



软件开发技术丛书

Windows 2000

编程技术内幕

Programming Microsoft Windows 2000 Unleashed

(美) Mikey Williams 著 前导工作室 译



机械工业出版社
China Machine Press

SAMS

软件开发技术丛书

Windows 2000 编程 技术内幕

(美) Mickey Williams 著

前导工作室 译



机械工业出版社
China Machine Press

本书是Mickey Williams 所著的一本讲述Windows 2000 编程的专著，全书共分为五个部分：Windows 2000 核心技术、用户交互/桌面、COM 和ActiveX、COM+以及分布式Windows 2000 服务，详细介绍了Windows 2000 所实现的各种服务，以及它所提供的API 函数。本书结构严谨，内容全面。作者提供了大量的实例，使读者能够轻松地理解复杂的概念及原理。所以，本书是学习Windows 2000 应用程序开发的一本难得的入门教材，同时也是Windows NT 编程人员必不可少的一本参考资料。

Mickey Williams: Programming Microsoft Windows 2000 Unleashed.

Authorized translation from the English language edition published by Sams Publishing, an imprint of Macmillan Computer Publishing U.S.A.

Copyright 1999 by Sams Publishing.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Chinese Simplified language edition published by China Machine Press.

Copyright 1999 by China Machine Press.

本书中文简体字版由美国麦克米兰公司授权机械工业出版社独家出版，未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

版权所有，翻印必究。

本书版权登记号：图字：01-1999-2601

图书在版编目(CIP)数据

Windows 2000 编程技术内幕 / (美) 威廉姆斯 (Williams, M.) 著; 前导工作室译.
—北京: 机械工业出版社, 1999.12

(软件开发技术丛书)

书名原文: Programming Microsoft Windows 2000 Unleashed

ISBN 7-111-07632-X

I. W… II. ①威… ②前… III. 窗口软件, Windows2000—程序设计 IV. TP316

中国版本图书馆CIP数据核字(1999)第64230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: 陈剑瓯

北京昌平第二印刷厂印刷·新华书店北京发行所发行

1999年12月第1版第1次印刷

787mm×1092mm 1/16·44.75 印张

印数: 0 001-7 000 册

定价: 88.00 元(附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

1993年5月24日Microsoft 公司正式推出了Windows NT, 几年来该操作系统不断推出新的版本, 这从一个侧面反映出人们对它的关心和重视程度。对于国内的广大用户来说, Windows NT 的最新版本——即Windows 2000, 既充满了不可抗拒的诱惑, 同时也意味着严峻的挑战。

在这种情况下, 广大Windows NT 应用程序开发人员对于有关Windows 2000 参考资料的需求与日俱增。我们很高兴能有机会及时将Mickey Williams 所著的这本《Windows 2000 编程技术内幕》介绍给大家。无论从深度还是从广度上看, 本书所具有的价值都是毋庸置疑的。

概括地说, 本书主要具有以下几个特点:

1) 实用。本书无论从内容的安排, 还是从例子的选择来看, 都极具实用性。它所讲述的内容都是学习和掌握Windows NT 编程所不可或缺的。所举的例子有大有小, 小的例子有时只介绍某个函数的使用, 大的例子可涉及Windows 2000 某一特征的编程实现。本书的主要目的非常明确, 那就是尽快帮助读者学会Windows 2000 的编程。

2) 全面。有关Windows 2000 编程的内容非常繁杂。但本书的作者过去曾经写过《Windows NT 4.0 编程指南》一书。他以丰富的实践经验, 在本书中覆盖了绝大多数常用的Windows 2000 编程的内容, 使本书足以作为一本编程指南和使用手册。

3) 易读。要介绍Windows 2000 编程, 其内容必然是专业性很强、很复杂的。但本书却能做到结构严谨, 语言简洁明快。作者下笔有条不紊, 分析详细透彻, 读起来引人入胜, 读者丝毫不会感到困难。

