleClient	623
	Links (链路) 对话框和 IOleUILinkContainer 接口	627
	调入 Links 对话框	643

	在装载一个文档时更新链路	646
	小结	649
第十三章	标记联编和链路源	651
	标记联编机制	652
	一个简单的链接对象：单个文件标记	653
	一个携一合成的 File! Item (文件/项目) 标记的链接对象	655
	联编一个合成的	658
	联接上下文	660
	运行对象表	660
	一个简单的链路源：Cosmo	661
	创建、登记及撤消一文件标识	662
	在数据传送时提供链接源格式	664
	实现 IPersistFile 接口	667
	实现 IOleObject::SetMoniker 和 IOleObject::GetMoniker	670
	复杂链接和链接到嵌入	673
	为什么要链接到嵌入?	673
	创建并管理合成标记	675
	源生 (source) 合成标识	679
	以 IPersistFile 为文档对象实现一个分类工厂	681
	为每个项目标记实现 IOleItemContainer	684
	小结	697
第十四章	与 OLE 1 的转换、模拟和兼容	699
	面向载体的 Convert (转换) 对话框	700
	支持一个 Convert 菜单项并调入 Convert 对话框	701
	处理 Convert To 选择	705
	处理 Activate As 选择	706
	处理 Display As Icon 更改	708
	在服务器之间的转换	710
	面向转换的 Registration Database (登记数据库) 条目	711
	AutoConvertTo 和 TreatAs	711
	Conversion (转换)	712
	对 IPersistStorage 的修改	713
	OLE 1 嵌入对象的转换和模拟	715
	对面向载体的 OLE 1 兼容性的注释	720
	OLE 1 服务器的异常表现	720
	文件转换	721
	小结	726

第四部分 复合文档：定位激活

第十五章	可视编辑：定位激活和定位载体	729
	定位对话的动机和内容	730
	从何而始呢？	731
	DoVerb——一个小傻瓜	732
	定位激活	733
	操作一个活动对象	739
	拔出插头：释放	741
	Active 和 UI Active 以及由里及表 (inside-out) 的对象	741
	是的，确实是这样的	742
	逐步实现定位载体	742
	准备载体	743
	实现骨架定位载体接口	745
	经验：访问框架和文档的现场要求	747
	激活和释放对象	753
	经验：服务器仍在运行	755
	混合一个菜单：要摇晃，不要搅拌	755
	经验：菜单销毁：就这样做 (对)	759
	协商工具空间	759
	经验：焦虑和 DeferWindowPos	765
	经验：“驯化” ShowObject	765
	提供定位加速键和焦点	768
	经验：古怪的 TranslateAccelerator	771
	请拐弯：其它杂项	771
	警告	774
	小结	780
第十六章	复合文档对象的定位激活	781
	定位对象的步骤	781
	驱动器，请准备好对象	782
	实现骨架定位对象接口及对象帮助器函数	784
	实现简单的激活和释放	787
	对象装饰	795
	菜单的汇编与反汇编	798
	警告	801
	警告	804
	创建和消除定位工具	804

前言

给我一条鱼，只够我吃一天，
教我如何捕鱼，可使我饱餐终身。

— 谚语

这是一本有关“鱼”的书。但如果不知道如何捕鱼，那最终还将挨饿。因此，它又是一本有关如何“捕鱼”的书。这里所谓的“鱼”，指的是作为一个软件开发者在应用程序中开发 OLE 2 功能所需要的信息。而所谓的“捕鱼术”指的是对为什么要设计你使用的特定信息以及它们的发展方向的描述。自然，即使已储备了足够的食物，也应当坚持不断地去捕鱼。因此，在每一章的开始，我都会力图吸引读者去详细地阅读。

人们常说作者写书并不因为是要让别人读懂它，而是因为作者自己懂得它。对这一点，我深有体会。1992 年中旬我在微软开发联络组（Microsoft's Developer Relations Group）接触 OLE 2 时，我不仅仅把这种技术看成是生成支持所谓“复合文档”的应用软件的方法。就象 OLE 版本 1 一样，这种观点在 Microsoft 已被广泛地接受。因为 OLE 2 是自 OLE 1 改进过来的。但事实上，OLE 2 的设计规范是以复合文档为核心，配以许多其它的技术，从而解决在 OLE 1 中出现的最令人头痛的问题的。

