
HZ BOOKS
COM+ Unleashed
软件开发技术丛书
COM+
技术大全
(美) Richard C. Leinecker 著
高智勇 赵崑 唐华平 黄蔚玲 等译
附赠
CD-ROM
机械工业出版社
China Machine Press
SAMS

软件开发技术丛书

COM+技术大全

（美） Richard C.Leinecker 著

高智勇 赵 崑 唐华平 黄蔚玲 等译

机械工业出版社
China Machine Press

本书介绍COM+技术的主要特性和编程技巧。主要内容包括Windows DNA和COM+的基本概念、高级COM编程技术、组件的管理、事务以及异步组件编程等。本书揭示了COM+的内幕，实例丰富，分析透彻。拥有本书，可以最大限度地发挥COM+的潜力，获得更好的编程技能。本书适合于所有Windows程序开发人员，尤其对具有编程经验的人更加具有参考价值。随书附带的光盘包含了书中所有实例代码，以及微软最新的极有价值的信息。

本书版权登记号：图字：01-2000-2813

图书在版编目（CIP）数据

COM+技术大全 /（美）莱耐克（Leinecker, R.C.）著；高智勇等译. -北京：机械工业出版社，2001.8

（软件开发技术丛书）

书名原文：COM+ Unleashed

ISBN 7-111-08951-0

Ⅰ.C… Ⅱ.①莱…②高… Ⅲ.软件接口，COM+-程序设计 Ⅳ.TP311.52

中国版本图书馆CIP数据核字（2001）第038022号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：宋燕红 张鸿斌

北京昌平第二印刷厂印刷 · 新华书店北京发行所发行

2001年8月第1版第1次印刷

787mm × 1092mm 1/16 · 35.5印张

印数：0 001-5 000册

定价：76.00元（附光盘）

前　言

本书的基本内容包括：

1) 讲述COM+在实际应用和高级解决方案中的强大能力。

2) 提供读者可能永远也想不到的，但是用COM+却可以轻松解决的捷径。

3) 揭开Windows DNA的神秘面纱，讲述其强大的服务器方产品，例如COM+目录的主框架。

4) 指出在实际的COM+/DNA工程中存在的缺陷和细微的差别。

本书面向的对象

本书决不是COM+的参考书或者其理论的概述。它面向于希望进一步深化的中高级COM+程序员。我们假定读者具有C++的使用经验，并且熟悉Windows环境。

本书所包含的内容

COM+是一个很大的话题，完整地讲述它需要好几本书。本书将重点放在使用Microsoft DNA技术编写高级COM+程序上。由于COM+的重要性和趋同性，我们花了很多的篇幅讨论类似于MTS和MSMQ的服务器方DNA产品。尽管DNA的焦点集中于发布的n层互联网程序上，但是我们讨论的编程技术却应用于COM+的编程和普通的互操作上。由于太复杂或者没有详尽的说明文档，COM+许多强大的特性往往被初级程序员所忽视。本书以尽量简单的捷径讲解更多深奥而又强大的、多方面的COM+知识。尽管仅从表面上理解还不够，但是一些特性，例如标记、自定义调度和永久存储，对于准备开发可升级的、可靠而又有效的应用程序的COM+开发者来说却是必须的。尽管我无法从理论的角度完全覆盖COM+的所有内容，但是我也没有完全分割各个主题之间的关系。在COM+中有一些诱人的算法和协议，我将在整本书贯穿讲述，以帮助读者理解COM+技术的优点和缺陷。

本书不包含的内容

本书假定你对COM+已经有所理解并且能用它进行一些基本的工作。我们将不解释最基本的组成构件。例如，你应该熟悉IUnknown接口并且明白它在COM+中所起的作用。本书不比较COM+与其他组件技术的异同，例如CORBA。

IIS是DNA中的一部分，本书不讨论它是因为它处于Web技术的边缘，而COM+通常可以通过脚本和其他高级语言编程。如果你想学习Web应用程序，可以查看本书第1和第2章中提到的参考书目。

与作者联系

我将大部分个人时间花费在开发Web站点和新闻组上以及编写本书的主要动机之一，是以帮

助我的同事开发程序为乐趣。非常感谢你购买了本书，当你对本书有什么疑问或者在开发DNA/Web的过程中遇到什么困难的话，不要犹豫，赶快与我联系。我的Web站点是www.sourceDNA.com。

下载例子

本书中的所有例子都包含在附带的光盘中，安装该光盘后将可以很容易地找到需要的东西。

由于某些原因，尽管许多开发者能够使其上司知道在程序中存在bug是一件很普遍的事情，但是在本书所包含的软件中存在错误却总是会令这些程序员恼火。编辑花了很多时间来测试本书中的例子，但是很遗憾，像其他任何软件一样，在我们的代码中肯定也会有错误。我将及时在我的ftp站点ftp://mcp.com/product_support或http://www.sourceDNA.com修改代码，以提供最新的版本。