本书由前导工作室全体人员进行翻译、录入和校对。由于原书中有较多的新术语, 加上译者水平有限, 译文中难免有不当之处, 恳请广大读者批评指正。

前导工作室
1999年9月

前 言

Windows 2000 保留了Microsoft 1992 年推出Windows NT 3.1 以来的一些传统。Windows NT 3.1 可以满足以下条件:

- 平台之间的可移植性。
 - 随市场需求变化而扩展的能力。
 - 多处理和可扩展性。
 - 可靠性和健壮性 (换句话说, 系统应该能够自我保护)。
 - 分布式计算。
 - 符合POSIX 标准。
 - 政府认证的安全机制。
 - 与Microsoft 以前的产品保持兼容。
 - 性能优良 (在满足以上条件的基础上, 运行速度尽可能快)。
- 自从推出最早的Windows NT版本以来, 该系统一直在不断地改进:
- Windows NT 3.5 在运行速度方面有很大的提高, 它还支持OpenGL。
 - Windows NT 3.51 引入了对PowerPC 和COM 的支持。
 - Windows NT 4.0 加入了对Windows 95所引入的shell 的支持。

Microsoft 对Windows 2000 寄予厚望, 希望它能成为Windows NT 的一次重要升级。下面是Windows 2000 的一些新特征:

- 活动目录 (Active Directory) 活动目录是一个企业范围的、可扩展的目录服务。活动目录支持工业LDAP 协议以及ADSI 协议。ADSI 是最近为Windows NT 4.0 引入的基于COM的目录管理协议。
- COM COM有两个重要的扩充。第一个是COM+, 它是COM 的一个运行时服务补充, 支持基于组件的分布式应用程序的创建。第二个扩充是异步COM, 它允许在客户程序和服务器程序之间做异步的方法调用。
- 提高了安全性 Windows 2000 可以支持SSL、Kerberos 以及公共密钥加密。此外, Windows 2000 引入了一些新的API函数, 这些函数可以简化安全性编程。
- VLM 通过使用甚大内存模型 (Very Large Memory Model), 应用程序可以使用28GB的数据地址空间。最初只有Alpha 处理器才支持VLM。

此外, Windows NT 4.0 通过服务包或者NT 4.0 选项包所引入的许多特征现在集成到了Windows 2000 中。例如Microsoft 消息队列服务程序 (Microsoft Message Queue Server), 这一特征现在已经紧密地集成在Windows 2000 中了。

本书适用对象

适用本书的读者是中级和高级C/C++ 程序员——即正在开发商业应用程序并且需要了解Windows 2000 的新特征的开发人员。如果应用程序的用户需要新的特征, 并且希望充分利用Windows 2000 的强大功能, 那么开发人员也可以参考本书。本书是专门提供给开发人员使用的。

本书中的例子既有Visual Basic 的，也有C和C++ 的，这要看哪种语言适合讲述那部分内容。例如，有些消息队列的例子使用的是VB，这是为了向读者介绍开发人员怎样在编写很少代码的情况下利用VB对消息队列的支持。

本书中的代码

在本书的例子中，读者可以看到一些代码段。这些代码主要是用Visual C++ 编写的。我们尽力给读者提供有用的例子，而不是提供一些填充页面的代码垃圾。本书的光盘上包含了书中所引用的工程。

本书内容安排

笔者写本书的第一步是编写出一个内容表，其格式与笔者以前所写的关于Windows NT 4 的书是一致的。这本新版本的书基本部分都重新写过了，并加入了介绍COM 和COM+ 特征的内容。本书还介绍了其他书很少涉及的内容，如Microsoft 集群服务器（Microsoft Cluster Server）和Microsoft 消息队列服务程序。

第一部分 Windows 2000 核心技术

本书的第一部分讨论了使用Windows 2000 创建分布式应用程序所涉及的一些基本概念。其中，第1章简要介绍了创建具有DNA 风格的应用程序的过程，而随后的9章则介绍了一些有关的主题，例如Windows 2000 中进程和线程的管理、分布式安全性和编写服务等。

第二部分 用户交互 / 桌面

本书的第二部分主要讨论了与Windows 2000 桌面有关的一些专题。这一部分的两章内容主要介绍了与高级GDI 有关的一些主题，例如如何处理256色的位图和如何使用所有者绘图函数和定制绘图函数等内容。

