

# COM

## 原理与应用

潘爱民 著

深入剖析 COM

清华大学出版社



# COM 原理与应用

潘爱民 著



2000.6.2

清华大学出版社

(京)新登字 158 号

内 容 简 介

JS117/11

本书不仅介绍了 COM 的基本原理及其扩展知识,还讲述了 MTS 及 COM+ 的一些知识。全书分为三部分,第一部分为 COM 基础,第二部分为 COM 扩展,第三部分为 COM 应用与发展,介绍了组件化程序设计思想以及多层软件结构模型。读者在学习了 COM 的基本原理之后,结合 MTS 和 COM+ 所倡导的一些概念,就可以从更高的层次来理解和使用 COM 及 COM+ 了。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

**图书在版编目(CIP)数据**

COM 原理与应用/潘爱民著. —北京:清华大学出版社,1999.12  
ISBN 7-302-02268-2

I . C… II . 潘… III . 程序设计-方法 IV . TP311.11

中国版本图书馆 CIP 数据核字(1999)第 63317 号

出 版 者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑:刘 彤 汤斌浩

印 刷 者: 国防工业出版社印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印 张: 29.5 字 数: 700 千字

版 次: 1999 年 11 月第 1 版 2000 年 3 月第 2 次印刷

书 号: ISBN 7-302-02268-2/TP·2127

印 数: 5001~9000

定 价: 39.00 元

# 序 言

从酝酿写这本关于 COM 的书到最后完成本书经历了一年半的时间,整个写作过程甚为艰难,多方面的原因使得本书的写作并不顺利。不过,最后总算完成了预期的目标,当然,是否实现了当初的愿望还得由读者来评判。

促使我写作这本书有多方面的原因。首先,我自己学习 COM 的时候并不轻松,因为我找不到合适的资料;后来我就去读 COM 规范,但是 COM 规范言简意赅,说得很不清楚,经常把我弄得稀里糊涂,后来通过一些实践经验,才明白了很多。其次,很多读者和朋友经常向我咨询如何学习 COM,让我推荐一些资料或者参考书,我也就尽我所能,为他们作一些介绍,但很难为他们找到合适的参考书。还有一个更重要的原因是,我看到国内市场上几乎买不到系统介绍 COM 的中文书,连翻译的书都看不到。由此我就萌生了写作本书的念头。

在本书的写作过程中,我参阅了大量的技术资料,但在讲述 COM 各项技术时,却并没有直接引用这些资料中的内容,更不是简单的翻译。我希望能够根据我在学习 COM 过程中的一些体会制定出一个合理的学习顺序,引导读者走上学习 COM 的捷径;我也希望能够把我对 COM 的理解融入书中,使读者更快地领会 COM 的技术精华。全书按三部分来讲述 COM:第一部分为 COM 基础,在写作这部分内容时,我参阅了“Inside COM”(《COM 技术内幕》,已由清华大学出版社引进出版)一书,但重新调整了内容结构,并增加了一些关于 Windows 开发和 Visual C++、MFC 方面的知识;第二部分为 COM 扩展,在讲述这部分内容时,我参阅了“Inside OLE”一书,但因为此书主要是针对 OLE 讲述一些技术专题,所以我删去了一些不必要的内容,并补充了一些最新的资料;第三部分为 COM 应用与发展,这部分内容我主要参阅了 Microsoft 提供的技术资料,由于这些技术发展变化很快,所以我尽可能按照最新的版本讲述。

本书不仅介绍了 COM 的基本原理,也讲述了 MTS 以及 COM+ 的一些知识,所以阅读本书可以了解到与 COM 相关的技术。本书的第三部分还介绍了组件化程序设计思想以及多层软件结构模型,这是目前计算机软件的一个发展方向。COM 作为 Windows 平台上的组件对象模型,为组件化程序设计和多层软件结构提供了基础平台。因此,读者在学习了 COM 的基本原理之后,结合 MTS 和 COM+ 所倡导的一些概念,就可以从更高的角度来理解 COM 和使用 COM 了。

Microsoft 引用术语非常随意,这为本书的写作带来了很大的困难。COM 技术涵盖面非常广,涉及到的术语又非常多,有大量的术语没有标准的中文说法,比如,“marshaling”、

