

*Creating Distributed Enterprise Applications
Java Message Service*

第2版
Updated for JMS 1.1



Java 消息服务

*Mark Richards, Richard Monson-Haefel
& David A. Chappell 著*
闫怀志 译

O'REILLY®



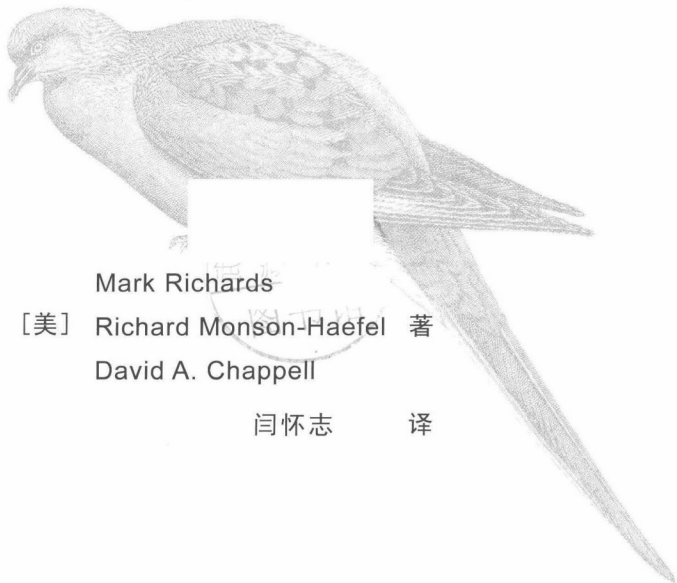
电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

O'REILLY®

Java 消息服务 (第2版)

Java Message Service, 2nd Edition



Mark Richards

[美] Richard Monson-Haefel 著

David A. Chappell

闫怀志 译

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书深入浅出地讲解了 JMS 1.1 规范的底层技术、Java 类和接口、编程模型及其不同实现等 Java 消息服务 (JMS) 和消息传送机制关键技术。通过对支持点对点和发布/订阅“消息传送”的标准 API 的完全解读及具体实例,介绍了如何利用“厂商无关”的 JMS 来解决许多体系结构面临的挑战。本书适用于掌握 Java 语言并有业务解决方案开发经验的读者,或者需要学习消息传送技术的读者。

978-0-596-52204-9 Java Message Service, Second Edition © 2009 by O'Reilly Media, Inc. Simplified Chinese edition, jointly published by O'Reilly Media, Inc. and Publishing House of Electronics Industry, 2009. Authorized translation of the English edition, 2009 O'Reilly Media, Inc., the owner of all rights to publish and sell the same. All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体版专有出版权由 O'Reilly Media, Inc. 授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字: 01-2009-3823

图书在版编目 (CIP) 数据

Java 消息服务: 第 2 版 / (美) 理查兹 (Richards, M.), (美) 蒙森-哈斐尔 (Monson-Haefel, R.), (美) 查普尔 (Chappell, D.A.) 著; 闫怀志译. —北京: 电子工业出版社, 2010.1

书名原文: Java Message Service, 2/e

ISBN 978-7-121-10050-5

I. J… II. ①理…②蒙…③查…④闫… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 224822 号

责任编辑: 周筠

封面设计: Karen Montgomery, 张健

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 21 字数: 400 千字

印 次: 2010 年 1 月第 1 次印刷

定 价: 59.80 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。服务热线: (010) 88258888。

O'Reilly Media, Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权电子工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在Unix、X、Internet和其他开放系统图书领域具有领导地位的出版公司，同时也是在线出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为20世纪最重要的50本书之一)到GNN(最早的Internet 门户和商业网站)，再到 WebSite(第一个桌面 PC 的Web服务器软件)，O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

译者序

获邀担纲《Java 消息服务（第2版）》的翻译，我的思绪一下回到了10年前。当时，不仅是初出茅庐的我，甚至很多资深的同行，都在为不同系统之间的信息交换而头疼不已：一方面，数据集成和系统整合的需求非常旺盛；另一方面，信息交换的技术实现不仅难度很大，而且实现方式也因各有利弊而难于选择。在技术路线的选择上，我们究竟何去何从？这在“雾里看花”的当年，真不啻于一场赌博。因为后来的事实证明，即便是其中较为常用的 CORBA、DCOM、RMI 等远程过程调用（RPC）中间件技术，虽可以解决一些问题，但它们在健壮性、性能和可伸缩性方面的表现很难让人满意。