第三部分 COM 和ActiveX

本书的第三部分主要讨论了与COM 和ActiveX 有关的一些知识。COM 现在已成为Windows 2000 操作系统不可缺少的一部分，我们很难想象在不使用COM 的情况下能编写出合适的Windows 2000 应用程序。这一部分的前几章首先介绍了COM 的相关知识，而随后的几章则讨论了如何创建定制COM组件与ActiveX控件以及如何使用异步COM 等内容。

第四部分 COM+

本书的第四部分主要讨论了构建于COM 基础之上的COM+ 服务，它是在Windows 2000 中首次发布的一种新技术。通过把COM+ 和COM 以及Microsoft Transaction Server 结合在一起，用户便可开发出以组件为基础的可扩充的分布式应用程序。

第五部分 分布式Windows 2000 服务

本书的最后一部分，即第五部分，主要介绍可支持开发分布式系统的Windows 2000 编程的有关主题。这一部分涉及到Microsoft 集群服务器、活动目录和消息队列服务程序等内容。

本书中使用的约定

位于一行代码开头的箭头 (➡) 称为续行标识符，它表示这一行代码太长，在一行中写不完而必须打断的位置。用户应该把箭头标识的后面字符看作是前一行的一部分。

注意 注意部分表示与所讨论的内容有关的一些较令人感兴趣的信息。

提示 提示部分给出了一些有益的建议以及完成某一任务更为简便的方法。

警告 警告部分指出了可能存在的一些潜在问题，它可以帮助用户避免出现一些灾难性的后果。

原出版社网址

www.sampublishing.com

原书书号

ISBN 0-672-31486-X

本书的内容是根据Windows 2000的Beta版写作的。这样，在Windows 2000正式发布以前，用户可以预先掌握这些知识。然而，在Windows 2000正式发布时，该系统可能会有一些改动。所以，在Windows 2000正式版发布以后，我们希望本书的读者能访问我们的网址www.sampublishing.com，以便查找更新的内容。

此外，用户可以访问MSDN网址msdn.microsoft.com，以便获得有关Windows 2000的最新信息。

目 录

译者序
前言

第一部分 Windows 2000 核心技术

第1章 Windows DNA 概述	1
1.1 Windows 和Windows NT 的历史	1
1.2 Windows DNA 剖析	2
1.3 Windows DNA 的结构	3
1.3.1 COM	4
1.3.2 表示层	4
1.3.3 防火墙层	6
1.3.4 业务逻辑层	7
1.3.5 数据层	10
1.4 小结	11
第2章 虚拟内存和内存管理	12
2.1 使用虚拟内存	12
2.2 使用虚拟内存的时机	13
2.2.1 保留一定的虚拟内存	14
2.2.2 提交一个虚拟内存页面	15
2.2.3 锁定虚拟内存页面	18
2.2.4 释放虚拟内存页面	19
2.2.5 为其他进程分配内存	19
2.3 使用虚拟内存的例子	19
2.4 Windows 2000 堆函数	27
2.4.1 Windows 2000 下的堆管理	27
2.4.2 在C++ 类中使用私有堆	28
2.5 小结	33
第3章 线程和进程	34
3.1 进程和线程	34
3.2 管理进程	36
3.2.1 使用CreateProcess 函数	36
3.2.2 结束进程	38
3.3 管理线程	38
3.3.1 创建线程	38

3.3.2 结束线程	42
3.3.3 取出并改变线程的优先级	42
3.3.4 线程局部存储	43
3.3.5 判断线程的标识符	44
3.3.6 使用线程相似性	45
3.3.7 使多个线程合作	47
3.3.8 何时应创建线程	49
3.3.9 何时不应创建线程	49
3.4 同步	50
3.4.1 使用Win32互锁操作	50
3.4.2 临界区	52
3.4.3 管理事件	54
3.4.4 信号灯与互斥	59
3.5 信号量	61
3.5.1 信号量和等待函数	62
3.5.2 信号量应用程序	63
3.5.3 创建信号量	64
3.5.4 返回信号量控制的资源	64
3.6 在应用程序中使用信号量和临界区	64
3.7 使用作业对象	70
3.7.1 控制作业对象属性	71
3.7.2 作业对象实例	71
3.8 小结	75
第4章 文件	76
4.1 使用Windows 2000文件系统	76
4.2 执行通用文件操作	76
4.2.1 收集卷信息	76
4.2.2 文件系统标志	77
4.2.3 例子	78
4.2.4 打开、关闭和删除文件	80
4.2.5 写文件	82
4.2.6 读文件	83
4.3 查找文件	85