“moniker”、“apartment”等,对于这些术语,我都采用意译的方法,尽可能地把这些词在 COM 范畴中的真实含义表达出来,同时还能符合汉语的习惯。如果本书的这些术语表达欠妥,或者与其他书籍的译法有出入,还请读者见谅。

最后,我要感谢所有直接或间接为本书作出贡献的同事或朋友。首先,我要感谢我的妻子,没有她多年来对我的支持和理解,我很难写成这样一本技术专题书。其次,我要感谢北大计算机研究所和方正技术研究院的许多同事和朋友:吕肖庆和邹维给了我很多帮助,他们的鼓励曾经使我克服了许多困难;杨巍使我转到 COM 的研究上来,并一直支持我从事相关技术的研究;孙杰、吕思伟、顾磷、王婷、赵绍君等几位研究生为我营造了很好的学习和使用 COM 的气氛,与他们的多次讨论使我受益匪浅。

我也要向微软(中国)有限公司的张伟伟先生表示谢意,他为我提供了一些最新的技术资料以及技术信息。最后,我要向清华大学出版社的汤斌浩先生表示感谢,与他的愉快合作使本书得以顺利出版。

潘爱民

1999 年 6 月于北大蔚秀园

# 目 录

---

引言	1
----	---

## 第 1 部分 COM 原理

<b>第 1 章 概述</b>	9
1.1 COM 的起源	9
1.1.1 OLE 的发展历史	10
1.1.2 组件的产生	10
1.1.3 面向对象的组件模型——COM	11
1.2 COM 结构	12
1.2.1 对象与接口	12
1.2.2 客户/服务器模型	13
1.2.3 COM 库	14
1.3 COM 特性	15
1.3.1 语言无关性	15
1.3.2 进程透明特性	16
1.3.3 可重用性	17
1.4 COM 的发展前景	17
1.4.1 COM 与 Windows 操作系统	17
1.4.2 COM 与数据库	18
1.4.3 COM 与 Intranet 应用	19
1.4.4 COM 与 Internet	19
<b>第 2 章 COM 对象和接口</b>	21
2.1 COM 对象	21
2.1.1 COM 对象的标识——CLSID	22
2.1.2 COM 对象与 C++ 对象的比较	23
2.2 COM 接口	25
2.2.1 从 API 到 COM 接口	25
2.2.2 接口定义和标识	26
2.2.3 用 C++ 语言定义接口	28

2.2.4	接口描述语言 IDL .....	29
2.2.5	接口的内存模型 .....	30
2.2.6	接口的一些特点 .....	32
2.3	IUnknown 接口 .....	33
2.3.1	引用计数 .....	34
2.3.2	实现引用计数 .....	35
2.3.3	使用引用计数规则 .....	39
2.3.4	接口查询 .....	41
2.3.5	COM 对象的接口原则 .....	42
2.3.6	QueryInterface 成员函数的实现 .....	43
2.3.7	COM 对象和接口图示法 .....	45
2.4	字典组件例子程序 .....	45
2.4.1	组件程序 DictComp .....	46
2.4.2	客户程序 DictCtrl .....	51
<b>第 3 章</b>	<b>COM 的实现</b> .....	<b>55</b>
3.1	进程内组件和进程外组件 .....	55
3.1.1	进程内组件 .....	56
3.1.2	进程外组件 .....	58
3.2	通过注册表管理 COM 对象 .....	60
3.2.1	注册表结构 .....	60
3.2.2	COM 组件注册信息 .....	61
3.2.3	COM 组件的注册操作 .....	63
3.3	类厂(class factory) .....	65
3.3.1	类厂和 DllGetObjectClass 函数 .....	65
3.3.2	COM 库与类厂的交互 .....	66
3.3.3	类厂的实现 .....	69
3.3.4	类厂对组件生存期的控制 .....	73
3.4	COM 库 .....	74
3.4.1	COM 库的初始化 .....	74
3.4.2	COM 库的内存管理 .....	75
3.4.3	组件程序的装载和卸载 .....	77
3.4.4	常用函数和 HRESULT .....	79
3.5	COM 实现过程 .....	83
3.5.1	进程内组件与客户的协作过程 .....	83
3.5.2	进程外组件与客户的协作过程 .....	85
3.6	字典组件例子程序的实现 .....	87
3.6.1	字典组件类厂的实现 .....	87
3.6.2	字典对象的实现 .....	87