好几个月来，我埋头苦干，通过 OLE 2 的预演信息来生成一些模版应用程序来展示复合文档。在 OLE 2 开发小组的许多成员帮助下，与其他工作人员一起，我在 Microsoft 内外开设了许多班，以帮助其他人学会使用 OLE 2 来创建复合文档应用程序。然而，似乎总有什么东西在暗示我，对于 OLE 2，有许多东西都超出我原先的理解，但似乎又很难打破将 OLE 2 和复合文档等同起来的概念，因为在每一个可找到的文档中都赋予这两者相同的含义。

在 1993 年 1 月的最初几个星期，我开始注意到，在解决 OLE 1 的最关键的问题中，OLE 2 的设计者实际上创建了一个在窗口下进行对象化编程的更庞大的系统。同时我注意到，OLE 2 中有些技术是从真正的复合文档技术中分离出来的。事实上，我也看到设计者如何能只同复合文档打交道，而应用这些技术，这一点并不是我首先发现的。OLE 2 正是以这种方法设计的，但这一设计方法在某些地方却不幸地消失在 OLE 2 设计者的想法与实际 OLE 2 设计规范之间。然而，逐步地，我又重新发现和领略到了隐藏在整個技术框架之下的许多精妙之处。我在 Microsoft 中的职位使我能在较深的层次开发 OLE 2，甚至可以浏览 OLE 2 的源程序，使我可以真正地进入 OLE 2 之中。

1993 年 1 月中旬的某个星期天下午，我在做一些与 OLE 2 完全不相干的事时，猛然领悟到了埃里克·麦非（Eric Maffei）先生（*Microsoft Systems Journal* 的编辑）所描述的“OLE 境界”。在经历了六个月的思维迷雾后，所有包围住 OLE 2 的次要技术轰然倒下，使我清楚地看到可以从小处入手，循序渐进地开发 OLE 2，而讨论整个版本的最初方法就是写一本书。我迅速打开我的袖珍计算机，三个小时内就拟就了一个提纲。读者现在看到的这本书就是紧紧地围绕着这个提纲写出来的。

我写这本书的出发点是将 OLE 2 分解成几个章节，而每一章与前后章节之间都是相

对独立的。当然，OLE 2 不是一种可以编写完整应用程序的技术（因为我们仍采用许多 Windows API 功能），这是集中介绍 OLE 2 的功能，以及在应用程序中使用这些功能的方法。在书中的每个部分仅给出较少的材料，以帮助读者在看完这部分内容后能巩固所理解的东西。我希望这本书能将你带上一条渐进的道路，走在这条路上，在前面所学到的东西能有助于后面阶段的学习。

在每一章之中都体现了类似的想法。我在介绍每一个特定的功能时，将精心安排的细节逐步地给出，每一步都是建立在前一步的基础之上，而不依赖于后一步的内容。这样的过程可使读者能从不断地加入一些小小的代码开始，到编译应用程序，直到真正看到某些东西运行起来。就我个人而言，我发现不断地重复这种过程是非常必要的。事实上，它能使编程变得非常有趣。也就是说，在今天，“痛苦”的职业编程者时代已经被取代了。我进入计算机世界是因为发现一些 BASIC 的代码是如此地激动人心，我希望能通过这本书使这一感受得到再现。

OLE 2 是从我们目前所用的以功能调用为基础的 Windows 操作系统向未来的对象化操作系统进化的第一步。读者在这本书中将要学到的对象模式在很长一段时间内将是 Windows 编程的一部分，我希望它能为你的编程生涯划出一道分水岭。由于 OLE 2 是第一步，在很长的时间内它将是陌生的。但如果某一天你终于需要自己养活自己了，那就时常需要学习如何“捕鱼”了。在想要学习“钓鱼专家”的本领时，这本书可帮助你抓到足够的“鱼”，以使你免遭饥饿。