4.3.1 使用FindFirstFile、FindNextFile 和FindClose	85	6.1.3 使用DLL的好处	132
4.3.2 即将面世的FindFirstFileEx	88	6.1.4 使用DLL的不利之处	132
4.3.3 使用FILETIME 结构	89	6.2 创建和使用动态链接库	132
4.4 执行其他文件操作	91	6.2.1 DLL 初始化	133
4.4.1 使用CopyFile	91	6.2.2 导出和导入DLL 函数和数据	134
4.4.2 使用MoveFile	91	6.2.3 在装入期间装入DLL	137
4.4.3 使用SetFilePointer	92	6.2.4 在运行时装入DLL	137
4.4.4 使用GetFileSize	93	6.2.5 DLL 版本控制	138
4.5 使用异步输入输出	93	6.2.6 创建DLL 共享内存	139
4.5.1 使用重叠的输入和输出	94	6.2.7 简单的DLL 实例	141
4.5.2 使用文件完成例程	97	6.2.8 使用C++ 和DLL	144
4.5.3 使用文件完成例程的例子	98	6.2.9 使用MFC 和DLL	146
4.6 利用加密文件系统	104	6.3 通用替换程序	152
4.7 使用文件通知函数	108	6.3.1 通用替换程序的结构	153
4.8 小结	112	6.3.2 通用替换程序函数	154
第5章 结构化异常处理	113	6.3.3 通用替换程序实例	155
5.1 异常处理的基础知识	113	6.3.4 通用替换程序的调试	163
5.2 SEH工作原理	114	6.4 小结	163
5.3 使用终止处理程序	114	第7章 分布式安全性	164
5.3.1 使用__leave 来离开__try 块	116	7.1 Windows 2000 安全性概述	164
5.3.2 使用正常和反常终止	117	7.2 基本安全性数据结构	165
5.4 使用异常处理程序	117	7.2.1 进程和线程安全性令牌	165
5.4.1 使用异常过滤器	118	7.2.2 安全标识符	167
5.4.2 处理浮点异常	121	7.2.3 ACE、DACL 和SACL 结构	171
5.5 C++ 异常处理	122	7.2.4 判断ACL 所需的大小	172
5.5.1 使用异常来检测错误	122	7.2.5 给DACL 添加允许访问的ACE	173
5.5.2 在错误产生后使用异常清除	123	7.2.6 安全描述符	174
5.5.3 使用标准异常库	124	7.2.7 获取安全描述符	176
5.5.4 使用运行异常	125	7.2.8 使用安全属性结构	178
5.5.5 使用逻辑异常	126	7.3 基于委托的访问控制	178
5.5.6 检测构造时的错误	127	7.3.1 TRUSTEE 结构	179
5.6 把Win32 SEH 与C++ 异常处理集成 起来	128	7.3.2 ACTRL_ACCESS_ENTRY 结构	181
5.7 小结	129	7.3.3 ACTRL_ACCESS_ENTRY_LIST 结构	182
第6章 动态链接库	130	7.3.4 ACTRL_PROPERTY_ENTRY 结构	183
6.1 理解库	130	7.3.5 ACTRL_ACCESS 结构	183
6.1.1 静态链接库	131	7.4 模仿客户程序	187
6.1.2 动态链接库	131		