---

3.6.3	全局引出函数的实现	88
3.6.4	客户程序 DictCtrl	89
<b>第 4 章</b>	<b>COM 特性</b>	<b>92</b>
4.1	可重用性:包容和聚合	92
4.1.1	包容和聚合	93
4.1.2	包容实现	95
4.1.3	聚合实现	99
4.1.4	聚合例子程序	108
4.2	进程透明性	110
4.2.1	列集(marshaling)	111
4.2.2	自定义列集	112
4.2.3	标准列集	116
4.2.4	标准列集的实现	121
4.2.5	进程外组件例子程序	125
4.3	安全性	126
4.3.1	Windows NT 安全机制	127
4.3.2	激活安全性	128
4.3.3	调用安全性	131
4.4	多线程特性	132
4.4.1	Win32 线程和 COM 线程	132
4.4.2	列集和同步	134
4.4.3	套间线程	135
4.4.4	自由线程	136
4.4.5	进程内组件的线程模型	137
<b>第 5 章</b>	<b>用 Visual C++ 开发 COM 应用</b>	<b>139</b>
5.1	Win32 SDK 和 MFC 介绍	139
5.1.1	Win32 SDK	140
5.1.2	MFC 基础:应用类和窗口类	144
5.1.3	MFC 应用结构	146
5.1.4	MFC 库结构	148
5.2	MFC 对 COM 应用的支持	149
5.2.1	用嵌套类实现 COM 接口	150
5.2.2	接口映射表	153
5.2.3	CCmdTarget 类实现 IUnknown	157
5.2.4	COM 引出函数和类厂实现	159
5.2.5	MFC 对 COM 支持小结	161
5.3	用 Visual C++ 开发 COM 应用	161
5.3.1	利用 AppWizard 创建 COM 程序工程框架	162



5.3.2	利用 ClassWizard 添加 COM 对象类 .....	164
5.3.3	ATL 介绍 .....	166
5.4	Visual C++ 与 ActiveX 应用 .....	170
5.4.1	MFC 对自动化的支持 .....	170
5.4.2	MFC 对 ActiveX 控制的支持 .....	171
5.4.3	MFC 对复合文档的支持 .....	172

## 第 2 部分 COM 扩展

<b>第 6 章</b>	<b>可连接对象 .....</b>	<b>175</b>
6.1	可连接对象结构模型 .....	176
6.1.1	客户与可连接对象的关系 .....	176
6.1.2	可连接对象的基本结构 .....	177
6.1.3	客户方基本结构 .....	178
6.2	实现可连接对象(源对象) .....	178
6.2.1	枚举器 .....	179
6.2.2	源对象和 IConnectionPointContainer 接口 .....	180
6.2.3	连接点和 IConnectionPoint 接口 .....	180
6.2.4	建立连接过程 .....	182
6.3	客户-源对象-接收器的协作过程 .....	182
6.3.1	接收器的实现 .....	183
6.3.2	事件的激发和处理 .....	184
6.3.3	与出接口有关的类型信息 .....	185
6.4	可连接对象的程序实现 .....	186
6.4.1	用 IDispatch 接口作为出接口 .....	187
6.4.2	MFC 对连接和事件的支持 .....	189
6.4.3	用 MFC 实现源对象 .....	192
6.4.4	用 MFC 在客户程序中实现接收器 .....	195
<b>第 7 章</b>	<b>结构化存储 .....</b>	<b>202</b>
7.1	结构化存储基础 .....	202
7.1.1	从文件系统进展到结构化存储 .....	203
7.1.2	存储对象和流对象 .....	205
7.1.3	用结构化存储设计应用 .....	210
7.2	结构化存储特性 .....	212
7.2.1	访问模式 .....	212
7.2.2	事务机制 .....	214
7.2.3	命名规则 .....	216
7.2.4	增量访问 .....	216
7.3	结构化存储实现:复合文档 .....	217