软件要求

下面的几个部分讨论了COM+技术内幕的软件需求。本书用到了几个不同的技术和编程工具。

COM

COM并没有指定任何开发语言或工具，它只需要可用的语言或工具以能够产生二进制编码的可注册的组件。尽管可以使用一些非常低级的语言来生成COM组件，例如用汇编语言，但是这往往需要付出特别多的劳动和努力才能在开发组中实现。微软开发了许多工具以尽可能地帮助程序员在程序设计和逻辑推理方面避免麻烦。

在本书中，我使用了微软开发的能够尽快帮助你完成工作并且尽量避免bug的软件。特别地，本书中的代码使用了Visual C++ 6.0和Service Pack 3及Active Template Library来编写，其中一些例子使用了Visual Basic 6.0和VB脚本来实现，以示范COM的多种解决方案。COM库使用了标准Windows NT、95/98和2000的库。

COM+

尽管Windows NT、95/98支持COM，但是只有Windows 2000支持COM+。为了开发、测试和使用本书中的例子，你的计算机上应该装有Windows 2000。

MTS/MSMQ

MTS和MSMQ可以从微软的站点上免费得到，它们是Option Pack 4.0的一部分。MSMQ至少需要NT Server 4.0以作为基础企业控制器，MTS则可以运行在Windows NT、95/98上。

本书的结构

本书包含四部分，后一部分建立在前一部分的基础之上。我尽量使每一部分包含自己的风格，但是这不会影响你的阅读。这四部分是：

Windows DNA 和COM+

本部分讲述了组件对象模型以及它与微软DNA结构的关系。主要的服务器方产品和开发工具包含在DNA环境中。在本部分你还可以学习多层建筑的声音设计概念（基于组件的编程思想）。

高级COM+编程技巧

假定读者已经是中高级的COM+程序员，在本部分中讲述了功能更加强大的、但是有时隐含的COM/COM+特性。比如组件的连续性、事件、标记、线程和其他高级问题。

组件管理和事务

本部分讨论了Microsoft事务服务和COM+服务。这些丰富的资源和组件管理服务比单纯的事务要有用多。

异步组件编程

在当今移动和非连接的计算机世界，MSMQ和COM+队列组件为之提供了异步通信。这些技术为从应用到应用的通信提供了可容错和可升级的解决方案。

本书由高智勇、赵崑、唐华平、黄蔚玲、王绣勤等翻译，高智勇作了最后的审稿，参加本书翻译工作的还有白桐、谭心、李明、郭东、刘波、刘彩伟、邓建国、徐丽、唐林平等。另外，姜春林、周余平对本书的翻译提出了不少好的建议。

目　录

前言

第一部分　Windows DNA和COM+

第1章　COM+：Windows DNA的粘合剂 ……1
1.1　Windows DNA ……1
1.2　谈谈因特网：HTML和XML ……2
1.3　Windows DNA服务 ……3
1.3.1　Windows DNA服务：COM和COM+ …3
1.3.2　Windows DNA服务：DNA内核 ……6
1.3.3　Windows DNA服务：工具 ……7
1.4　DNA：功能概述 ……10
1.4.1　浏览器 ……10
1.4.2　IIS ……10
1.4.3　ASP ……11
1.4.4　MTS ……12
1.4.5　MSMQ和SQL Server ……12
1.4.6　Visual Studio 6.0 ……12
1.4.7　Visual Basic ……14
1.4.8　Visual C++ ……14
1.5　小结 ……15
第2章　多层组件结构 ……16
2.1　Ad-Hoc设计 ……16
2.2　基础应用程序边界：外观、逻辑、数据服务 ……17
2.3　三层设计 ……18
2.4　保持层与层之间的均衡 ……20
2.5　多层设计 ……20
2.6　本地或分布 ……22
2.7　几种不错的设计技巧 ……22
2.7.1　将应用抽象为各层 ……23
2.7.2　确定组件 ……23
2.7.3　创建接口 ……24
2.7.4　实现组件 ……24
2.7.5　设计约束 ……24
2.7.6　设计目标 ……26
2.8　设计工具 ……28
2.9　小结 ……29

第二部分　高级COM编程技巧

第3章　COM+结构与管理 ……31
3.1　COM的发展 ……32
3.2　MTS的缺点 ……33
3.3　COM+结构 ……33
3.3.1　创建COM+对象 ……34
3.3.2　用参数表示的对象结构 ……34
3.3.3　标记 ……35
3.3.4　中立线程单元 ……35
3.3.5　对象池 ……35
3.3.6　对象池管理 ……36
3.3.7　动态负荷均衡 ……36
3.4　COM+配置服务 ……37
3.5　COM+的资源管理 ……37
3.6　开发COM+应用程序 ……38
3.7　队列组件 ……38
3.7.1　放入队列的事务 ……39
3.7.2　管理队列组件 ……39
3.8　松散耦合事件 ……39
3.9　COM+的数据访问 ……40
3.9.1　读取最优化的数据访问 ……40
3.9.2　事务中的共享属性管理器 ……41
3.10　COM+的安全性 ……41
3.11　基本的COM特性 ……41
3.11.1　结构存储 ……41
3.11.2　取消未完成的COM调用 ……42