12.2.3 处理WM_COMPAREITEM消息	276
12.2.4 处理WM_DELETEITEM消息	277
12.3 所有者作图实例: OwnDraw	277
12.3.1 创建CListItem结构	278
12.3.2 处理所有者作图消息	278
12.3.3 向列表框中加入项目	278
12.3.4 处理WM_MEASUREITEM消息	279
12.3.5 处理WM_DRAWITEM消息	279
12.3.6 处理WM_DELETEITEM消息	280
12.4 使用定制作图	281
12.4.1 定制作图所使用的数据结构	281
12.4.2 CustDraw: 使用MFC的定制作图实例	285
12.5 小结	291
第三部分 COM 和ActiveX	
第13章 COM 和OLE	293
13.1 COM 和OLE 概述	293
13.2 常用COM 和OLE 技术	293
13.2.1 复合文档	293
13.2.2 可视化编辑	295
13.2.3 自动化	296
13.2.4 OLE 数据传输	296
13.2.5 拖放	296
13.2.6 与其他应用程序集成	296
13.3 使用COM 和OLE 编程	296
13.3.1 使用COM 接口	297
13.3.2 标识COM 接口和对象	297
13.3.3 处理GUID	298
13.3.4 处理版本问题	299
13.3.5 使用IUnknown 接口	299
13.3.6 创建对象	302
13.4 COM 组件的一个例子	305
13.4.1 使用Shell 扩展机制	305
13.4.2 理解快捷菜单扩展	306
13.4.3 创建实例	306
13.5 小结	319
第14章 自动化	320
14.1 创建使用自动化的可编程应用程序	320
14.2 IDispatch 接口	320
14.2.1 IDispatch 使用的数据类型	320
14.2.2 IDispatch 中的函数	325
14.2.3 IDispatch 接口与虚函数表	326
14.3 创建使用MFC 的自动化服务程序	327
14.3.1 修改自动化选项	328
14.3.2 修改文档类	328
14.3.3 为AutoBub 创建对话框	330
14.3.4 为AutoBub 画泡泡对象	332
14.3.5 往AutoBub 中加入自动化支持	332
14.4 使用Visual Basic 创建自动化控制程序	335
14.5 在VBScript 中使用自动化	337
14.6 小结	339
第15章 OLE 拖放	340
15.1 OLE 剪贴板	340
15.1.1 使用FORMATETC 和 STGMEDIUM 结构	340
15.1.2 处理FORMATETC 结构	341
15.1.3 使用DVTARGETDEVICE 结构	342
15.1.4 使用STGMEDIUM 结构	343
15.1.5 使用IDataObject 接口	344
15.2 使用OLE 拖放	345
15.2.1 OLE 拖放来源	346
15.2.2 OLE 拖放目标	346
15.3 拖放实例	346
15.4 小结	354
第16章 COM 线程化模型	355
16.1 COM 线程化方案	355
16.1.1 进程内COM 服务程序	355
16.1.2 进程外服务程序	356
16.1.3 管理进程间的内存	357
16.1.4 使用住所中的COM	359
16.2 STA 和MTA	360
16.2.1 为每个线程选择一个住所	360
16.2.2 线程的最佳住所	361
16.2.3 为COM 对象选择住所	361
16.2.4 在客户程序和COM 对象间不匹配的住所	362