期间，从我一位留美的同学处，偶然获悉了 Java 消息服务（JMS）机制，令我闻之一震——它很好地解决了让我困惑已久的在不同系统间进行数据和功能共享的问题。我不仅自己如获至宝，还因为职业的关系，在我的领地——三尺讲台上竭力传播这一思路。

简而言之，JMS 是 1999 年由 Sun Microsystems 领街开发的一种访问消息系统的方法，也就是供 Java 程序员使用的面向消息的中间件（MOM）。这种基于消息传送的异步处理模型，具有非阻塞的调用特性。发送者将消息发送给消息服务器，服务器会在合适的时候再将消息转发给接收者；发送和接收采用异步方式，这就意味着发送者无须等待，发送者和接收者的生命周期也无需相同，而且发送者还可以将消息传给多个接收者。如此一来，这种异步处理方式就大大提升了应用程序的健壮性、性能和可伸缩性，使数据集成和系统整合工作变得易如反掌，特别是在分布式应用上让同步处理方式望尘莫及。Java 消息服务作为一个与具体平台无关的 API，已经得到了绝大多数 MOM 提供商的支持。

本书深入浅出地讲解了 JMS 规范的底层技术、Java 类和接口、编程模型及其不同实现。尤其值得指出的是，当年我学习 Java 消息服务的启蒙教材正是 O'Reilly 在 2000 年出版的本书第 1 版。所以看到第 2 版，真是一见如故。较之 9 年前的第 1 版，除了著名的 Richard Monson-Haefel 和 David A. Chappell 之外，又有消息传送机制、面向服务体系结构和事务管理领域的公认权威 Mark Richards 来担纲第一作者。我在仔细研读之后发现，本版继承了初版的优点，但在内容及结构方面有了很大变化，某些地方甚至可以说是“质”变，究其原因主要有二：其一，初版两年后（即 2002 年 3 月）发布了 JMS 1.1，统一了消息域，新版必须为适用 JMS 1.1 而修订；其二，经过 10 年的发展，Java 平台和消息传送机制日臻完善，消息驱动 bean、Spring 消息框架、事件驱动体系结构（EDA）、面向服务体系结构（SOA）、RESTful JMS 接口，以及企业服务总线（ESB）等新的消息传送技术也层出不穷。这些亮点不仅是第 2 版的精华所在，它们本身也代表了 Java 消息服务技术的发展方向。

当然，任何一项技术都是有利有弊的，也都会经历发展、鼎盛直至消亡的生命旅程，Java 消息服务也不例外。毋庸讳言，近些年来在异构系统集成领域，Java 消息服务就遭到了 Web 服务的强力挑战。不过，在 Web 服务解决可靠性等问题之前，Java 消息服务仍然是异构系统集成的不二之选。我相信，如果您读完了本书，并切实领会了 JMS 的精髓和要义之后，会十分认同这个观点。

翻译一本书，弄不好就会是“出力不讨好的苦差”，因为无论是技术本身、原著水平、读者领悟力等哪个环节出了问题，板子通常都会打在翻译者身上。当然就本书而言，JMS 技术本身、原著的水平和读者的领悟力都不会有什么问题。所以，既然我愿意做这样的知识传播者，那就要勇于承担所有的责任。套用我所敬仰的一位长者之言：我只希望在我翻译完以后，全体读者能说一句，他是一个负责的译者，不是敷衍了事的译者，我就很满意了。如果他们再慷慨一点，说闫怀志某些地方译得还不错，我就谢天谢地了。

本书翻译期间得到了电子工业出版社博文视点公司徐定翔、白爱萍、杨绣国、陈元玉、许莹等编辑的热心帮助。书中绝大部分术语采用了 Sun 公司的标准译法，在此一并致谢。最后，我要感谢家人的支持，否则完成这样一项工作几乎是不可想象的。由于译者水平有限，加之时间较紧，虽已尽力避免错误，难免仍有疏漏，恳请广大读者将意见和建议发至：bityhz001@sina.com，不胜感激。