7.3.1	LockBytes 对象和 ILockBytes 接口	218
7.3.2	复合文档 API 函数介绍	219
7.3.3	零内存保存特性和 IRootStorage 接口	222
7.3.4	存储对象、流对象和文件的 CLSID 信息	222
7.3.5	复合文档与结构化存储	224
7.4	永久对象实现	224
7.4.1	永久接口	225
7.4.2	永久对象的存储特性	226
7.4.3	永久对象例子程序	228
7.4.4	复合文档例子程序	232
7.5	本章小结	238
<b>第 8 章</b>	<b>命名和绑定技术</b>	<b>239</b>
8.1	名字技术基础	239
8.1.1	COM 名字对象	240
8.1.2	使用名字对象:绑定过程	240
8.2	IMoniker 接口	242
8.2.1	名字管理	243
8.2.2	利用 IMoniker 实现绑定	244
8.2.3	管理复合名字对象	245
8.2.4	名字解析	246
8.3	复合名字对象	248
8.3.1	复合名字对象结构	248
8.3.2	ROT 和绑定环境对象	249
8.3.3	复合名字对象绑定过程的剖析	253
8.3.4	IoleItemContainer 接口	255
8.4	COM 名字对象分类	257
8.4.1	COM 提供的系统名字对象	257
8.4.2	URL 名字对象	263
8.4.3	自定义名字对象	266
8.5	COM 名字对象的应用	267
8.5.1	服务程序中名字对象的应用	268
8.5.2	客户程序中名字对象的应用	269
<b>第 9 章</b>	<b>统一数据传输</b>	<b>272</b>
9.1	统一数据传输基础	272
9.1.1	数据交换与传输协议的分离	272
9.1.2	应用:剪贴板和拖-放	274
9.2	数据传输机制	275
9.2.1	数据结构 FORMATETC 和 STGMEDIUM	275

9.2.2	数据对象和 IDataObject 接口 .....	279
9.2.3	通报连接机制 .....	282
9.2.4	有关数据对象的小结 .....	285
9.3	通过剪贴板传输数据 .....	287
9.3.1	剪贴板协议机理和用法 .....	287
9.3.2	MFC 对剪贴板的支持 .....	290
9.3.3	支持剪贴板操作的例子程序 .....	293
9.4	拖-放数据传输协议 .....	295
9.4.1	OLE 拖-放传输协议 .....	296
9.4.2	MFC 对拖-放特性的支持 .....	299
9.4.3	支持拖-放特性的例子程序 .....	300
<b>第 10 章</b>	<b>分布式 COM(DCOM)</b> .....	<b>302</b>
10.1	DCOM 基本结构 .....	302
10.1.1	从 COM 转向 DCOM .....	303
10.1.2	DCOM 对象的定位 .....	304
10.1.3	列集(marshaling) .....	305
10.1.4	对象 RPC .....	306
10.1.5	DCOM 特性 .....	306
10.2	对象激活 .....	308
10.2.1	创建 DCOM 组件(一) .....	308
10.2.2	创建 DCOM 组件(二) .....	310
10.2.3	远程创建进程内组件;代理进程(surrogate) .....	313
10.2.4	如何连接到指定的远程对象实例 .....	314
10.3	连接管理 .....	316
10.3.1	更好地控制远程对象的生存期 .....	316
10.3.2	pinging 机制 .....	317
10.3.3	连接点管理 .....	317
10.3.4	连接传递 .....	318
10.4	并发管理 .....	318
10.4.1	线程模型 .....	319
10.4.2	消息过滤器 .....	321
10.5	DCOM 安全模型 .....	325
10.5.1	安全性策略 .....	325
10.5.2	安全性配置 .....	327
10.6	本章小结 .....	330

