

Enterprise JavaBeans 3.0
Developing Enterprise Java Components

第5版
涵盖 Java Persistence

Enterprise JavaBeans 3.0

中文版



Bill Burke &
Richard Monson-Haefel 著
莫映 徐昊 胡凯 译

O'REILLY®



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

O'REILLY

Enterprise JavaBeans 3.0

Enterprise JavaBeans 3.0

第5版 中文版

FIFTH EDITION

[美] Bill Burke 著
Richard Monson-Haefel
莫映 徐昊 胡凯 译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书主要介绍 EJB3.0 标准, 是获奖权威参考书 Enterprise JavaBeans 的最新第 5 版。全书内容主要分为两大部分, 第一部分是 EJB 3.0 的标准技术手册, 主要介绍 EJB 概念、EJB 工作原理及使用方法; 第二部分是 JBoss 实践指南, 详细阐述了 JBoss 的安装、配置、运行、JBoss Application Server 等相关内容。全书实例丰富, 讲解细致, 深入浅出, 适合 Java 平台开发人员, 特别是企业级 Java 开发人员阅读参考。

0-596-00978-x Enterprise JavaBeans 3.0, Fifth Edition. Copyright © 2006 by O'Reilly Media, Inc. Simplified Chinese edition, jointly published by O'Reilly Media Inc. and Publishing House of Electronics Industry, 2007. Authorized translation of the English edition, 2006 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: 01-2007-0567

图书在版编目 (CIP) 数据

Enterprise JavaBeans 3.0: 第 5 版 / (美) 布克 (Burke, B.), (美) 蒙森-哈斐尔 (Monson-Haefel, R.) 著; 莫映, 徐昊, 胡凯译. —北京: 电子工业出版社, 2007.5

书名原文: Enterprise JavaBeans 3.0, 5th Edition

ISBN 978-7-121-04001-6

I. E... II. ①布...②蒙...③莫...④徐...⑤胡... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 033297 号

责任编辑: 周筠 晓菲

项目编辑: 梁晶

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市金马印装有限公司

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: 48.25 字数: 1112 千字

印 次: 2007 年 5 月第 1 次印刷

定 价: 98.00 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 zltz@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 and 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。

For my girls: Monica, Molly, Abby, and Winnie

—Bill Burke

For my wife and best friend,

Hollie



—Richard Monson-Haefel

在过去十年中，整个计算机世界的面貌发生了深刻的变化。一方面，图形界面的出现解放了用户在人机界面上所花费的巨大成本，模块化、一致化的图形界面使任何应用程序的操作方式都变得统一，用户可以更好地把注意力集中到软件本身的理解上。另一方面，快速程序开发技术使得开发具有标准化外观的应用程序变得极其方便。因此桌面应用迎来了它的黄金时代。

但同时，在最终客户的视野之外，企业应用的演化体现出了完全不同的复杂性。企业运算所需的特性包括：牢不可破的数据完整性、永续运行的高可靠性、应对业务能力扩展的高伸缩性，让任何简单的事情都变得更加复杂。此外，Internet 的迅速普及使得应用可以全球廉价接入，B/S 成了企业应用的必然选择，此时企业对平台无关性、数据库独立性的追求，使得开发所需关注的细节也更多。

如果说在桌面开发领域中，Delphi 是一艘导弹快艇，在发现目标后可以实施快速突击，那么，现在企业应用所需的就已经是一支完整的航母舰队了：你必须拥有全天候跨越 Windows 和 Unix 大洋的远洋能力，完善可靠的舰艇间的 C3I 通讯系统。为减少研发工作量和保障成本，还必须统一弹药规格，这样才可连接到名为 Oracle 和 DB2 的不同弹药供应船上去，从而可以支持数量庞大的舰载机编队，以对付高强度的并发业务挑战。因为规模庞大，你作为管理一支航母舰队的司令官，要操心的事情，远远超过只需在近海进行近程攻击的导弹快艇的艇长，而客户还要求你的舰队具有和快艇一样快的部署和战略调整速度，甚至可以随意替换舰队中船只的供应商。