16.3 在COM 住所间调度	367	18.3.2 事件	413
16.3.1 共享接口指针的错误方法	368	18.3.3 方法	413
16.3.2 共享接口指针的正确方法	370	18.4 一个ActiveX 控件示例	413
16.4 小结	373	18.4.1 创建工程	414
第17章 COM 对象	374	18.4.2 绘制控件	414
17.1 使用IDL	374	18.4.3 定义AxEdit 的属性	415
17.1.1 深入MIDL 属性	376	18.4.4 字符输入处理	417
17.1.2 编译MIDL 源文件	376	18.4.5 修改控件的位图	419
17.1.3 使用类型库	377	18.5 测试ActiveX 控件	419
17.1.4 使用IDL 中的结构	377	18.5.1 选择控件的测试容器	419
17.1.5 使用IDL 中的枚举类型	378	18.5.2 使用TSTCON32 测试容器	419
17.1.6 指针和IDL	378	18.5.3 测试属性	420
17.1.7 使用IDL 中的方向属性	380	18.6 使用ATL 创建ActiveX 控件	421
17.2 使用ATL 创建定制组件	380	18.6.1 使用ATL 实现 stock 属性	421
17.2.1 常用的ATL 类	380	18.6.2 使用ATL 实现custom 属性	422
17.2.2 ATL COM 接口映射	381	18.6.3 通过ATL 使用ambient 属性	424
17.2.3 使用ATL 向导	381	18.6.4 在ATL 工程中实现事件	424
17.2.4 将代理/存根代码合并到DLL 中	385	18.6.5 添加消息和事件处理程序	426
17.2.5 一个定制COM 对象的示例	385	18.7 用ATL 构建的ActiveX 控件	427
17.3 使用moniker	395	18.7.1 PopButton 的基本设计思路	427
17.3.1 系统moniker 的示例	395	18.7.2 创建PopBtn 工程	429
17.3.2 创建moniker 实例	395	18.7.3 用Visual Basic 测试PopButton	
17.3.3 将moniker 转换为文本字符串	396	控件	443
17.3.4 根据显示名创建moniker	398	18.8 小结	444
17.4 连接点	398	第19章 异步COM	445
17.5 定制调度	400	19.1 COM 中的同步方法和异步方法	445
17.5.1 深入定制调度	400	19.2 Windows 2000 中对异步方法的	
17.5.2 深入按值调度	402	支持	446
17.5.3 实现定制代理	402	19.2.1 IDL 中的异步支持	446
17.5.4 在服务器上实现IMarshal	407	19.2.2 Windows 2000 中其他新的异步	
17.5.5 比较使用定制代理和标准代理		组件	447
的性能	410	19.2.3 使用Fire 和Forget 模式	452
17.6 小结	410	19.3 一个异步COM 的示例	452
第18章 ActiveX 控件	411	19.3.1 IShop 接口	452
18.1 什么是ActiveX 控件	411	19.3.2 AsyncTest 工程	453
18.2 ActiveX 控件接口	411	19.3.3 一个同步测试客户	455
18.3 ActiveX 控件的属性、事件和		19.3.4 IShop 接口的第二版	457
方法	412	19.3.5 一个异步测试客户	458
18.3.1 属性	412	19.4 小结	461

第20章 DCOM	462	22.1.3 活动	522
20.1 位置透明性	462	22.2 编写事务组件	523
20.1.1 创建远程对象的实例	463	22.2.1 创建通用事务组件	523
20.1.2 使用IMultiQI 接口	466	22.2.2 使用Visual C++ 创建事务性 COM+对象	524
20.1.3 确定外部引用是否有效	467	22.2.3 使用Visual Basic 创建事务性 COM+对象	531
20.2 DCOM中的安全性	467	22.3 使用COM 编译器支持类	533
20.2.1 安全性的类型	467	22.3.1 TLH和TLI: 检查输出	533
20.2.2 深入调用安全性	468	22.3.2 COM编译器命令属性	536
20.2.3 深入模仿与伪装	468	22.3.3 使用com_error 产生和捕捉 错误	536
20.2.4 设置计算机的安全级别	469	22.3.4 使用任何数据: _variant_t	537
20.2.5 设置进程的安全属性	471	22.3.5 使用BSTR:_bstr_t	538
20.2.6 深入安全模块	482	22.4 使用多个COM+ 对象	538
20.2.7 设置接口的客户端安全属性	483	22.4.1 组合COM+ 对象的事务生命 周期	538
20.2.8 管理服务器端的安全性	485	22.4.2 有多个对象参与的事务的输出	540
20.3 使用DCOM 配置程序	486	22.4.3 错误处理	541
20.3.1 使用DCOM Config 设置系统 端属性	486	22.4.4 传递参数和接口	541
20.3.2 使用DCOM Config设置特定 AppID 的属性	486	22.5 小结	543
20.4 小结	487	第23章 使用IIS 与COM+	544
第四部分 COM+			
第21章 COM+服务简介	489	23.1 Web 应用程序基础	544
21.1 事务简介	489	23.1.1 ASP 概览	544
21.1.1 ACID: 事务处理的四大特性	490	23.1.2 脚本示例	545
21.1.2 事务的分类	491	23.1.3 ASP 应用程序	545
21.1.3 两阶段提交协议	492	23.1.4 内置的ASP 对象	549
21.2 深入COM+ 内的事务	493	23.2 构建Web应用程序	549
21.3 构建一个可支持事务处理的COM+ 组件	495	23.3 实现方法	552
21.3.1 使用Visual Basic 构建COM 对象	495	23.3.1 对HTML 窗体的处理	552
21.3.2 启示	509	23.3.2 管理cookie	554
21.3.3 向COM+ 应用程序中添加组件	510	23.3.3 创建cookie	554
21.4 小结	518	23.3.4 ServerVariables 集合	556
第22章 构建COM+ 服务组件	519	23.3.5 对ASP 错误的处理	557
22.1 好的Component Service 组件	519	23.3.6 定义全局关键字	558
22.1.1 生命周期/无状态	519	23.3.7 强制次序	560
22.1.2 无状态对象	521	23.3.8 用JavaScript 简化重定向过程	563
		23.3.9 管理对象的更为简单的方法	565
		23.3.10 使用二进制数据	565