## 第 3 部分 COM 应用与发展

<b>第 11 章 自动化(Automation)对象</b> .....	333
11.1 自动化对象基础 .....	333
11.1.1 自动化产生与发展 .....	334
11.1.2 属性和方法 .....	334
11.1.3 类型库和 ODL .....	335
11.1.4 IDispatch 接口 .....	337
11.1.5 自动化兼容的数据类型 .....	339
11.1.6 参数顺序、可选参数和命名参数 .....	342
11.1.7 IDispatchEx 接口 .....	344
11.2 自动化对象实现 .....	345
11.2.1 类型库支持 .....	346
11.2.2 Invoke 函数实现 .....	350
11.2.3 异常处理 .....	353
11.2.4 多语种——本地化 .....	355
11.2.5 用 CreateStdDispatch 函数实现自动化对象 .....	355
11.3 自动化对象应用 .....	357
11.3.1 双接口 .....	357
11.3.2 迟绑定和早绑定 .....	359
11.3.3 自动化集合对象 .....	360
11.3.4 以 IDispatch 作为出接口 .....	361
11.3.5 自动化控制器 .....	362
11.4 自动化对象编程 .....	364
11.4.1 MFC 对自动化对象的支持 .....	364
11.4.2 COleDispatchDriver 类 .....	367
11.4.3 在 Visual Basic 中使用自动化对象 .....	368
11.5 例子程序 .....	370
11.5.1 例子程序 1——具有计算器功能的自动化对象 .....	370
11.5.2 例子程序 2——自动化客户程序 .....	371
11.5.3 例子程序 3——在 Excel 中使用计算器对象 .....	373
<b>第 12 章 ActiveX 控制</b> .....	376
12.1 ActiveX 控制基础 .....	377
12.1.1 实地激活 .....	377
12.1.2 属性页 .....	380
12.1.3 属性变化通知 .....	384
12.2 ActiveX 控制结构 .....	385
12.2.1 ActiveX 控制基本结构 .....	385

12.2.2	容器基本结构 .....	387
12.2.3	IObject 和 IClientSite 接口 .....	389
12.2.4	IControl 和 IControlSite 接口 .....	391
12.2.5	用于 ActiveX 控制的标准分发 ID .....	394
12.2.6	ActiveX 控制的其他一些特点 .....	397
12.3	ActiveX 控制与 Internet .....	399
12.3.1	从桌面环境转向 Internet .....	399
12.3.2	包装 ActiveX 控制 .....	402
12.3.3	许可证管理 .....	405
12.3.4	Web 页面中 ActiveX 控制的初始化 .....	407
12.3.5	脚本支持与初始化安全性 .....	408
12.4	ActiveX 控制开发和应用 .....	409
12.4.1	用 MFC 实现 ActiveX 控制——Smile 例子程序 .....	410
12.4.2	用 MFC 实现 ActiveX 控制容器——Smiley 例子程序 .....	412
<b>第 13 章</b>	<b>基于 COM 的程序设计 .....</b>	<b>415</b>
13.1	组件化程序设计 .....	415
13.1.1	面向对象程序设计方法 .....	415
13.1.2	组件的概念和特点 .....	417
13.1.3	基于 COM 的组件化程序设计方法 .....	418
13.2	MTS 介绍 .....	420
13.2.1	MTS 基本结构 .....	420
13.2.2	MTS 程序设计概念 .....	422
13.2.3	MTS 管理程序 .....	424
13.2.4	MTS 特性 .....	426
13.3	多层软件结构 .....	427
13.3.1	应用结构的发展 .....	427
13.3.2	多层应用软件结构 .....	429
13.3.3	多层结构的优点 .....	432
13.4	用 COM 设计 Web 应用 .....	433
13.4.1	Web 应用基本结构 .....	433
13.4.2	ASP 与 COM 组件 .....	434
13.5	本章小结 .....	437
<b>第 14 章</b>	<b>COM 发展——COM+ .....</b>	<b>439</b>
14.1	COM+ 基本介绍 .....	439
14.1.1	Windows DNA 策略 .....	440
14.1.2	COM+ 基本结构 .....	441
14.1.3	对象环境 .....	443
14.2	COM+ 系统服务介绍 .....	444

---

14.2.1	COM + 队列组件 .....	445
14.2.2	COM + 事件模型 .....	446
14.2.3	负载均衡 .....	447
14.2.4	内存数据库(IMDB).....	448
14.2.5	对其他服务的增强 .....	449
14.3	COM + 应用开发 .....	451
14.3.1	应用开发支持 .....	451
14.3.2	基于属性的 C + + 编程语言 .....	452
14.4	本章小结 .....	453
<b>附录：部分术语英汉对照表 .....</b>		<b>455</b>

# 引 言

在 20 世纪 90 年代末的今天,面向对象已经不再是新鲜的说法,从早期的面向对象语言到面向对象设计方法,面向对象的思想已经渗入计算机软件科学各个领域。随着软件科学的不断发展,新的应用系统越来越复杂,尤其是这几年 Intranet/Internet 的飞速发展,使软件应用置身于更加广阔的环境中,从而对应用软件提出了更高的要求,这就使得软件设计更加困难。在这样的情况下,面向对象的思想已经难以适应这种分布式软件模型,于是组件化程序设计思想得到了迅速的发展。