闫怀志

2009 年秋于北京中关村

推荐序

近十年来，我逐渐变成了基于消息传送系统的铁杆粉丝。这种系统所提供的可靠性、灵活性、扩展性及模块化程度，实为传统 RPC 或分布式对象系统所不及。使用基于消息传送的系统只须些许调整，因为它们的行为方式与体系结构师或设计师所期望的传统 n 层系统截然不同。这并不是说基于消息传送的系统更好抑或更差，它们仅仅是不同而已。这时候不再是直接调用对象的方法（这时对象能够保持会话状态或上下文），而是要求消息自身必须是自包含的而且是状态完备的。

这就提出了一个重要问题。

任何特定的开发人员使用任何特定的技术，都会有 4 个明显不同的阶段。

首先是门外汉（Ignorant）阶段。此时，我们只知道该技术存在与否，除此以外，对其能力一无所知。它充其量是经常和其他技术一起提及的一个字母组合，既可能和我们的日常工作十分相关，也可能无关紧要。

第二是探索者（Explorer）阶段。无论自愿与否，由于有某些东西激发了我们的好奇心和求知欲。我们开始迈出丛林探险的第一步，可能是下载一个实现，也可能是阅读几篇文章。开始了解这项技术在更大范围内的基本框架定位及其大致工作方式，不过，我们的实战经验通常仅限于诸如“Hello World”和为数不多的其他例子而已。

第三是熟手（Journeyman）阶段。在运行了一些例子和阅读过几篇文章后，我们对它有了些基本了解，并尝试使用它编写代码。感到用它生成代码并调试所犯的低级错误会非常轻松。无论如何，我们不是专家，但是至少能够写出个东西来编译，并且它在大多数时间可以运行。

最后是大师阶段。在构建一些系统并了解它们的实际运行情况后，我们会对该工具或技术有全面、深入的认识，甚至经常是没有运行代码也能够预知它会如何作用。我们会看到它如何与其他技术取长补短，并了解如何实现一些令人不可思议的效果，比如能够抵御网络

运行中断或机器失效等。回溯到 1999 年 Java 消息服务 (JMS) API 初次发布时,它并无任何非商业/开源实现可用,我还清晰地记得当时我边看边想:“不错,它看起来令人很感兴趣,但是没有真正实现,我就没法使用”,并把该规范印刷本放在一边供以后研读。几年之后,我进入 JMS 的探索者和熟手阶段,此时我开始领悟消息传送系统的功能,这一方面是因为有了一些实现,另一方面是因为我自己对其他消息传送系统(最多的是 MSMQ 和 Tibco)的探索,但是主要应归功于 Java 消息服务第 2 版的作者。

时至今日,我依然愧称大师。所幸您和我都知道谁不是大师。

Mark Richards 已经在消息传送领域花费了数年心血,他既是架构师和实现者,又是领袖和导师:这首先是因为他的顾问身份,其次是他作为 No Fluff Just Stuff (NFJS) 专题研讨会定期演讲者的地位(译注 1)。他对构建基于消息传送的系统的来龙去脉及其要义了然于胸,而且是他提议近乎完全重写 Richard Monson-Haefel 和 Dave Chappell 的第 1 版。即使您仍是 JMS 的门外汉,Mark 对消息传送基础知识、实现及设计利弊的完全攻略,会带您快速进入熟手阶段,并使您很快就可了解达到大师阶段所必需的知识结构。

而且,我的朋友是咨询本书的最佳人选。

妙哉,消息传送机制!

—Ted Neward

ThoughtWorks 首席顾问

2008 年 12 月 10 日,于比利时安特卫普(译注 2)

译注 1: No Fluff Just Stuff 是一个 Java 和开源技术会议,网址是 <http://www.nofluffjuststuff.com>。