经过过去持续数年的努力，Java 和 J2EE 技术终于可以构筑这样一支航母舰队了。Java 提供了“一次编写、到处运行”的便利底层，而 EJB2、JMS 的分布式能力构建了 C3I 系统，成了集群中的通讯枢纽，JDBC 则提供了最基本的弹药标准化指标。但不幸的是，战斗打响后，总装备部在设计室中进行的设计和实际战场的状况存在着差异，从战地指挥官到第一线的士兵都对这一套复杂难用的 C3I 系统提出了严厉的批评，他们抱怨整套系统给从后勤到飞行员的各个岗位都带来了严重的工作压力，而整体效果却乏善可陈，准备缓慢，攻击困难。在经过若干年的实战检验和痛苦抉择之后，整个战争机器终于进行了平台升级，即 EJB3.0。

EJB3.0 的设计组深刻地认识到，是战斗主体本身对整个战役发挥了至关重要的作用，而这就意味着要让 C3I 系统变得更加好用，更加隐形。因此作为系统组件的各舰艇和战斗机不再需要主动依赖于 EJB 容器，而变成 EJB 容器为这些主体 (Entity Bean) 提供服务。依靠

升级后的 Java 5 平台带来的 Annotations 注解语法和更多的默认配置, EJB 现在变得更像是在幕后运作。这显著减少了不必要的声明和准备, 各个战斗主体 Entity Bean 可以把主要精力放在战术也就是业务本身上, 战斗意图贯彻得更快, 战术转换可以更加频繁, 以应对更加复杂的战场形势。

与此同时, 独立出来的 Persistence API 让后勤体系达到了前所未有的现代化高度。通过采纳经过大量实战检验的 Hibernate 思想, 整个持久层有了一个完整的 ORM 体系。对类继承体系如何映射到数据库有了明确的规定, 这样, 程序员可以充分地利用面向对象技术所带来的灵活性、可扩展性和可重用性的强大动力。

EJB3.0 显著减轻了开发团队的压力, 让开发人员从繁杂的配置中解放了出来, 从而把精力集中在那些有意义的业务中去。对于项目经理来说, 减少了开发周期, 减少了开发成本, 就意味着对业务掌控的能力大为增强。

现在, 让我们回到这本书上来。

我们都知道, 由于 EJB2 对生产力的桎梏, 产生了众多的英雄人物。JBoss 成员 Gavin King 和 Bill Burke 都是最耀眼的明星, 他们在实际工作中积累了真正有价值的经验, 并且通过努力把这些经验形成了完整的产品, 而且, 最终在 EJB 3.0 这一正式规范形成的道路上起到了关键作用。众所周知, Persistence API 几乎完全采纳了 Hibernate 的 ORM 实现思路, 而 Bill Burke 则是 EJB3 容器标准的灵魂之一。

如果说在此之前, 还有很多公司对是否采用 Hibernate 这样的 ORM 技术犹豫不决, 那么现在, EJB3 给了 ORM 一个强力的背书。通过 javax.persistence 这样的包前缀, ORM 成为了技术主流。

在这样的大背景下, 由 JBoss 首席架构师, JBoss AOP 项目的领导者, JBoss EJB3.0 的领导者之一, EJB 3.0 专家组成员 Bill Burke 亲手操刀的《Enterprise JavaBeans 3.0 (第 5 版)》, 是理解 EJB 3.0 和 Persistence API 1.0 技术当之无愧的首选读物。愿读者通过他细致入微的分析和讲解, 得以有机会领会新一代 Enterprise JavaBeans 的完整体系结构和实现精髓。

本书的翻译水准当称一流。译者长期工作于企业应用开发的一线, 对重型和轻型框架都有深入的理解。他们的工作尽心尽力, 翻译准确流畅, 这是本中文版图书的质量保证。

战争还在继续, 而本书就是新的条令手册。进攻的命令已经下达, 舰队在夜幕中已经倾巢而出。现在, 战争的结果就掌握在读者您的手上©

满江红开放技术研究组织 创始人
曹晓钢

www.redsaga.com
2007 年 3 月于北京

译者序

过去几年间，J2EE 及 EJB 技术始终被认为是企业应用领域的主力军。EJB 从一开始的备受追捧，及至后来的饱受诟病，一直都是人们议论的焦点。而如今，像 Spring、Hibernate 这样的开源软件大获成功，轻量级思潮与敏捷方法风行于市，在企业应用这一主战场上，曾为人们所熟识的 J2EE 技术将会何去何从？想必这是多数 Java 从业者们最为关切的事情。而作为这一焦点的最新成果——EJB 3.0 规范，自 2003 年 5 月始，历经整整 3 年的酝酿以后，其最终的发布版本终于在 2006 年 5 月面世了。