3.12 小结 ……42
第4章 持久存储 ……43
4.1 IPersist接口 ……43
4.1.1 IPersistStorage ……44
4.1.2 IPersistFile ……45
4.1.3 IPersistStreamInit ……45
4.2 IStream接口 ……46
4.2.1 IStream::Write() ……47
4.2.2 IStream::Read() ……48
4.2.3 IStream::Seek() ……49
4.3 创建实现IPersistStreamInit的ATL对象 …50
4.4 使用一个持久对象 ……54
4.5 简化持久对象的创建 ……56
4.6 简化持久对象的使用 ……58
4.7 小结 ……61
第5章 标记 ……62
5.1 COM+对象和标记 ……62
5.2 探究标记类型 ……66
5.2.1 文件标记 ……66
5.2.2 运行对象表 ……69
5.2.3 项目标记 ……70
5.2.4 组合标记 ……70
5.2.5 类标记 ……71
5.2.6 指针标记 ……72
5.3 小结 ……72
第6章 可连接的对象 ……73
6.1 连接点 ……73
6.2 连接点容器 ……75
6.3 连接点举例 ……75
6.4 事件和VB ……85
6.4.1 重写事件源 ……86
6.4.2 ATL代理程序生成器 ……88
6.4.3 编写VB客户程序 ……90
6.5 各种工具实现事件时有何不同 ……91
6.5.1 事件和VB ……91
6.5.2 事件和C++Builder ……93
6.6 小结 ……96
第7章 COM+线程 ……97
7.1 PC线程的发展 ……97
7.2 COM+线程类型 ……98
7.2.1 工作者线程 ……98
7.2.2 消息队列线程 ……99
7.2.3 窗口线程 ……100
7.2.4 单元线程 ……103
7.2.5 线程池 ……104
7.3 COM+线程模型 ……105
7.3.1 单线程服务程序 ……105
7.3.2 单元线程服务程序 ……106
7.3.3 中立线程服务程序 ……108
7.3.4 自由线程服务程序 ……108
7.4 线程同步 ……111
7.4.1 线程局部存储 ……111
7.4.2 消除并发问题 ……111
7.5 小结 ……117
第8章 COM和注册表 ……119
8.1 注册表API ……119
8.2 Regedit和Regedt32 ……125
8.3 COM的注册表结构 ……127
8.3.1 文件扩展名 ……127
8.3.2 ProgID ……128
8.3.3 AppID ……130
8.3.4 CLSID ……133
8.3.5 接口 ……135
8.3.6 TypeLibs ……136
8.4 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole ……137
8.4.1 允许和禁止DCOM ……138
8.4.2 默认权限 ……138
8.4.3 传统的安全性 ……138
8.5 注册COM+服务程序 ……139
8.5.1 Regsvr32 ……139
8.5.2 自注册进程外服务程序 ……140
8.5.3 框架 ……141
8.6 小结 ……141

第9章 COM+的最优化、继承及集合 ……142
9.1 DCOM的速度 ……142
9.1.1 对象定位 ……143
9.1.2 网络循环 ……143
9.1.3 混合线程模型 ……144
9.2 远程激活 ……148
9.3 远程引用计数 ……149
9.4 代理进程 ……150
9.5 IClassFactory ……153
9.6 继承 ……155
9.7 小结 ……160
第10章 使用NT服务 ……161
10.1 剖析服务 ……163
10.1.1 main()和WinMain ……163
10.1.2 ServiceMain() ……164
10.1.3 ServiceCtrlHandle() ……167
10.2 ATL和服务 ……168
10.3 为使用服务而提供的工具 ……179
10.3.1 Administrative Tools中的Services Applet ……179
10.3.2 Diagnostic实用工具 ……180
10.3.3 Service Controller ……181
10.3.4 Event Viewer ……181
10.4 OpenSCManager() ……181
10.4.1 服务的句柄 ……182
10.4.2 操作服务 ……182
10.5 经由注册表安装服务 ……183
10.6 使用事件日志 ……184
10.6.1 消息编译器 ……184
10.6.2 RegisterEventSource()，Deregister EventSource()和ReportEvent() ……186
10.6.3 事件日志阅读器 ……187
10.7 调试你的服务 ……188
10.7.1 系统账号 ……188
10.7.2 任务管理器：调试 ……188
10.7.3 使用AT命令启动调试器 ……188
10.8 小结 ……189
第11章 调度 ……190
11.1 理解调度 ……190
11.2 类型库调度 ……190
11.3 标准调度 ……191
11.3.1 定义DLL入口点 ……192
11.3.2 类定义 ……193
11.3.3 定义IID、TypeLib GUID和CLSID ……197
11.3.4 代理程序和存根程序的定义 ……198
11.3.5 注册表文件 ……203
11.3.6 转换MIDL的输出文件 ……203
11.4 自定义调度 ……205
11.4.1 声明对象的类 ……206
11.4.2 定义对象的类 ……207
11.4.3 定义代理程序的类 ……211
11.4.4 客户程序 ……214
11.5 小结 ……217
第12章 COM的安全性 ……218
12.1 COM与DCOM的安全性对比 ……218
12.2 Windows安全性 ……219
12.2.1 完善域的安全性 ……219
12.2.2 安全性描述符 ……219
12.2.3 验证 ……230
12.3 模拟 ……232
12.3.1 伪装 ……232
12.3.2 CoImpersonateClient()和 CoRevertToSelf() ……233
12.3.3 伪装 ……235
12.4 说明性安全性 ……235
12.5 程序的安全性 ……235
12.5.1 安全外壳 ……235
12.5.2 IClientSecurity ……236
12.5.3 访问和运行的安全性 ……237
12.6 小结 ……238
第13章 配置和错误处理 ……239
13.1 使用DCOMCNFG配置COM+对象 ……239
13.1.1 传统COM服务程序 ……240
13.1.2 创建自动服务程序 ……242

