

<<基于组件的应用程序设计>>

图书基本信息

书名：<<基于组件的应用程序设计>>

13位ISBN编号：9787301041857

10位ISBN编号：7301041853

出版时间：1999-10

出版时间：北京大学出版社

作者：Mary Kirtland(美)

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<基于组件的应用程序设计>>

### 内容概要

Designing  
Component - Based  
Applications

基于组件的应用程序设计

使用经过论证的方法开发可伸缩、  
高性能的企业范围的应用程序

这是一本基于COM和MTS的正式参考文档。

《基于组件的  
应用程序设计》以图例的方式介绍了有效的COM驱动设计和  
MTS部署的原则，这些原则都是经过MicrosoftCOM和  
MicrosoftVisualStudio设计小组的合作项目IslandHopper严  
格检验过的。

使用一套经过多个3 - 2 - 1项目顾问亲自测试的  
实际开发规则，构建了这个功能全面的企业应用程序。

当我们  
们将重点放在该模型应用程序的每个方面时，您还会了解到  
如何应用这种经过论证的方法快速设计、测试、调试及部署  
您自己的多层、基于组件的程序。

在《基于组件的应用程序设计》中，您还将发现一些从设计  
到部署的明确指导：

MTS技术概述 研究DNA结构，回顾COM的基本原则，  
并了解它们是如何一起工作，将前所未有的可伸缩性和灵活  
性带人到企业开发中

应用程序设计策略 确定在三层的MTS环境下设计和部署  
特定应用程序的挑战；构建数据和商务对象、组件及表示层

准备开始 应用调试、疑难解答及性能验证程序

扩展 使用诸如“Microsoft消息队列服务器”和COM+  
等工具来扩展MTS应用程序；了解了Microsoft的COM +  
分布式计算策略就可以把握您自己的未来

## <<基于组件的应用程序设计>>

### 书籍目录

目录
译者序
致谢
引言
第一部分 技术概览
第一章 Windows DNA
1.1商业问题，结构化的解决方案
1.1.1应用程序体系结构
1.1.2组件
1.1.3三层模型启用器
1.2Windows DNA的体系结构
1.2.1组件对象模型
1.2.2 表示层
1.2.3商业层
1.2.4数据访问层
小结
第二章 组件对象模型
2.1为什么要采用COM？
2.2COM编程模型
2.2.1对象
2.2.2接口
2.2.3类
2.2.4组件
2.3分布式COM
2.3.1COM的安全性
2.3.2重新访问注册
2.3.3远程激活和调度
2.4Automation（自动化）
2.4.1Dispatch接口
2.4.2类型库
2.4.3双重接口
小结
第三章 数据访问基础
3.1为什么采用MDAC？
3.2访问数据
3.2.1ODBC
3.2.2万能存储
3.2.3万能数据访问
3.3OLEDB
3.4ActiveX数据对象
3.5远程数据服务
小结
第四章 Microsoft Transaction Server