读者对象

我在前面已经提到，OLE 2 不是一种可以编写完整应用程序的工具。要使用 OLE 2，必须熟悉如何在 Windows 下编写应用程序。在此，我不打算介绍如何使用现有的 Windows API 功能，同样也不介绍有关 Windows 本身的一些复杂细节。本书将集中介绍 OLE 2。

自然，在此我假设读者已经熟悉 Windows 环境下的编程，并且至少具有 Windows API 的编程经验。此外，由于我们在此将讨论对象化编程，故有关 C++ 的知识也是很有帮助的，当然这不是必需的。事实上，C++ 的知识有时反而会对理解 OLE 2 中的对象模式起到阻碍作用。尽管这本书中的例子都用 C++ 编写，但我尽量使它们看起来非常象标准的 C 语言 Windows 程序。第二章从一个 C 程序员的观点出发（这仅是我在开始写这本书时的个人观点），简单地讨论了一下在本书的范例中使用的 C++。

本书不仅仅是为编程者写的。每章的编排尽量使应用程序结构的设计者只需阅读开始的一小部分即可理解 OLE 2 中的构造是如何运行的，而不一定要了解代码的细节。每章前面的 5 到 20 页均讨论总体结构，而把真正的有关编写代码的详细细节安排在该章的后面部分。因此，如果仅需了解 OLE 2 是如何运行的，那只需阅读每章的第一或第二节。

有关汇编语言的要求

不，我们不使用任何汇编语言，但本书假定读者已装入了如下的合适的软件开发环境：

- C++ 编译程序，如 Microsoft C 7.0 版本或 Microsoft Visual C++ 1.0 版本。本书中作为例子的实用文件均是针对 Microsoft 编译程序的，在其它的环境下应作适当的修改。
- Windows 3.1 软件开发工具包 (SDK)。
- 最重要的是，OLE 2 软件开发工具包在试图建立书中的任一例子前，必须确认 OLE 2 目录已加到用户的 PATH, INCLUDE 和 LIB 环境变量中。

第二章将进一步介绍如何为本书中的范例程序创建合适的构造环境。

书中的代码格式

当读者开始读到此书中的一些代码时，可能会对本书的代码格式的出处感到惊奇。

和许多广泛公开的标准不一样，本书的代码格式是我个人在多年的工作中创造出来的，它不仅可使代码易读、易懂，而且可以防止产生错误。例如，当要比较一个变量和一个常数时，我通常将常数放在前面，即写成 `if (OL == m_cRef)` 而不写成 `if (m_cCRef == OL)`。和所有的 C 和 C++ 编程者一样，我也经常会犯错将 `==` 打成 `=`。当少打了一个“`=`”符号时，若将常数放在左面仅引起编译错误，而将变量放在左面，在编译时通过，却会产生令人讨厌的运行时间错误。

其它所有的格式均具有其合理性并在使用中是前后一致的。我经常听到有人讲，他们宁愿在代码格式上前后一致，而不喜欢总是作特别的定义，而本书正符合了这一要求。这里还要指出的是，在我引用例子时，将实际磁盘文件中能看到的有关文件及其功能的冗长说明部分删去了，这仅仅是为了节省篇幅。而在执行实际功能时，每一个代码都将都与磁盘中的文件完全一样。

致谢

读过此书的读者经常问我，编写此书的灵感从何而来。灵感来自于许多地方，现在让我把这些来源一一列出，并向帮助我完成这部书的以下团体和个人表示感谢。

感谢所有在只拥有极少资料的情况下作出了极大贡献的与我一起工作的程序员，没有他们，我无法写成这本书。

感谢在微软中所有花时间定期检查该项工作的开发者，他们是：查理·卡特尔 (Charlie Kindel)，内格尔·汤普森 (Nigel Thompson)，司考特·斯格拉帕 (Scott Skorupa)，萨拉·威廉姆斯 (Sara Williams)、维诺·查理安 (Vinoos Cherian)、克雷格·威登伯格 (Craig Wittenberg)、道格拉斯·荷杰斯 (Douglas Hodges)、埃里克斯·梯尔斯 (Alex Tilles)、马克·白德 (Mark Bader)、丁·麦克克拉里 (Dean McCrory)，尤其是南特·邵因克斯·布朗 (Nat “Zoinks” Brown)。感谢你们对该书提出的有益的见解。