13.1.3 默认属性 ……244
13.1.4 默认安全性 ……245
13.1.5 配置COM+服务程序 ……249
13.1.6 服务程序的位置 ……250
13.1.7 服务程序的安全性 ……250
13.1.8 服务程序的身份 ……252
13.2 使用OLE2View程序 ……253
13.2.1 OLE2View的缺点 ……254
13.2.2 使用OLE2View配置COM+对象 ……254
13.2.3 指定远程进程内服务程序的代理 …254
13.3 错误处理 ……257
13.3.1 错误处理策略 ……258
13.3.2 通过ISupportErrorInfo传递信息 ……259
13.4 小结 ……264
第14章 COM的互联网服务 ……265
14.1 一个新的COM+传输协议 ……265
14.2 隧道TCP协议概述 ……266
14.2.1 配置隧道TCP协议 ……267
14.2.2 Windows 95和Windows 98中的客户程序配置 ……267
14.2.3 Windows NT 4.0 SP4和Windows 2000中的客户程序配置 ……268
14.2.4 客户机代理服务器的配置 ……268
14.2.5 Windows NT Server 4.0上的服务器配置 ……269
14.2.6 在Windows 2000 Server上配置RPC代理 ……270
14.3 使能CIS ……271
14.4 代理服务器的配置 ……271
14.4.1 配置微软代理服务器 ……271
14.4.2 防火墙的配置 ……272
14.5 配置技巧和已知的问题 ……272
14.5.1 CIS客户端上不正确的代理服务器设置 ……272
14.5.2 关于Multihomed CIS服务器的问题 ……272
14.5.3 MTS对回调的使用 ……272
14.5.4 有关HTTP高速缓存设备的问题 ……272
14.5.5 影响CIS的注册表键 ……273
14.6 OBJREF标记 ……273
14.7 必要的编程改变 ……274
14.8 小结 ……275
第15章 MTS ……276
15.1 商业事务 ……276
15.1.1 协调事务过程 ……277
15.1.2 事务过程与COM ……277
15.2 什么是MTS ……278
15.3 使用MTS的好处 ……278
15.3.1 组件的代理进程 ……278
15.3.2 基于角色的安全性 ……278
15.3.3 准时激活 ……279
15.3.4 MTS资源管理器 ……279
15.3.5 事务协调 ……279
15.3.6 MTS与微软互联网信息服务器的集成 ……279
15.3.7 MTS与微软消息队列服务的集成 …279
15.4 MTS的结构 ……279
15.4.1 程序包 ……280
15.4.2 活动 ……281
15.4.3 角色 ……281
15.5 配置MTS ……281
15.6 MTS对象 ……283
15.6.1 为MTS开发对象 ……283
15.6.2 向一个程序包中添加对象 ……286
15.6.3 程序包的属性 ……287
15.6.4 对象属性 ……288
15.6.5 配置基于MTS的对象 ……289
15.6.6 导出程序包 ……289
15.6.7 导入程序包 ……289
15.7 高级MTS技巧 ……290
15.7.1 为程序包和组件提供安全性 ……290
15.7.2 为程序包创建角色 ……291
15.7.3 给组件或接口分配角色 ……291
15.7.4 通过编程影响安全性 ……291

15.7.5 直接调用者与原始调用者的对比 …292
15.7.6 负载均衡 …293
15.8 创建基于MTS的应用程序 …293
15.8.1 使用MTS进行设计 …295
15.8.2 使用MTS扩展应用程序 …296
15.8.3 远程管理 …296
15.9 小结 …296

