Java Message Service

Von David A Chappell, Richard Monson-Haefel
*DOC | *audiobook | ebooks | Download PDF | ePub*

Produktinformation -Verkaufsrang: #513107 in eBooksVerffentlicht am: 2000-12-04Erscheinungsdatum: 2009-02-09File Name: B0026OR3JY | File size: 72.Mb

Von David A Chappell, Richard Monson-Haefel : Java Message Service before purchasing it in order to gage whether or not it would be worth my time, and all praised Java Message Service:

KundenrezensionenHilfreichste Kundenrezensionen

KurzbeschreibungThis book is a thorough introduction to Java Message Service (JMS), the standard Java application program interface (API) from Sun Microsystems that supports the formal communication known as "messaging" between computers in a network. JMS provides a common interface to standard messaging protocols and to special messaging services in support of Java programs. The messages exchange crucial data between computers, rather than between users--information such as event notification and service requests. Messaging is often used to coordinate

programs in dissimilar systems or written in different programming languages.Using the JMS interface, a programmer can invoke the messaging services of IBM's MQSeries, Progress Software's SonicMQ, and other popular messaging product vendors. In addition, JMS supports messages that contain serialized Java objects and messages that contain Extensible Markup Language (XML) pages.Messaging is a powerful new paradigm that makes it easier to uncouple different parts of an enterprise application. Messaging clients work by sending messages to a message server, which is responsible for delivering the messages to their destination. Message delivery is asynchronous, meaning that the client can continue working without waiting for the message to be delivered. The contents of the message can be anything from a simple text string to a serialized Java object or an XML document.Java Message Service shows how to build applications using the point-to-point and publish-and-subscribe models; how to use features like transactions and durable subscriptions to make an application reliable; and how to use messaging within Enterprise JavaBeans. It also introduces a new EJB type, the MessageDrivenBean, that is part of EJB 2.0, and discusses integration of messaging into J2EE..deThe Java Message Service (JMS) provides a way for the components of a distributed application to talk asynchronously, or for welding together legacy enterprise systems. Think of it as application-to-application e-mail. Unlike COM, JMS uses one or more JMS servers to handle the messages on a store-and-forward basis, so that the loss of one or more components doesn't bring the whole distributed application to a halt. JMS consists of a set of messaging APIs that enable two types of messaging, publish-and-subscribe (one-to-many) and point-to-point (one-to-one). The highly lucid explanation of the ways in which these work makes the technical content a lot more approachable. In practice, however, Java Message Service is still a book for Java programmers who have some business programming experience. You need the background. After a simple JMS demonstration in which you create a chat application using both messaging types, the authors dissect JMS message structures, explore both types in detail, and then move on to real-world considerations. These include reliability, security, deployment, and a rundown of various JMS server providers. The appendices list and describe the JMS API, and provide message reference material. Considering the complexity and reach of the subject matter, Java Message Service does a great job of covering both theory and practice in a surprisingly efficient manner. It's easy to see why JMS has become so popular so quickly. Recommended. --Steve Patient, .co.uk.co.ukThe Java Message Service (JMS) provides a way for the components of a distributed application to talk asynchonously, or to weld together legacy enterprise systems. Think of it as application to application e-mail. Unlike COM, JMS uses one or more JMS servers to handle the messages on a store and forward basis so the loss of one or more components doesn't bring the whole distributed application to a halt.JMS consists of a set of messaging APIs which enable two types of messaging: publish and subscribe (one to many) and point to point (one to one). The authors' highly lucid explanation of the way these work makes the technical content a lot more approachable. In practice, though, Java Message Service is still a book for Java programmers with some business programming experience. You need the background. After a simple JMS demonstration in which you create a Chat application using both messaging types the authors dissect JMS message structures, explore both types in detail and then move on to real world considerations. These include reliability, security, deployment and a run-down on various JMS server providers. The appendices list and describe the JMS API and provide message reference material.Considering the complexity and reach of the subject matter, Java Message Service does a great job covering both theory and practice in a surprisingly efficient manner. It's easy to see why JMS has become so popular so quickly. Recommended. --Steve Patient