译注 2: 安特卫普:比利时北部一港口城市,位于布鲁塞尔以北的斯海尔德河边。

获悉有机会修订《Java 消息服务》，我为此雀跃不已。2000 年，该书第 1 版由 O'Reilly 出版后非常畅销，而且在当时毫无疑问是 JMS 和消息传送的权威参考资料。编写第 2 版，这种机会非常令人兴奋，它赋予这样的名作以新的生命，并为其增加了当今使用消息传送机制的有关新内容。我接手这个项目时完全没有想到，过去 10 年中，消息传送机制（或者更确切地说，我们使用消息传送机制的方式）发生了这么巨大的变化。现在已经研发出若干种消息传送新技术，仅举几个为例：消息驱动 bean（作为 EJB 规范的一部分）、Spring 消息框架、事件驱动体系结构（Event-Driven Architecture, EDA）、面向服务体系结构（Service-Oriented Architecture, SOA）、RESTful JMS 接口，以及企业服务总线（ESB）等。我最初规划的这本小书很快变成了一个庞大的图书项目。

我最初设想在新版中尽量保留原来的内容。不过，根据第 1 版编写以来 JMS 规范的变化，以及消息传送新技术的研发情况，只好大量缩减原来的内容。结果您会发现，第 2 版中有 75% 左右是新内容，或者是修订的内容。

在本书第 1 版付印后两三年内，JMS 规范就升级到了 1.1 版。尽管新版本对 JMS 规范并没有做重大修改，它仍然在解决原 JMS 规范某些缺陷方面迈出了重要的一步。该规范最大的变化之一是在统一的公共 API 基础上增加了队列和主题 API，允许队列和主题共享同一事务性工作单元。不过，单单规范的发展并不是促成本书修订出第 2 版的唯一因素。由于 Java 平台已经成熟，相应地，消息传送的方式也已经成熟。从新的消息传送技术和框架，到复杂的集成和吞吐量需求，消息传送机制已经改变了我们考虑和设计系统的方式，特别是在过去的 10 年中更是如此。正是这些因素，加上 JMS 规范的发展，促成了本书第 2 版的问世。

除了第 2 章中的 Chat 应用程序以外，全书所有示例代码均已修改，以反映更新的消息传送用例和阐释第 1 版中未曾包括的一些 JMS 附加特性。

由于显而易见的原因，我在新版中增加了几章内容。您会发现第 1 章新增加了几小节，主要是关于 JMS API、更新的消息传送用例及讨论消息传送如何改变系统设计方式。您还会发现关于消息过滤、Java EE 和消息驱动 bean、Spring JMS、消息驱动 POJO 及消息传送设计的几章新内容。

除了增加新章之外，我还对现有各章做了大幅修订。由于更新了全书多个要点的示例代码，所以，我必须逐一大量重写对应的正文。这恰好为我提供了增加另外的小节和议题的机会，尤其是对于第 4 章的点对点消息传送模型和第 5 章的发布/订阅消息传送模型来说，更是如此。我还调整了第 1 版中这两章的顺序，因为我相信，使用队列的点对点消息传送模型更容易深入领会消息传送的概念，而不是使用主题和订阅者的发布/订阅消息传送模型。

我衷心希望本书新版会有助于您更加全面地理解 Java 消息服务和消息传送机制。

—Mark Richards

谁应该阅读本书

本书讲解并演示了 Java 消息服务的基础知识。它对于 JMS 规范的底层技术、Java 类和接口、编程模型及其不同实现的讲解，既直观、简洁，又不失严谨、扎实。

虽然本书着重于讲解基础知识，但它并不是一本“傻瓜式”书籍。尽管 JMS API 易于学习，但 API 抽象却是相当复杂的企业级技术。在阅读本书之前，您应该熟练掌握 Java 语言，并有过一些开发业务解决方案的实践经验。我们并不要求您具备消息传送系统方面的相关经验，但是必须掌握 Java 语言的应用知识。

组织结构

全书共分为 11 章和 4 个附录。第 1 章介绍了消息传送系统、消息传送用例、集中式和分布式体系结构及 JMS 的重要性所在。第 2 章到第 6 章深入探讨了使用点对点和发布/订阅这两种消息传送模型开发 JMS 客户端的细节，包括如何使用消息选择器来过滤消息。第 7 章和第 10 章应被视为“高级议题”，其范围涵盖了消息传送系统的部署和管理等部分。第 8 章是有关 JMS 的 Java 2 企业版（Java EE）概览，包括作为企业级 JavaBean（Enterprise

JavaBeans, EJB) 3.0 规范一部分的消息驱动 bean。第 9 章介绍了和消息传送有关的 Spring 框架 (Spring Framework)。最后, 第 11 章提供了对消息传送相关设计注意事项和反模式等诸多问题的深入思考。