XIV

28.2.1 队列属性	662	第29章 集群服务器	685
28.2.2 消息属性	663	29.1 集群服务器的结构	685
28.2.3 机器属性	664	29.1.1 集群服务器的硬件	685
28.2.4 内存分配	664	29.1.2 MSCS 软件组件	686
28.3 使用Win32 API 创建和管理消息 队列	665	29.1.3 集群的启动和恢复	688
28.3.1 创建消息队列	666	29.2 集群API	688
28.3.2 打开已有的队列	670	29.2.1 集群对象管理API	688
28.3.3 向消息队列发送消息	671	29.2.2 资源DLL API	693
28.3.4 从消息队列中接收消息	675	29.2.3 集群管理扩展API	696
28.3.5 关闭消息队列句柄	679	29.3 一个可感知集群的示例	697
28.3.6 删除消息队列	679	29.3.1 ClusterQuote 工程	697
28.4 用Visual Basic 编写的消息队列客户 程序	681	29.3.2 QuoteClient 工程	698
28.5 小结	684	29.3.3 High Availability Quotes 集群资源 类型	698
		29.4 小结	700

第一部分 Windows 2000核心技术

第1章 Windows DNA概述

本章内容

- Windows和Windows NT的历史
- Windows DNA剖析
- Windows DNA的结构

本章将讲述Windows 2000在Windows分布式Internet应用程序体系结构（Windows Distributed Internet Application Architecture, Windows DNA）中所处的地位。Windows DNA的完整概述可能会有一点模糊，下面我们给出一个简单的定义。

注意 Windows DNA是一种操作系统的实现，它提供了大量的服务和技巧，使得用户可以使用逻辑设计过程来创建应用程序。

在这个定义中有三点是比较重要的。第一是“操作系统”，操作系统是计算机系统的基础，它还是整个框架的基础。这个操作系统必须是快速的、稳定的和可扩展的。第二点是这个操作系统提供一系列的服务和技巧。这些服务可以完成一定的任务，比如事务处理或发送消息。最后一点是，操作系统和服务应该结合起来，这样就可以以它们为基础来开发应用程序。应用程序的开发过程应该是一个逻辑过程。

本章将讲述Windows DNA是什么，并详细解释各种服务和技巧。

1.1 Windows和Windows NT的历史

为了理解Windows DNA，读者需要知道Windows发展到了什么地步。这一点是很重要的，因为只有这样才能了解Windows DNA的发展过程。所以，让我们来回顾一下Windows的历史：