EJB 3.0 可谓 EJB 技术有史以来最大的一次变革，它所带来的，不仅是技术上的变革，更是思想上的。它将主旨定位于“化繁为简”：从开发者的角度出发，通过降低复杂度，着力改善系统的架构（援引自规范）。新规范从成功开源项目中汲取养分，给我们带来了许多有益的实践：藉由 Persistence API，entity bean 可以是任意的普通 Java 对象，而无需再继续继承既定的接口；大量利用 Java 注解，再辅以少量的 XML 部署描述文件，使部署工作大为轻松，且灵活自如；还有轻量级的对象/关系映射机制，外部资源的依赖注入手段，以及注解的“Configuration By Exception”思想，凡此种种，无不令开发人员获益匪浅。

本书是讲解 EJB 技术的经典书籍。英文版自 1999 年问世起，先后经历了 5 个版本，期间曾经获得过 Java Pro 最受读者欢迎大奖，Amazon.com 的“2000 年最佳图书”大奖，以及 *Java Developer Journal* 的最受欢迎大奖。而作为本书的最新一版——《Enterprise JavaBeans 3.0（第 5 版）》，则全面介绍了 EJB 3.0 技术的方方面面。

本书的原作者 Richard Monson-Haefel 是 EJB 领域的资深专家。2004 年 7 月间，他离开 Java 社区，作为高级分析师加入了 Burton Group。其时，一则贴于 TSS 上的消息从者甚多，记得还曾有人嗟叹过：“Gads... this sounds like an obit!”，足见其个人影响力以及该书在读者心中的地位。关于这一点，还可从 Amazon 上读者对该书累计四颗半星的评价得到佐证。

Richard 在挂笔之后，将该书写作的接力棒交给了现任 JBoss 首席架构师 Bill Burke。Bill 曾为本书的第 4 版撰写过《JBoss 操作手册》，因此接手此书当是驾轻就熟的。事实证明 Bill Burke 不辱使命，经典得以再次延续。有趣的是，英文第 5 版几乎是和 EJB 3.0 规范同时问世的，不知是否是 Bill 的有意安排©

很荣幸能有机会将这本经典作品译成中文呈现给诸位。也许是机缘巧合，作为译者之一的我，因为项目应用的需要，曾于 2001 年有幸拜读过本书的第 2 版。当时的 EJB 正值 1.1 版，作为一项新兴技术，国内在这方面的资源还不多见，实际的项目运用则更是凤毛麟角。

本书是引领我步入 EJB 殿堂的启蒙之作，因而给我留下了深刻的印象。

时光荏苒，转眼间到了 2006 年夏，当博文视点的编辑联系到我，并告知该书第 5 版中译本的出版计划时，我的欣喜之情溢于言表。一系列简短而必要的手续过后，样书很快就送到了。当我手捧着厚重的 700 页浩卷时，恰似故友重逢一般。

翻译的过程异常辛苦。我们投入了无数个深夜与几乎全部的节假日，但即便如此，原定三个月的翻译计划依然拖延了近半年的时间。如果说，2007 年初送出最后一份译稿时的感觉是如释重负，那么随着本书出版的日渐临近，作为译者的我们，则感到如芒在背。由于是多人翻译，因此统稿工作尤为繁冗。为了尽量保证翻译质量，在交予编辑进行后期校审之前，每章译文都经过了至少两轮的初校，想必这也是进度拖延的重要原因。作为补充，中文版随附了翻译期间译者所用的中英文术语对照表，其中也涉及了这一译本中，相关翻译习惯的若干说明。希望本书中文版能够得到诸位读者的认可。

这本译作的完成是大家共同努力的结果，这包括了参与本书翻译、审校，以及提供支持和关注的所有人。感谢 Bill Burke 先生在本书翻译期间为我们答疑解惑。感谢刘铁锋，没有他的引荐，我们将与这本经典之作失之交臂。感谢博文视点的周筠老师对我们的信任，能够容忍本书翻译进度的一再拖延。感谢本书的两位责编，方舟与晓菲，前者是多年以前活跃于 CSDN 论坛的故交，而后者则为本书的后期审校与编加工作付出了辛劳，我们的合作非常愉快。此外，还要感谢三位初校者认真细致的工作，他们是李唯一、张辰雪和王开福。还要感谢曹晓钢老师，为本书撰写推荐序，并在本书翻译期间为我们提供了 Wiki 在线交流平台。

由于译者水平所限，译文难免有错误之处，欢迎读者批评指正。

莫映

2007 年 3 月于北京

中英文术语对照表

说明:

* 中译本对书中出现的所有与 EJB 技术相关的专有软件框架、系统、平台以及技术,都尽量保留了原文。在一些新生词汇还未曾见到一致认可的译法之前,想必这是较为妥帖的选择,而且同时还能保证阅读的流畅感。比如:Java Persistence, EntityManager 等。

* 以往的一些 EJB 技术名词,如果已有常见译法,则依然保留原文,但会在首次出现之处附上常见的中文翻译,比如:session bean (会话 bean)。

* 关于 entity 译法的考量。在本书上下文中,单独出现的 entity 通常都代表着 entity bean 及其相关概念,只是作者做了省略。此时,作为专用术语,在翻译时选择了保留原文;若是作为一般性概念出现于文中,则译为实体。

* 关于 column 与 field 译法的考量。按照习惯,此二者均可译为“字段”。不过为了不至混淆,中译本做了统一区分,将 column 译作“字段”,而 field 则参考了侯捷先生的翻译,译作“数据成员”。

* 关于 attribute 与 property 译法的考量。按照习惯,此二者均可译为“属性”。但在本书中,attribute 多指 XML 元素中的属性,而 property 则指代 bean 中的 getter/setter。为了不至混淆,中译本做了统一区分,将 attribute 译作“属性”,而 property 则译作“成员属性”,与 field 互为呼应。

* 关于 modeling 译法的考量。Modeling 一般可以译作“为……进行建模”,不过在中译本中,我们选择了一种更为简练,但又不致遗漏信息的译法——模型,这一译法亦参考了侯捷先生的翻译。

英 文	中 文	说 明
abstract schema name	抽象结构名称	
annotation	注解	
attribute	属性	
build	生成, 构建	
column	(数据库的) 字段	
constructor	构造函数	
delegate	委托, 委派	
deployment descriptor	部署描述文件	
detached	游离	

英 文	中 文	说 明
distributed object	分布（式）对象	
eagerly fetch	立即获取	
EJB object	EJB object, 或 EJB 对象	
element	(XML 的) 元素	
endpoint interface	端点接口	
entity listener	entity 监听器	
field	数据成员	即通常所指的类中的成员变量
heuristic decision	试探性决策	
local interface	本地接口	
managed	托管, 受管	
message destination	消息目标地址	
method	成员函数	
modeling	模型	即为……进行建模
named	具名的	
operator	操作符/运算符	
partial XML deployment	部署描述片段	
phantom read	幻读	
property	成员属性	尤指 bean class 中的 getter/setter
proxy	代理	
relationship	关联, 关系, 关联关系	
remote interface	远程接口	
response time	响应时间	
standalone application	独立桌面应用程序	
stub	存根	
transaction-aware	事务感知的	
transaction demarcation	事务划分	
transient property	非持久型成员属性	transient 在单独使用时也译为“瞬时的”
undeploy	卸载	
unqualified class name	非限定类名	
wrapper type	封装类（型）	即 Java 中对应于基本类型的封装类

Enterprise JavaBeans 是 Java Enterprise Edition 平台的核心组件技术。作为一种企业基础架构，它为开发者提供了针对企业应用中多数必备服务的自动化管理功能。而 EJB 容器——作为 enterprise bean 组件直接面对的环境，为组件提供了众多托管服务——正是这一架构的核心。

然而，在以往版本的 EJB 中，为了使用托管环境，开发者编写的 API 不得不更多的着力于满足 EJB 容器的需求，而非企业应用的业务逻辑。这就导致了 EJB 开发被不必要地复杂化了。例如：

- 由于要实现各种不同的 EJB 接口，因而导致了大量的样板代码，这些代码用于实现接口所要求的方法，但对应用系统而言却是多余的。
- 为了使应用系统与其所在的环境及容器服务融为一体，XML 部署描述文件必不可少。而对组件环境的访问又异常繁芜，且不够直观。
- 受容器托管的持久化使得针对领域对象的建模被不必要地复杂化和重量化了。提供容器托管持久化的初衷是出于易用性的考虑，然而在实践中，它却异常难用，并有诸多局限。

发布 EJB 3.0 的目的，就是要让 EJB 重新着力于简化开发者的工作，解决上述所有的问题，以及更多其他的问题。

这一过程中的首要工作很多，其中之一就是要对导致以往版本 EJB 过于复杂的原因进行评估。这包括：

- 分析人们对 EJB 的众多批评；
- 理解 EJB 的哪些设计模式实际上是反模式；
- 找出那些难以使用，对初学者而言不够直观，或者可以被完全弃用的 API；
- 寻找这项技术中对易用性构成阻碍的其他方面。