第三部分 组件管理与事务

第16章 作为组件管理器的COM+ …299
16.1 COM+编程及其他基于组件的服务 …300
16.2 COM+可扩展性特性 …300
16.3 COM+和标准COM组件 …301
16.3.1 标准COM组件 …301
16.3.2 将标准COM组件用于COM+ …304
16.3.3 COM+对标准COM组件的好处 …306
16.4 通向COM+组件之路 …307
16.4.1 软件复用 …307
16.4.2 性能、可扩展性和稳定性 …313
16.5 COM+和状态 …314
16.5.1 状态的类型 …314
16.5.2 状态存储 …315
16.6 COM+组件必备的条件 …316
16.7 编写COM+组件 …317
16.7.1 环境对象 …317
16.7.2 对象控制 …318
16.7.3 使用ATL编写COM+组件 …319
16.7.4 共享属性管理器 …321
16.7.5 在COM+内引用对象 …324
16.7.6 在COM+内创建对象 …324
16.8 小结 …325
第17章 作为事务协调器的COM+ …326
17.1 对事务的需求 …326
17.1.1 定义的事务 …327
17.1.2 ACID …327
17.2 MS DTC …328
17.3 一个简单的事务例子 …329
17.4 事务协议 …332
17.4.1 OLE事务 …332
17.4.2 XA事务 …332
17.4.3 CICS和IMS事务 …332
17.5 COM+事务编程模型 …332
17.5.1 创建事务 …333
17.5.2 完成事务处理 …337
17.6 旅行社实例 …339
17.7 监视事务 …344
17.8 设计中的考虑因素 …344
17.8.1 提出细粒度的组件 …345
17.8.2 定位靠近其数据源的组件 …345
17.8.3 在同一应用程序中将使用相同资源的组件放在一起 …345
17.9 小结 …345
第18章 COM+的安全性 …346
18.1 COM+的安全概念 …346
18.1.1 角色 …347
18.1.2 安全性的职责 …348
18.2 安全支持供应商接口 …349
18.3 COM+声明安全性 …349
18.3.1 创建角色 …350
18.3.2 将角色加入到组件和接口中 …351
18.3.3 启用安全性 …351
18.3.4 验证 …352
18.4 过程com+安全性 …352
18.4.1 识别用户 …352
18.4.2 给用户授权 …356
18.5 小结 …359
第19章 COM事务集成器 …360
19.1 COMTI的要求 …360
19.2 大型机和Windows DNA …361
19.2.1 SNA Server …362
19.2.2 在COMTI之前 …363
19.2.3 COMTI …363
19.2.4 COMTI警告 …364
19.3 CICS和CICS-LINK …365

19.4 COMTI组件创建器 ……367
19.4.1 组件创建器COBOL向导 ……368
19.4.2 CICS TP ……369
19.4.3 CICS-LINK ……375
19.5 COMTI的管理控制台 ……375
19.6 COMTI运行时间 ……377
19.7 小结 ……379
第20章 负载均衡组件 ……380
20.1 负载均衡组件的定义 ……380
20.2 负载均衡组件的必要性 ……381
20.2.1 可扩展性 ……381
20.2.2 有效性 ……382
20.2.3 灵活性 ……382
20.3 并行性和粒度大小 ……382
20.4 动态负载均衡算法 ……384
20.5 负载均衡组件设计 ……385
20.6 负载均衡组件客户机设计 ……385
20.7 坏消息 ……385
20.8 不用中央并行处理器的负载均衡组件 ……386
20.8.1 用SCM工作 ……386
20.8.2 CoCreateInstance 带来的问题 ……388
20.8.3 创建一个包套 ……390
20.8.4 算法 ……391
20.8.5 时间方法 ……392
20.8.6 时间方法的算法 ……393
20.8.7 负载均衡的实现 ……395
20.8.8 其他均衡和分类技术 ……396
20.8.9 运行时编译执行技术活化 ……396
20.9 小结 ……397
第21章 优化Windows DNA应用程序 ……398
21.1 估计你的需要 ……398
21.2 最优化技巧 ……399
21.2.1 使用用户或系统DSN代替文件DSN ……399
21.2.2 优化算法，特别是反复循环 ……399
21.2.3 避免注册表存取访问 ……400
21.2.4 在任何可能的时候都用运行时编译执行即时激活 ……401
21.2.5 修复资源漏失 ……402
21.2.6 面向对象的负载均衡组件实用性体系结构 ……402
21.2.7 选择工作语言 ……403
21.2.8 避免中间层状态 ……403
21.2.9 避免数据访问中间层 ……403
21.3 使用微软Windows DNA工具包 ……404
21.4 观察测试的结果 ……408
21.5 小结 ……408