第 1 章: 消息传送机制基础

定义了企业消息传送和消息提供者使用的通用体系结构。本章还定义和讲解了 JMS 及其两种编程模型: 发布/订阅模型和点对点模型。本章描述了消息传送机制的多个用例和实际场景, 同时还介绍了 JMS API 的基础知识。

第 2 章: 编写一个简单的示例程序

带领读者一起开发一个简单的发布/订阅 JMS 客户端。

第 3 章: 深入剖析一条 JMS 消息

详细分析了 JMS API 最为重要的部分——JMS 消息。

第 4 章: 点对点消息传送模型

通过开发一个简单的借方和贷方 JMS 应用程序, 深入分析了点对点消息传送模型。本章还介绍了点对点消息传送模型的一些核心亮点, 包括消息关联、动态队列、负载均衡及队列浏览等。

第 5 章: 发布/订阅消息传送模型

通过强化第 4 章中开发的借方和贷方应用程序, 深入分析了发布/订阅消息传送模型。本章还包括持久订阅者、非持久订阅者、动态持久订阅者及临时主题等。

第 6 章: 消息过滤

详细讨论了如何使用消息选择器过滤消息。

第 7 章: 保证消息传送和事务

全面深入地讲解了有关高级议题, 包括保证消息传送、事务、确认、消息编组及失效等。

第 8 章: Java EE 和消息驱动 bean

提供了有关 JMS 的 Java 2 企业版 (Java EE) 3.0 版本概览, 而且还包括了消息驱动 bean (MDB)。

第 9 章: Spring 和 JMS

详细讲解了提供有关 JMS 的 Spring 框架 (Spring Framework), 包括 Spring JMS 模板 (Spring JMS Template) 和消息驱动 POJO (MDB)。

第 10 章: 部署注意事项

系统深入地分析了选择提供者和部署 JMS 应用程序时应该注意的若干特点和问题。

第 11 章：消息传送设计注意事项

提供了对若干设计注意事项的深入思考和解释说明，包括内部目的地和外部目的地的使用、请求/应答处理，另外还讨论了某些更常见的消息传送反模式。

附录 A：Java 消息服务 API

提供 JMS 包定义的类和接口的快速参考。

附录 B：消息头

提供有关消息头的具体信息。

附录 C：消息属性

提供有关消息属性的具体信息。

附录 D：安装和配置 ActiveMQ

提供运行本书示例所需的 ActiveMQ 安装和配置的具体信息。

软件和版本

本书内容涵盖了 Java 消息服务 1.1 版（Java Message Service version 1.1，JMS 1.1）。它使用来自 Java 6 平台的 Java 语言特性。由于本书重点是开发“厂商无关”的 JMS 客户端和应用程序，因此不会讨论依赖于某个厂商的私有扩展和惯用法。无论使用哪种 JMS 兼容提供者，都可以阅读本书，不过，您应该非常熟悉该提供者的特定安装、部署和运行时管理过程，只有这样才能更好地运行本书的示例。如果要查找某个特定 JMS 提供者 JMS 客户端的安装和运行细节，请查阅该 JMS 提供者的文档；JMS 规范中并未包括这些细节。本书附录 D 提供了使用流行的开源 JMS 提供者 ActiveMQ 运行示例的具体细节。

第 8 章的示例源代码与解释参考了企业级 Java-Beans 3.0（EJB 3）规范。第 9 章的示例源代码与解释参考了 Spring 框架 2.5 版。

要获得本书开发的示例，可以访问 <http://oreilly.com/catalog/9780596522049/examples>。这些例子是按章组织的。另外还提供了为特定厂商而修改的专用源代码。这些特定厂商的示例包括一个 readme.txt 文件，它指向用于下载和安装 JMS 提供者的文档，以及为每个示例安装相应提供者的特定使用说明。

本书使用的约定

本书使用如下排版约定：

等宽字体 (Constant Width)

用于示例代码和代码段、类、变量、方法名称、文本内使用的 Java 关键字、SQL 命令、表名、列名及 XML 元素和标记。

等宽黑体 (Constant width bold)