## <<基于组件的应用程序设计>>

### 4.1为什么要使用MTS？

#### 4.2MTS的多种面孔

##### 4.2.1作为ORB的MTS

##### 4.2.2作为COM对象代理的MTS

##### 4.2.3作为TP - Monitor的MTS

#### 4.3应用程序服务器编程模型

##### 4.3.1易用性

##### 4.3.2通过资源共享实现伸缩性

##### 4.3.3规则1：调用 SetComplete

##### 4.3.4规则2：获取和独占接口指针

##### 4.3.5规则3：使用和释放资源

##### 4.3.6规则4：使用基于角色的安全性

##### 4.3.7规则5：使用事务

#### 小结

### 第五章 Active Server Pages

#### 5.1为什么要使用ASP

##### 5.1.1Automation组件

##### 5.1.2ASP编程模型

#### 5.2三层式应用程序体系结构

#### 5.3使用MTS进行集成

#### 小结

### 第六章 扩展MTS的功能

#### 6.1UNIX上的COM

#### 6.2XA的交互操作性

#### 6.3COM事务集成器

##### 6.3.1COMTI 体系结构

##### 6.3.2创建一个COMTI 自动化组件

#### 6.4Microsoft MessageQueueServer消息队列服务器

##### 6.4.1消息队列

##### 6.4.2使用MSMQ

#### 小结

### 第二部分 构造应用程序

### 第七章 应用程序的定义和模型化

#### 7.1创建一个应用程序定义

##### 7.1.1定义商业问题

##### 7.1.2定义功能规范

##### 7.1.3建立应用程序需求

#### 7.2应用程序的模型化

##### 7.2.1持久数据模型化

##### 7.2.2设计类

##### 7.2.3数据对象模型化

##### 7.2.4商业对象模型化

##### 7.2.5表示层模型化

#### 7.3定义物理结构

##### 7.3.1将类分组为组件

##### 7.3.2将组件分组为程序包和进程

## <<基于组件的应用程序设计>>

7.3.3将程序包和进程指定到机器

7.3.4建立物理结构文档

小结

第八章 构造数据对象

8.1从设计到实现

8.2为MTS实现组件

8.3数据对象技术

8.3.1利用AIDO编程

8.4实现数据对象

8.4.1在Visual Basic中实现db - CategoryC

8.4.2在Visual Basic中实现db - CustomerPasswordC

8.4.3快速考察db - CustomerC

8.4.4用VisualC++实现数据对象

小结

第九章 构造商业对象

9.1从设计到实现

9.2为MTS实现组件

9.2.1构造功能

9.2.2管理状态

9.2.3处理错误

9.2.4安全地访问您的组件

9.3构造商业对象

9.3.1在Visual Basic中完成实现 bus - PaymentC

9.3.2在C++中实现bus - InvoiceC

9.3.3利用SPM实现util - TakeANumber

小结

第十章 包装组件

10.1程序包

10.2设计程序包

10.2.1激活

10.2.2共享资源

10.2.3故障隔离

10.2.4 安全性隔离

10.2.5 “IslandHopper” 程序包设计

10.3构造程序包

10.3.1创建程序包

10.3.2向程序包中添加组件

10.3.3设置激活属性

10.3.4配置安全性

10.3.5设置事务属性

10.3.6 输出程序包

小结

第十一章 构造表示层

11.1从设计转到实现

11.1.1表示层应用程序的类型

11.1.2选择一种表示层体系结构

11.2实现表示层

## <<基于组件的应用程序设计>>

- 11.2.1使用商业对象
- 11.2.2使用RDS访问远程对象
- 11.2.3使用数据绑定
- 11.2.4使用ASP
- 11.2.5使用客户端组件
- 11.3实现“IslandHopper”应用程序的表示层
- 11.3.1设计决策
- 11.3.2实现Classifieds应用程序
- 11.3.3实现Internet客户

小结

### 第十二章 调试和排除故障

- 12.1组件级测试
- 12.2本地测试
- 12.2.1调试MTS组件
- 12.2.2追踪和确认
- 12.2.3将错误消息解码
- 12.2.4检查并排除数据库方面的问题
- 12.2.5用于排除故障的MTS工具
- 12.2.6Windows NT事件查看器
- 12.3测试分布式应用程序

小结

### 第十三章 性能验证

- 13.1为什么要进行性能验证？

#### 13.2性能验证的工作过程

- 13.2.1定义性能需求
- 13.2.2评估性能
- 13.2.3找到并消除瓶颈
- 13.3常见瓶颈
- 13.3.1SQLServer瓶颈
- 13.3.2数据访问瓶颈
- 13.3.3事务瓶颈
- 13.3.4其他瓶颈

小结

### 第十四章 部署分布式应用程序

- 14.1选择拓扑结构
- 14.1.1集成到公司现有的应用程序中
- 14.1.2满足性能需要
- 14.1.3常用部署配置
- 14.1.4容错功能
- 14.2部署过程
- 14.2.1安装预先建立好的程序包
- 14.2.2创建客户安装程序
- 14.2.3最终确定程序包设置
- 14.2.4安装客户程序

小结

### 第三部分 MTS 的未来

## <<基于组件的应用程序设计>>

### 第十五章 扩展应用程序

#### 15.1 利用COM+开发组件

##### 15.1.1 使用ComponentBuilder

##### 15.1.2 COM+支持的事务处理程序

#### 15.2 利用MSMQ开发应用程序

##### 15.2.1 MSMQ应用程序模型

##### 15.2.2 编制MSMQ应用程序

##### 15.2.3 配置MSMQ应用程序组件

#### 小结

### 第十六章 COM+

#### 16.1 发展中的COM+和MTS

##### 16.1.1 一种统一的体系结构

##### 16.1.2 一种统一的编程模型

##### 16.1.3 一种统一的管理模型

#### 16.2 针对企业应用程序的新服务

##### 16.2.1 对象池

##### 16.2.2 对象构造器

##### 16.2.3 补偿资源管理器

##### 16.2.4 安全性强化

##### 16.2.5 队列组件

##### 16.2.6 事件

##### 16.2.7 内存中的数据库

##### 16.2.8 动态负荷平衡

#### 小结

#### 附录 关于3 - 2 - 1程序

#### 参考书目

<<基于组件的应用程序设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>