第四部分 异步组件程序设计

第22章 松散耦合程序设计 ……409
22.1 什么是消息传递 ……409
22.2 消息传递的优点 ……410
22.2.1 用消息传递加强大型应用程序的开发 ……410
22.2.2 消息传递更好地利用通信资源 ……410
22.2.3 消息传递在不同系统中取得一致 ……411
22.3 消息传递的弱点 ……411
22.3.1 延长处理时间 ……412
22.3.2 异步执行 ……412
22.4 同步与异步程序设计 ……412
22.5 可扩展性 ……414
22.6 面向消息的中间设备 ……414
22.6.1 MOM程序接口 ……415
22.6.2 MOM系统软件 ……415
22.6.3 管理工具 ……415
22.7 微软消息队列服务器 ……415
22.7.1 MSMQ连接器 ……416
22.7.2 MSMQ和别的API ……416
22.7.3 MSMQ和Email ……416
22.8 小结 ……416
第23章 MSMQ管理机构和体系结构 ……417
23.1 MSMQ对象和属性 ……417
23.2 消息 ……419

23.3 队列 ……420
23.3.1 队列类型 ……420
23.3.2 消息队列 ……421
23.3.3 管理队列 ……421
23.3.4 应答队列 ……421
23.3.5 日志队列 ……421
23.3.6 死信队列 ……421
23.3.7 报告队列 ……422
23.4 消息队列信息服务 ……422
23.5 本地队列存储 ……422
23.6 队列属性 ……422
23.7 优先级 ……423
23.8 事务队列 ……423
23.9 标识队列 ……423
23.9.1 路径名 ……423
23.9.2 格式名称 ……423
23.9.3 示例标识符 ……425
23.9.4 标志 ……425
23.9.5 类型 ……425
23.9.6 私有队列 ……425
23.10 机器 ……425
23.11 MSMQ企业 ……426
23.11.1 站点连接 ……426
23.11.2 连接的网络 ……426
23.11.3 MSMQ控制器 ……426
23.12 MSMQ客户机 ……426
23.13 MSMQ管理机构 ……427
23.14 小结 ……427
第24章 MSMQ程序设计 ……428
24.1 MSMQ库API ……428
24.2 用MSMQ库API创建一个应用程序 ……429
24.2.1 格式名称 ……429
24.2.2 路径名 ……430
24.2.3 查找格式名称 ……430
24.2.4 用属性工作 ……430
24.2.5 创建队列 ……431
24.2.6 解散队列 ……431
24.2.7 打开队列 ……432
24.2.8 发送一条消息 ……432
24.2.9 接收一条消息 ……434
24.2.10 关闭队列 ……435
24.3 MSMQ ActiveX控制API ……441
24.4 用COM+接口创建MSMQ应用程序 ……442
24.4.1 定义接口和GUID ……442
24.4.2 初始化COM ……443
24.4.3 创建队列 ……443
24.4.4 变体型 ……444
24.4.5 BSTR ……444
24.4.6 解散队列 ……445
24.4.7 打开队列 ……445
24.4.8 发送消息 ……445
24.4.9 接收消息 ……446
24.4.10 关闭队列 ……448
24.5 用灵巧指针创建一个MSMQ应用程序 …454
24.5.1 定义接口和GUID ……454
24.5.2 .tli文件 ……457
24.5.3 ATL从属物 ……457
24.5.4 创建队列 ……458
24.5.5 解散队列 ……458
24.5.6 打开队列 ……458
24.5.7 发送消息 ……459
24.5.8 接收消息 ……459
24.5.9 关闭队列 ……460
24.6 用VBScript创建一个MSMQ应用程序 …463
24.7 小结 ……466
第25章 高级MSMQ程序设计 ……467
25.1 游标 ……468
25.1.1 MSMQ API 游标 ……469
25.1.2 MSMQ ActiveX组件游标 ……471
25.2 查找队列 ……473
25.3 消息确认、应答和记录 ……475
25.3.1 行政管理队列 ……475
25.3.2 应答队列 ……480
25.3.3 消息ID ……480