- **Windows 3.x** Windows家族是随着Windows 3.0的发行而开始在世人面前亮相的。因为它具有丰富多彩的用户界面，这个版本的Windows相当流行；然而，除了界面以外，Windows 3.x还带来了一些更重要的东西：公共用户界面和设备驱动程序模型。在Windows之前，设备驱动程序必须专门为每个DOS应用程序定制。这是很麻烦的。如果应用程序中带有定制的驱动程序，那么它们必然会有定制的内存需求和特定的问题。Windows改变了这一切，客户可以使用Windows设备驱动程序。用户界面和应用程序的任务分配也很出色，它们允许使用统一的用户界面来完成一定的任务。
- **Windows NT** Windows 3.x的问题是缺乏稳定性。Windows NT是一个为企业使用而开发的新操作系统。从某种程度上讲，它是Windows 3.x的一个稳定的版本，它包括集成的网络和服务器组件（如服务）。然而，服务器服务还不足以书写服务器方应用程序，因为它没有提供事务、消息通信等等。在最初引入Windows NT时，它有三个版本，但从技术的角度讲，这三个操作系统并没有什么不同，客户访问的许可人数也没有什么不同。

- Windows 95/98 Windows家族继续发展，Windows 95和98取代了Windows 3.x。Windows 95和98是32位操作系统。但它们与Windows NT在特征方面存在一些不同之处。Windows 95/98是专门提供给一般用户使用的操作系统，这就要求设备可以自动地完成配置工作，这样客户计算机就更容易管理了。这是一个艰巨的任务，因为有超过一亿台计算机在运行Windows程序。Windows 95/98的重点是为了更容易访问网站，以及执行那些需要访问网站的特定任务。
- NT选项包 Windows NT提供了一个稳定的和可扩展的操作系统的基础。然而，它并没有提供人们所需要的某些服务，NT选项包实现了这个功能，选项包内含IIS、MSMQ和MTS服务等等。这些增加的功能满足了Microsoft开发的某些需求——它标志着独立的软件开发人员和需要这些服务的企业应用程序开发人员之间的不同。
- Windows 2000 Windows 2000的发行带来了更多的服务以及更统一和完善的COM+。Windows 2000是一个能够处理大型任务的操作系统，它所处理的大型任务可以大到涉及几千个并发用户。

1.2 Windows DNA剖析

从上下文看，Windows DNA 的问题是怎样开发健壮的和可扩展的应用程序。健壮性和可扩展性这两个属性似乎表明Windows DNA 是与服务器有关的。这种看法是错误的，因为Windows DNA 适用于客户机、服务器以及任何其他机器。它是一个较为通用的解决方案，它的定义如下：

- 它可以通过键盘、屏幕和/或鼠标与用户进行交互。
- 它可以使用其他输入设备（如触摸屏和笔式输入设备），它甚至还可以用声音来驱动。
- 它可以通过网络与其他机器相联接。
- 它提供可靠的和统一的环境，在这个环境中，多台计算机可以共享数据。
- 它可以很轻易地管理自己以及它的环境。

与传统的Windows API 比较起来，这种类型的解决方案在需求方面发生了巨大的变化。在传统的Windows API 时代，开发人员关心的是他的或她的用户界面有多酷，以及画些什么样的按钮图形等等。这一切发生了改变，因为计算机已经变成了我们日常生活的一部分。所以，它必须解决更多的任务，并且可以和其他计算机及环境相交互。人们只希望计算机能够完成所需的功能，就跟烤箱和微波炉一样。

解决方案

Windows DNA 提供了一种解决方案，可以利用一系列的服务和技巧来实现所需的功能，既然所需的功能如此之大，那么是不是Windows DNA 只适用于团体客户呢？答案是否定的。理由很简单，因为过去的经验表明：只适用于特定用户的策略都是注定要失败的。比如说EDI——它虽然可以被大公司所接受，但是对小公司来说却完全不适用。所以，Windows DNA 必须能满足大公司、小公司和普通用户的需求。

Windows DNA 的规则

Windows DNA 平台是一种多层分布式应用程序模型（下一节我们将详细讨论该平台）。它提供了一系列的服务和技巧，这些服务和技巧是该操作系统基础结构的一部分。这些服务组合了个人计算机、大型主机、Internet、客户/服务器等等的功能。

因为它组合了所有这些技术，所以看上去该操作系统应该是很庞杂的。但是，这种看法不