按照组件化程序设计的思想,复杂的应用程序被设计成一些小的、功能单一的组件模块,这些组件模块可以运行在同一台机器上,也可以运行在不同的机器上,甚至可以运行在跨越太平洋的两台机器上。在理想的情况下,每台机器的运行环境可以不同,甚至可以是不同的操作系统。为了实现这样的应用软件,组件程序和组件程序之间需要一些极为细致的规范,只有组件程序遵守了这些共同的规范,软件系统才能正常运行。为此,OMG (Object Management Group,对象管理组织)和 Microsoft 分别提出了 CORBA(Common Object Request Broker Architecture,公共对象请求中介体系结构)和 COM(Component Object Model,组件对象模型)标准,目前 CORBA 模型主要应用于 UNIX 操作系统平台上,而 COM 则主要应用于 Microsoft Windows 操作系统平台上。

Microsoft 提出的组件对象模型标准不仅仅体现了组件化的程序设计思想,在实现上,它也采用了面向对象程序设计方法。本书将详细讨论 COM 的原理,同时也介绍一些建立在 COM 基础上的实用技术。阅读完本书后,相信您能够全面了解 COM 原理,而且可以在实际工作中利用 Microsoft 提供的工具进行 COM 应用的开发。

## COM 是什么

COM 是由 Microsoft 提出的组件标准,它不仅定义了组件程序之间进行交互的标准,并且也提供了组件程序运行所需的环境。在 COM 标准中,一个组件程序也被称为一个模块,它可以是一个动态链接库(缩写为 DLL),被称作进程内组件(in-process component);也可以是一个可执行程序(即 EXE 程序),被称作进程外组件(out-of-process component)。一个组件程序可以包含一个或多个组件对象,因为 COM 是以对象为基本单元的模式,所以在程序与程序之间进行通信时,通信的双方应该是组件对象,也叫做 COM 对象,而组件程序(或称做 COM 程序)是提供 COM 对象的代码载体。

COM 对象不同于一般面向对象语言(如 C++ 语言)中的对象概念,COM 对象是建立在二进制可执行代码级的基础上,而 C++ 等语言中的对象是建立在源代码级基础上的,因此 COM 对象是语言无关的。这一特性使用不同编程语言开发的组件对象进行交互成为可能。

Microsoft 不仅仅定义了 COM 规范标准,而且也提供了具体的实现,在 Microsoft Windows 系统平台上,COM 技术被应用于系统的各个层次,从底层的 COM 对象管理到上层的应用程序交互都用到了 COM 标准。各种层次上的 COM 技术应用为我们提供了很好的范例,我们几乎总是可以在 Microsoft 的软件产品中找到 COM 技术的应用。

## 读者对象

在写作本书之前很长一段时间,我一直在考虑:“本书适合于哪些读者?”从目前图书市场上已经出版的一些与 COM 技术有关的书籍来看,几乎看不到介绍 COM 原理的书,大多是介绍用一些工具来实现某项 COM 技术的书,例如,用 Visual Basic 编写 ActiveX 控制。这些书的实用性很强,但都简略了对 COM 原理的介绍,当读者在工作过程中碰到一些深层的错误或者开发工具支持不够时,往往找不到可以参考的资料。COM 原理并不复杂,但人们往往觉得 COM 非常深奥,我想其原因也许在于人们对 COM 的原理了解太少。

如果读者希望在工作中应用 COM 技术,包括一些建立在 COM 基础上的技术,则通过阅读本书了解 COM 的原理和 COM 的应用之后,一定会有所获益,在工作中更好地发挥 COM 的优势,同时为解决一些复杂的 COM 技术问题储备必要的知识。

对于学生和计算机软件科研人员来说,COM 带给他们的不仅仅是一项技术,更是一种软件结构的实现。COM 为他们提供了程序互连的基础,程序之间的通信有了统一的标准,而且这种通信可以从桌面环境扩展到网络环境,甚至是异构网络。对于他们来说,COM 原理是开展工作的必备知识,本书将为他们提供所需要的一般性知识。对于深入的知识,可以从列出的其他参考信息中获得。