25.3.4 记录 ……481
25.4 事务处理 ……481
25.4.1 消息事务处理 ……482
25.4.2 ITransaction ……482
25.4.3 创建事务队列 ……483
25.4.4 事务处理的类型 ……483
25.4.5 外部事务处理 ……487
25.5 MSMQ Email API ……494
25.6 异步操作 ……495
25.6.1 自动事件 ……496
25.6.2 系统事件对象 ……496
25.6.3 回调函数 ……500
25.6.4 完成端口 ……505
25.7 队列安全性 ……510
25.8 小结 ……511
第26章 松散耦合事件 ……512
26.1 一些基本术语 ……512
26.1.1 设计模式 ……513
26.1.2 发布人 ……513
26.1.3 订阅人 ……513
26.1.4 COM+事件服务 ……513
26.2 发布-订阅选项的比较 ……513
26.2.1 轮询 ……513
26.2.2 紧密耦合事件 ……514
26.2.3 过紧密耦合 ……514
26.2.4 要求并行的组件生存期 ……515
26.2.5 无法过滤噪声 ……515
26.2.6 松散耦合事件 ……515
26.3 COM+事件服务 ……516
26.4 事件服务的演示 ……518
26.5 高级COM+事件服务问题 ……523
26.5.1 订单和IEventSubscription接口 ……523
26.5.2 使用短期订阅单 ……524
26.5.3 注册短期订阅单 ……524
26.5.4 取消注册短期订阅单 ……526
26.6 事件过滤 ……527
26.6.1 生成过滤器串 ……527
26.6.2 利用程序生成过滤器串 ……528
26.7 小结 ……529
第27章 队列组件 ……530
27.1 队列组件概述 ……530
27.2 分布式计算及队列组件 ……531
27.2.1 确认所接收的数据 ……532
27.2.2 服务器请求更多的数据 ……532
27.2.3 确认所执行的操作 ……532
27.2.4 需要查找数据 ……533
27.2.5 确定是否排队 ……533
27.3 队列组件结构 ……534
27.3.1 生成并定义一个队列组件 ……534
27.3.2 客户方的队列组件 ……535
27.3.3 服务器方的队列组件 ……535
27.4 编写应用队列组件的一个演示程序 ……536
27.4.1 使用Visual C++和ATL编写一个队列组件 ……536
27.4.2 安装队列组件 ……539
27.4.3 定义COM+应用程序 ……539
27.4.4 标记需排队的COM+应用程序 ……541
27.4.5 向COM+应用程序中添加组件 ……542
27.4.6 标记COM接口为排队的 ……542
27.4.7 用Visual C++编写客户端应用程序 ……543
27.5 测试组件和客户程序 ……548
27.6 导出COM+应用程序 ……549
27.7 小结 ……550
光盘使用说明 ……551

第一部分 Windows DNA和COM+

这部分内容包括：

- COM+：Windows DNA的粘合剂
- 多层组件结构

第1章 COM+：Windows DNA的粘合剂

本章内容：

- Windows DNA
- 谈谈因特网：HTML和XML
- Windows DNA服务
- DNA：功能概述

1.1 Windows DNA

DNA，即分布式互联网应用程序结构，是目前很多软件开发人员常用的一个词。它是有关编写运行于微软分布式环境下（或者运行于经由国际互联网或内部互联网进行浏览的浏览器）的可扩展且强健的应用程序的。互联网的发展正呈现指数增长，还暂时看不到平缓的趋势。微软预见到这一发展趋势，力图使它的IE浏览器成为市场的领导者，不仅在用户和客户端，而且也要在开发端占据领导地位。因此，DNA出现了。

> **汇集** 互联网正在快速地成为大多数新软件首选的开发平台。但是现存的所有程序模型、技术和软件代码该怎么办呢？利用互联网进行会聚的时代即将到来。通过将原有代码封装在组件当中，同时使用DNA进行新的开发，则可能得到最好的客户机-服务器系统（多层结构、分布式工作、事务处理、队列），再加上互联网要素（脚本、平台、可复用组件），就可以生成一个坚固的框架以适应所有现存的技术。这使得一个企业可以逐渐修改原有系统，并以可扩展、可复用、可靠的系统取而代之（参见第19章“COM事务集成器”，详细介绍了传统的系统和DNA）。

DNA是一个抽象概念。对于编写DNA程序来说，不像COM/DCOM/COM+那样有统一的规范，决不存在带DNA标志的需求或者任何要求DNA遵守的规则。

微软在发展DNA上做出了不懈的努力，使之成为编写可扩展的多层互联网程序的稳定框架（多层结构的设计问题将在第2章中进行讨论）。

对于开发者来说，这一框架更像是微软的工具和产品的指南，指导你在何时使用它们进行

多层设计。例如，MSMQ和MTS这两个DNA服务，极大地减轻了实现互联网现有应用程序的工作量，并且提高了可靠性。更重要的是，大多数DNA结构提供的服务对于程序员来说是免费的，尤其是对COM+的介绍。在某种情况下（如MTS），只需要将组件拖放到服务的操作环境中，就可以自动地获得服务提供的所有好处。你可以通过改变组件管理器中的一项设置，轻松实现动态的负荷均衡。

事实上，COM+已经成为独立于DNA的一个整体，它为程序员们提供了开发多层应用程序所必需的底层结构，而无需编写底层结构代码，使他们可以专注于业务问题的解决。

提示　无论你是在编写简单的网上零售商品目录，还是一个完善的企业范围的、可以对用户交互进行控制的、安全的内联网，DNA都可以显著地减少你的Web应用程序的开发时间。

在信息技术管理者看来，DNA极大地减少了为拥有一个系统而付出的代价，并且可以均衡用于传统技术和技能的投资。DNA提供了处理与可扩展互联网应用程序有关的、更复杂的低层通信细节的服务。这一点会极大地减少开发系统所用的经费。

处在DNA前端的是HTML和XML。HTML是一种与平台无关的语言，事实上它允许所有个人和商用电脑进行相互间的通信。XML是HTML的延伸，它给了你极大的灵活性，包括创建COM+对象的能力。