在某些代码示例中用于强调。



本图标代表提示、建议或一般性注释。

术语 **JMS 提供者** 是指实现了 JMS API 以提供到其企业消息传送服务连接的厂商。术语 **JMS 客户端** 是指使用 JMS API 和 JMS 提供者发送和接收消息的 Java 组件或应用程序。JMS 应用程序是指为提供一个软件解决方案而一起工作的任何 JMS 客户端组合。

使用示例代码

本书的目的是帮助您更好地完成工作。一般而言，您可以在自己的程序和文档中使用书中代码。除非您原封不动地大量引用代码，否则您不需要征得我们的许可。例如，编写程序时引用本书中的若干代码片段，这并不需要获得许可，而销售或发布 O'Reilly 图书的示例光盘则需要许可。通过引用本书内容及示例代码来解答问题并不需要许可。将本书中的大量示例代码加入到您的产品文档之中则需要得到许可。

如果您在引用时注明出处，我们将不胜感激，但我们并不强求。标注引用通常应包括书名、作者、出版商及 ISBN（国际标准书号）。例如：“Java Message Service, Second Edition, by Mark Richards, Richard Monson-Haefel, and David A. Chappell. Copyright 2009 Mark Richards, 978-0-596-52204-9。”

如果您觉得示例代码的使用超出了合理使用或上述许可范围，请随时通过 permissions@oreilly.com 和我们取得联系。

如何联系我们

我们已尽力核验本书所提供的信息，尽管如此，仍不能保证本书完全没有瑕疵，而网络世界的变化之快，也使得本书永不过时的保证成为不可能。如果读者发现本书内容上的错误，不管是错字、错字、语意不清，甚至是技术错误，我们都竭诚虚心接受读者指教。如果您有任何问题，请按照以下的联系方式与我们联系。

奥莱理软件（北京）有限公司
北京市 西城区 西直门 南大街2号 成铭大厦C座807室
邮政编码：100080
网页：<http://www.oreilly.com.cn>
E-mail：info@mail.oreilly.com.cn

O'Reilly & Associates, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international/local)
(707) 829-0104 (fax)

与本书有关的在线信息如下所示。

<http://www.oreilly.com/catalog/978059622049> (原书)
<http://www.oreilly.com.cn/book.php?bn=978-7-121-10050-5> (中文版)

北京博文视点资讯有限公司（武汉分部）

湖北省 武汉市 洪山区 吴家湾 邮科院路特1号 湖北信息产业科技大厦1402室
邮政编码：430074
电话：(027)87690813 传真：(027)87690813转817
读者服务网页：<http://bv.csdn.net>
E-mail:
reader@broadview.com.cn（读者信箱）
bvtougao@gmail.com（投稿信箱）

致谢

这些致谢来自 Mark Richards，而且是指本书第2版。

写出一本书绝非一人之功；恰恰相反，它是许多人一起不辞辛劳、努力工作的最终结果。我要感谢许多人在此项目期间的辛勤工作及他们的大力支持。

首先,我要赞赏和感谢 Julie Steele 编辑在本项目期间对我的宽容,以及为本书付印所做的编辑、协调和所有这一切不平凡的工作。我还要感谢 Richard Monson-Haefel (和 David Chappell 一起)所做的编写本书第 1 版这样伟大的工作,以及为我提供编写第 2 版的机会。

我要感谢好友和同事 Ted Neward 在繁忙的旅途中为本书作序,并在此项目期间一直为我领路指航。他的建议和指导使得本书新版得以完成。我还要感谢以下朋友: Neal Ford、Scott Davis、Venkat Subramaniam、Brian Sletten、David Bock、Nate Shutta、Stuart Halloway、Jeff Brown、Ken Sipe,以及所有的 No Fluff Just Stuff (NFJS) 伙伴,感谢他们在 NFJS 会议内外持之以恒的支持、踊跃的讨论,以及真挚的情谊。他们是最伟大的。

我还要感谢帮助确保资料技术准确性的许多专业技术评论者,他们是:超级软件工程师和技术专家 Ben Messer、August Technology Group, LLC 首席软件开发者和所有者 Tim Berglund、Collaborative Consulting, LLC 首席技术体系结构师 Christian Kenyeres,最后还有(但绝非最不重要) Ken Yu 和 Igor Polevoy。我知道在假期编辑和审查原稿并不容易(对我来说恐怕是很糟糕的时间安排),但是他们的实战经验、建议、注释、意见及技术编辑,使得本书日臻完美。