在阅读本书之前,建议读者先有 C++ 语言基础。虽然 COM 技术并不要求一定用面向对象语言来实现,但由于 COM 本身包含了面向对象的思想,而且 COM 规范定义中的一些语法与 C++ 语言很类似,因此,熟悉 C++ 语言的读者可以学得更轻松,理解得更好。

如果读者已经对 COM 原理比较了解了,这本书仍然有益。首先,本书在介绍了原理之后,还介绍了一些建立在 COM 之上的,但比较深奥、又比较广泛的底层技术,比如通过名字对象(moniker)的学习,读者可以理解在 Windows 平台上文件连接的实现机制;通过可连接对象(connectable object)的学习,读者可以理解 COM 对象之间事件的机制,等等。最后,本书还对一些实用的技术进行了深入浅出的讲解,并展望了 COM+ 的新特性。

## 本书内容组织

在本书的内容组织上,作者进行了精心安排,正如目录所示,共分三大部分。



## 第 1 部分：COM 基础

这部分主要讲述 COM 的原理,并通过一些小程序来说明 COM 的许多细节问题。第 1 章讲述了 COM 的发展过程以及 COM 的结构和特点;第 2 章给出了 COM 规范的核心内容,细致描述了 COM 对象和 COM 接口的细节;第 3 章讲述了 COM 实现的过程,通过对 COM 对象通信过程的描述,帮助读者理解 COM 在应用过程中的各种行为。第 4 章讲述的是 COM 模型的特性,以及使用 COM 模型构造应用系统为我们带来的好处。最后,第 5 章给出了使用 Visual C++ 开发工具的 MFC 库(Microsoft Foundation Class Library, Microsoft 基本类库)实现 COM 组件开发的一种实现方案,本书后面两部分将利用 MFC 进行示例说明。

## 第 2 部分：COM 扩展

这部分讲述了 COM 规范中扩展的一些内容。第 6 章介绍了 COM 的可连接对象通信机制,这是 COM 组件和客户程序之间进行双向通信的基础;第 7 章介绍了结构化存储(structured storage)技术,因为结构化存储是 ActiveX 技术的底层基础,所以通过这部分内容的学习,读者可以理解 ActiveX 控制和复合文档等技术的数据存储机制;第 8 章介绍了 COM 对象的命名技术,即名字对象技术,揭示了这种技术的一些细节和用法;第 9 章介绍了统一数据传输(uniform data transfer)机制,在 Windows 平台上,统一数据传输机制是剪贴板操作、对象拖-放的基础。第 10 章讨论了 DCOM(Distributed Component Object Model, 分布式组件对象模型)的一些特殊性,虽然 COM 对象具有位置透明特性,但本地 COM 对象和远程 COM 对象在技术上有不同的要求,这一章重点在于 DCOM 的技术细节以及实现上的一些支持。

这部分所有的内容都建立在 COM 原理的基础上,这些技术为 COM 得以广泛应用提供了重要的支持。如果读者要全面理解 COM 标准,则这部分内容可为读者提供所有技术细节。

## 第 3 部分：COM 应用与发展

第 11、12 章从 COM 应用的角度出发,针对目前常用的自动化(Automation)和 ActiveX 控制技术,利用前两部分介绍的原理和扩展规范,对它们作了细致的分析,然后简单介绍了 MFC 的支持,并提供了一些示例子程序来说明其用法。第 13 章介绍了 COM 在组件化程序设计领域中的重要地位,并通过对 MTS(Microsoft Transaction Server, Microsoft 事务服务系统)和多层软件结构的介绍,说明了 COM 在更高层次上的应用,而不仅仅是限于应用的局部实现。第 13 章最后还讨论了用 COM 实现 Web 应用的结构模型。第 14 章介绍了 COM+ 的基本结构以及 COM+ 的一些系统服务。

这一部分在介绍 COM 应用的同时,也介绍了一些最新发展起来的技术,说明了 COM 规范具有广泛的可扩展性,为 COM 以及 COM+ 的进一步发展提供了广阔的空间。

以上三部分内容从不同的角度讲述了 COM 标准,形成了 COM 的全貌,三部分内容既形成了统一的整体,同时每一部分又有自己的主题,保持一定的独立性,因此读者可以根