<<Java消息服务>>

图书基本信息

书名：<<Java消息服务>>

13位ISBN编号：9787564119300

10位ISBN编号：7564119306

出版时间：2010-1

出版时间：东南大学出版社

作者：（美）布朗　等著

页数：305

前言

When I was presented with the opportunity to revise Java Message Service, I jumped at the chance. The first edition, published by O'Reilly in 2000, was a bestseller and without a doubt the definitive source for JMS and messaging in general at that time.Writing the second edition was an exciting chance to breath new life into an already great book and add new content that was relevant to how we use messaging today.What I failed to fully realize when I took on the project was just how much messaging(or, more precisely, how we use messaging) has changed in the past 10 years. New messaging techniques and technologies have been developed, including messagedriven beans (as part of the EJB specification), the Spring messaging framework, Event Driven Architecture, Service-Oriented Architecture, RESTful JMS interfaces, and the Enterprise Service Bus (ESB), to name a few. The somewhat minor book project that I originally planned quickly turned into a major book project. My original intent was to preserve as much of the original content as possible in this new edition. However, based on changes to the JMS specification since the first edition was written, as well as the development of new messaging techniques and technologies,the original content quickly shrank. As a result, you will find that roughly 75% of this second edition is new or revised content. The JMS specification was updated to version 1.1 a couple of years after the printing of the first edition of this book. While not a major change to the JMS specification, the JMS 1.1 specification was nevertheless a significant step toward fixing some of the deficiencies with the original JMS specification. One of the biggest changes in the specification was the joining of the queue and topic APl under a unified general APl,allowing queues and topics to share the same transactional unit of work. However, the specification change alone was not the only factor that warranted a second edition of the book. As the Java platform has matured, so has the way we think about messaging.From new messaging technologies and frameworks to complex integration and throughput requirements, messaging has changed the way we think about and design systems, particularly over the past 10 years. These factors, combined with the specification changes, are the reasons for the second edition.

<<Java消息服务>>

内容概要

《Java消息服务》第二版，是对标准API的一个彻底详尽的介绍——它支持消息传递，即网络计算机间的关键数据从软件到软件的交换。
你将学习到JMS如何帮助你解决许多架构难题，如整合不同的系统和应用程序，增加可扩展性，消除系统瓶颈，支持并行处理，提升灵活性和敏捷性。
由于更新了JMS 1.1，第二版还说明了厂商无关的规范将有助于你写基于messaging的应用程序，无论是使用IBM的MQ、Progress软件的SonicMQ、ActiveMQ的，还是其他专有的消息服务。
有了《Java消息服务》，你将可以： 使用点对点和出版与订阅消息传递模型建立应用程序 使用如事务和持久订阅功能，以使应用程序可靠 使用消息驱动的beans在Enterprise JavaBeans(EJB)中实现消息传递 使用JMS时结合RESTful应用程序和Spring应用程序框架 消息传递是一个强大的范例，可以更容易地使企业应用程序的不同部分分离。
《Java消息服务》第二版将迅速教会你如何运用其背后的关键技术。

作者简介

布朗(Mark Richards)，一位实践经验丰富的设计师，也是一位在消息传递、系统集成和面向服务的架构方面有领导地位的专家和作家。
Richard Monson-Haefel，O ’ Reilly 出版的《Enterprise JavaBeans》和《Java消息服务》第一版的合著者，是企业计算方面的世界级专家。
David A. Chappell，Oracle公司副总裁兼SOA首席技术专家，是《Java Web Services》和《Java消息服务》第一版（均为O ’ Reilly出版）的合著者。

<<Java消息服务>>

书籍目录

<<Java消息服务>>

章节摘录

插图：Event-Driven Architecture （EDA） is an architecture style that is built on the premisethat the orchestration of processes and events is dynamic and very complex, and there-fore not feasible to control or implement through a central orchestration component.When an action takes place in a system, that process sends an event to the entire systemstating that an action took place （an event）. That event may then kick off other pro-cesses, which in turn may kick off additional processes, all decoupled from each other.Some good examples of EDA include the insurance domain and the defined benefitsdomain. Both of these industry domains are driven by events that happen in the system.For example, something as simple as changing your address can affect many aspects ofthe insurance domain, including policies, quotes, and customer records. In this case,the driving event in the insurance application is an address change. However, it is notthe responsibility of the address change module to know everything that needs to happen as a result of that event. Therefore, the address change module sends an eventmessage letting the system know that an address has changed. The quoting system willpick up that event and adjust any outstanding quotes that may be present for thatcustomer. Simultaneously, the policy system will pick up the change address event andadjust the rates and policies for that customer.Another example of EDA is within the defined benefits domain. Getting married orchanging jobs triggers events in the system that qualify you for certain changes to yourhealth and retirement benefits. Many of these systems use EDA to avoid using a large,complex, and unmaintainable central processing engine to control all of the actionsassociated with a particular "qualifying event."Messaging is the foundation for systems based on an Event-Driven Architecture. Eventsare typically implemented as empty payload messages containing some informationabout the event in the header of the message, although some pass the application dataas part of the event. Not surprisingly, most architectures based on EDA leverage thepub/sub model as a means of broadcasting the events within a system.

媒体关注与评论

“这是对基本的Java集成技术的一个深度论述，涵盖了恰当程度的JMS API细节，并考虑周全地把开放源码解决方案并入需要了解的Enterprise Java的话题讨论。
”　　——Tim Berglund，August科技集团董事长

<<Java消息服务>>

编辑推荐

《Java消息服务(第2版·影印版)》由东南大学出版社出版。

版权说明