一份待解决问题的初步清单，加上一项阐明如何开展工作的提案，便构成了 JSR 220，亦即 JCP 规范请求（Java Community Process specification request）的基础。藉此，我发起了 EJB 3.0 规范的制订工作。从待改进 API 的最初清单入手，为了设计出更加完善，也更为简单的架构，EJB 3.0 的专家组开始了头脑风暴的过程。Bill Burke，JBoss 应用服务器的首席架构师（本书的作者）正是这项工作的主要参与者之一。

专家组的工作使 EJB 得到了极大的简化和改善。与之前的版本相比，所有 EJB 的关键 API 都变得更加易于使用了，而且，只对例外情况进行配置（configuration-by-exception）的方式使开发者得以依赖于预期的默认行为。除非解决一些高阶问题，XML 部署描述文件已经变得可有可无了。

新近被加入 Java SE 的 Java 语言元数据机制，为我们达成上述简化目的提供了帮助。借助元数据注解，EJB 3.0 在 Java 代码中表达组件对容器的依赖关系，这可以免去部署描述文件的编写工作。而且，EJB 3.0 为元数据提供了默认取值，如此一来，元数据在通常情况下就可以用得很少。

通过使用元数据注解来指定环境依赖及生命周期回调，EJB3.0 也已经能够使 bean class 不必非得实现 EnterpriseBean 的接口了。现在，一个 bean class 可以有选择地指定所需实现的接口，并且可以只实现必要的方法，而不是充斥着不必要的样板代码。

我们可以在组件查询或注入期间创建指向组件及其实例的引用，通过要求容器具备更为巧妙的注入机制（对应用系统而言是透明的），我们还可以省去以前的 EJBHome 工厂模式。现在的 session bean 可以是具备普通业务接口的普通 Java 类，而不必是重量级的组件。

对开发者而言，EJB 的诸多变化极大地简化了开发工作。而且，这些改变对 EJB 的底层架构并没有造成根本性影响，这使旧有系统向 EJB 3.0 的迁移成为可能。

对容器托管持久化的简化给我们带来了更大的挑战。与简化 session bean 和 message-driven bean 的步骤类似，我们从减少不必要的接口，利用注解提供配置信息等工作开始入手。不过，我们很快就意识到，EJB 的持久化需要更为激进的变革。而诸如 Hibernate 这样的轻量级对象/关系映射技术的成功，为这一变革的方向给予了清晰的指引。

正如本书中所描述的，Java Persistence API 利用轻量级的 POJO 持久层取代了受容器托管的持久化。轻量级持久层为领域对象的建模提供了丰富的支持，其中包括继承和多态；针

对 EJB QL 的众多功能增强丰富了实体查询的能力；而注解或 XML 描述文件的使用则给出了一个具有移植能力的 O/R 映射规范。持久化实体现在成了普通 Java 类的实例（并且是受托管的）。因此，我们可以使用关键字 new 来创建实体，并将其作为普通的 Java 对象传入其他应用层。Java Persistence API 提供的受托管 persistence context 在 EJB 3.0 环境中起到了特别的作用，同时，它还使长期客户端会话（译注：extended client "conversations"，即 extended persistence context）的建模变得简单。

由于这些工作，EJB 3.0 极大地简化了企业应用开发，它所引入的很多特性业已成为 Java EE 平台的组成部分。不仅如此，我们还对 Java Persistence API 进行了扩展，使其可以支持在容器外的 Java SE 环境下使用。

Bill Burke 对 EJB 3.0 及 Java Persistence API 的订立功不可没。作为 JBoss 应用服务器的首席架构师，他为 EJB 3.0 的专家组带来了容器创新方面的重要见解，Hibernate 对象/关系持久化技术方面的丰富经验，还有在真实应用中使用 EJB 技术时急开发者之需的广博见识。这套崭新的，经过简化的 EJB 3.0 API 对企业 Java 开发的格局会产生怎样的影响？在新版的《Enterprise JavaBeans》一书中，在 Richard Monson-Haefel 早期工作的基础上，Bill Burke 将与应用开发者们一起分享他的真知灼见与深度观点。