提示　你的应用程序不需要Web或互联网要素就可以得到本章讨论的DNA服务的全部利益。MTS和MSMQ可以像用于分布式环境一样，很容易地用于独立的本地程序的开发。事实上，由于COM的位置透明特性，你的DNA应用程序根本就不知道自己到底是运行在独立环境中，还是分布环境中，而且它根本不关心这些问题。

1.2　谈谈因特网：HTML和XML

普通的HTML（因特网语言）不提供控制流机制或变量。当一个HTML页面创建之后，该页面是静态的，不可能因为我们的输入而改变它的结果。幸好你可以借助动态HTML、XML和ASP来补充HTML，使人产生一种错觉，即你的网络环境是丰富的，并且具有各种状态。依靠这样的程序，状态（保持一个程序的各个变量的能力）也许就不是必需的了。

无状态环境

具有所有装饰性多媒体的浏览器，其实是为那些以浏览为实际目的的人准备的。它们无状态且非线性；也就是说，它们会随机地从一个页面跳到另一个页面，根本没有事先预定的路线。

没有DNA技术的帮助（如IIS会话或DHTML变量或XML scriptlet），基于浏览器的应用程序不可能从一个页面状态保持到另一个页面。这对于使用浏览器作为交互式开发平台的软件开发人员来说，是个很大的挑战。

保持状态对于传统的事件驱动或有限状态机模式的Windows应用程序来说，是不成问题的。为此，HTML、XML和DNA改变了Windows的编程模式（受控的GUI环境），迫使开发人员考虑无状态模式。对于复杂问题，DNA应用程序的用户（网上浏览者）的工作环境将更加无拘无束。

客户或者浏览者可以不经过预定的步骤或状态，随机地浏览一个网站。如果你对自动机理论比较熟悉，浏览者就像一个非确定的有限自动机，它有一个初态和一个终态。

只要将数据嵌入URL就可能实现将信息从一页带到下一页，但这对编程来说是很困难的（需要做很多的分析工作），而且不利于扩展。Cookie是一个很好的选择，因为它可以把服务器端的信息保存在客户端，但对于较老的浏览器来说是不具有这项功能的，而且用户也可能因为安全或隐私的原因关掉这项功能。

COM+可以作为解决这一问题的桥梁。它可以很容易地将信息保存在客户机上，可以解决大多数由保存状态而引发的问题。

另一方面，服务器面临着这样的问题，当一个表面上是匿名的连接随机地从一页跳到另一页时，如何追踪它。通过映射客户的IP地址并使用定时器，可以创建会话线程，并将每个访问服务器的浏览器的状态记录下来。这是微软的IIS为客户保存状态的一种方法。你可以从下面几部分更加详细地了解到这种方法和其他的几种方法，但首先你必须了解Windows DNA服务。

1.3 Windows DNA服务

从右到左读图1-1，你可以从一个比较高的层次开始了解DNA。这张图突出了DNA框架中相关的服务和技术。再说一遍，DNA没有标准，这张图只代表作者眼中的DNA。不存在任何关键组成部分使你的程序适应DNA。这张图提出的是一种典型的DNA框架的安排。

提示 如果你对三层结构模型比较熟悉，你会注意到下面的讨论中提到了它的模式。如果你没注意到也没关系，第2章中将深入探讨。

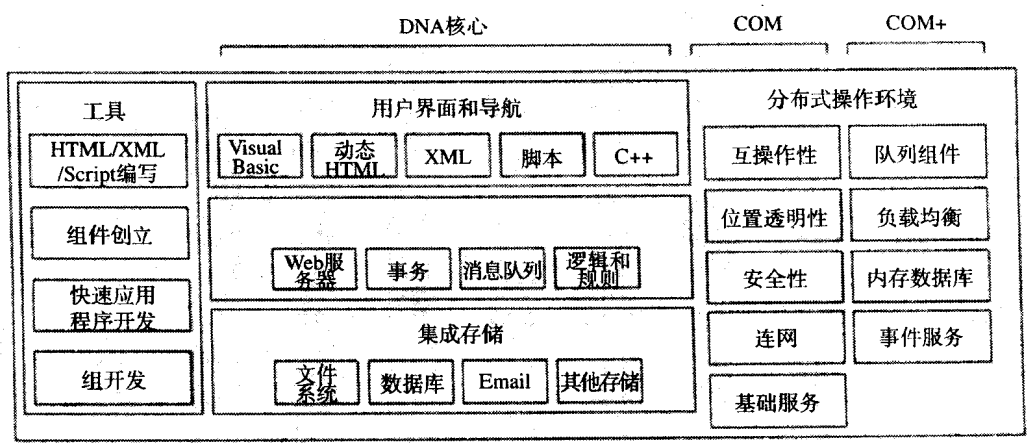

图1-1 Windows DNA服务

1.3.1 Windows DNA服务：COM和COM+

为了使DNA工作，操作系统的内核必须予以支持。一个特权服务必须随时存在，以完成通信协议的解释和检查以及组件的协作。这就是为什么COM/DCOM/COM+（以及服务控制管理器）