感谢 Scotland Macallan Distillery 的亲属们,他们做出了世界上口味最淡的苏格兰威士忌。它减轻了我写作时那些漫漫长夜的痛苦,特别是隆冬季节。

最后,我要感谢可爱的妻子 Rebecca 对我写书持之以恒的支持。Rebecca,你对我来说意味着整个世界,直至永远。

第 1 版致谢

这些致谢摘录自本书第 1 版,来自原作者 Richard Monson-Haefel 和 David A.Chappell。

虽然封面仅有两位作者署名,但是,本书成书和面世的荣耀应由多人共享。Michael Loukides 编辑对本书的成功至关重要。没有他的丰富经验、专业技能和悉心指导,写出本书将几无可能。

许多专业技术评论者的帮助,确保了资料的技术准确性,并且真正阐释了 Java 消息服务的精髓。在此要特别鸣谢: Sun Microsystems 公司的 Joseph Fialli、Anne Thomas Manes 和 Chris Kasso, Progress 的 Andrew Neumann 和 Giovanni Boschi, Softwired 的 Thomas Haas, International Systems Group 的 Mikhail Rizkin, 以及 ExoLab 的 Jim Alateras 等。这些技术专家的贡献对保证本书技术和概念的正确性至关重要。他们实现了行业标准和实战经验二者的融合,并使得本书成为迄今出版的 JMS 图书当中最好的一本。

我还要感谢 Sun Microsystems 公司 Java 2 企业版的首席体系结构师 Mark Hapner, 他为我们解答了最为棘手的几个问题。感谢 Sun Microsystems 公司主办的 JMS-INTEREST 邮件

列表的所有参与者，感谢他们热情的、内涵丰富的邮件。

我要特别感谢 SonicMQ 技术写作团队的 George St.Maurice 参与组织 O'Reilly 网站的示例。最后，我们要把最真挚的谢意献给我们的家庭。Richard Monson-Haefel 要感谢妻子 Hollie 对他接连出书的一贯支持和帮助。她的爱使得一切皆有可能。David Chappell 要感谢妻子 Wendy 和孩子 Dave、Amy 和 Chris 在他奋力写作期间对他的宽容。

David Chappell 还要感谢 Progress SonicMQ 团队的部分成员：Bill Wood、Andy Neumann、Giovanni Boschi、Christine Semeniuk、David Grigglesstone、Bill Cullen、Perry Yin、Kathy Guo、Mitchell Horowitz、Greg O'Connor、Mike Theroux、Ron Rudis、Charlie Nuzzolo、Jeanne Abmayr、Oriana Merlo 和 George St.Maurice，感谢他们帮助确定合适的议题，并确保对这些议题的正确阐释。另外，David Chappell 还要特别感谢 George Chappell 在“分裂不定式”方面的帮助。

目录

CONTENTS

推荐序.....	I
前言.....	III
第1章 消息传送机制基础.....	1
1.1 消息传送机制的优点.....	3
1.1.1 异构集成.....	3
1.1.2 缓解系统瓶颈.....	3
1.1.3 提高可伸缩性.....	4
1.1.4 提高最终用户生产率.....	4
1.1.5 体系结构灵活性和敏捷性.....	5
1.2 企业消息传送.....	5
1.2.1 集中式体系结构.....	7
1.2.2 分散式体系结构.....	7
1.2.3 混合体系结构.....	8
1.2.4 以集中式体系结构作为模型.....	8
1.3 消息传送模型.....	9
1.3.1 点对点模型.....	10
1.3.2 发布/订阅模型.....	10
1.4 JMS API.....	11
1.4.1 点对点 API.....	13
1.4.2 发布/订阅 API.....	14
1.5 实际场景.....	14
1.5.1 面向服务体系结构.....	15
1.5.2 事件驱动体系结构.....	16
1.5.3 异构平台集成.....	16
1.5.4 企业应用集成.....	17
1.5.5 企业到企业.....	17
1.5.6 地理分散.....	18
1.5.7 信息广播.....	18
1.5.8 构建动态系统.....	18
1.6 RPC 和异步消息传送.....	21