Linda DeMichiel

EJB 3.0 Architect and Specification Lead

Sun Microsystems

Santa Clara, California

作者按语

2000 年春，我辞去了在 Iona 的 CORBA 开发工作，加入了一个构建电子商务门户的小型创业公司。从 CORBA 背景转向 Enterprise JavaBeans 项目，对我来说还有一些不太适应。我记得自己一直在对服务器端为何需要组件模型这一问题困惑不已。很快我了解到，声明性事务，安全，以及持久化元数据在商业应用开发中扮演着至关重要的角色。

2000 年末，和那个时候其他的许多创业公司一样，我们遇到了资金问题。我们有了第一位客户，系统即将进入 beta 阶段。就在这个时候，问题接踵而至，我们的应用服务器供应商收回了 demo 许可并要求付款。为此，我们不得不马上寻找免费的解决方案，这便是我与开源应用服务器——JBoss 的第一次不期而遇。

将应用迁移到 JBoss 并没有花费太长的时间，但是我发现 JBoss 2.0 在某些方面还稍显粗糙。就职于 Iona 期间，在 Steve Vinoski 的指导下，我在中间件开发方面积累了丰富的经验，因而要定位到代码的关键部分并非难事。我立刻开始着手修复 JBoss 中的各处 bug，并将补丁提交到 JBoss 的邮件列表。在持续几个月的贡献之后，我收到了一封陌生的电子邮件：

```
what do you do in real life  
你在现实中是做什么的？
```

这封邮件来自 JBoss 的创始人，Marc Fleury。说实话，JBoss 是我参与的第一个开源项目，收到项目创始人写来的邮件几乎就像是收到了来自上帝的信息。也许我说的有些夸张了。它更像是在电影《黑客帝国》中，收到 Morpheus 发来的一条经过编码的讯息。我回复 Marc 说，我正在为一家苦苦支撑的创业公司工作，而我的前一份工作则是在 Iona 开发中间件。Marc 再次回复的邮件中只有一句话：

```
do you want to take the red pill  
你想不想尝红色药丸？
```

你想不想尝红色药丸？（译注：原文此处为“red pill”，出自电影《黑客帝国》，与 blue pill 相对，是 Neo 所服的药丸。服用蓝色药丸会使人依旧存在于虚幻之中，而服用红色药丸则会让人知道整个事实的真相。）

我上了钩，回答是，于是我便开始了掉入兔子洞的奇异之旅（译注：原文此处为“rabbit hole”，也出自《黑客帝国》的台词，不过源出于英国人 Lewis Carroll 的畅销儿童读物《爱丽丝漫游奇境记》）。我成了 CVS 的提交者，并且开始着手解决 entity bean 的同步问题。在那年的晚些时候，我帮助 Sacha Labourey 完成了 JBoss 的首个集群实现。往后的事情众所周知，我就不再赘言了。

回到 2002 年，Richard Monson Haefel，本书前几个版本的作者，就为该书增加 JBoss 操作手册一事联系了 Marc。Marc 对此兴趣不大，因而他向 Richard 引荐了 Sacha Labourey 和我，于是我们一起编写了操作手册，该手册与 O'Reilly 出版的 Enterprise JavaBeans 第 4 版同步发行。Richard 于去年挂笔，我想大概是我的工作还算令人满意，所以 O'Reilly 给了我这样的机会，将这本系列书籍更新至最新的 EJB 3.0。

谁应该阅读本书

本书讲解并演示了 EJB 3.0 及 Java Persistence 编程模型的基础知识。虽然 EJB 令应用开发简化了许多，但它仍旧是一项复杂的技术，需要大量的时间来学习掌握。本书就底层技术、Java 类与接口、组件模型，以及 EJB 运行期行为这些问题所给出的解释，简洁明了而又不失严谨。只是，书中去除了仅出现于旧版规范中的内容。

虽然本书着重于讲解基础知识，但它并不是一本傻瓜式的书籍。EJB 是一项极度复杂而又雄心勃勃的企业级技术。使用 EJB 也许相当简单，但理解并掌握 EJB 却需要大量的工作。在阅读本书之前，你应该熟练掌握 Java 语言，并有过一些开发业务解决方案的实际经验。我们不要求你具备分布式对象系统方面的经验，但是你需要具备 JDBC 方面的经验（或者至少理解其基本概念），这样才能理解本书的示例。如果你对 Java 语言知之甚少，我向你推荐 *Learning Java* 一书，该书以前被称为 *Exploring Java*（二者均由 O'Reilly 出版）。如果你对 JDBC 不太熟悉，我向你推荐 *Database Programming with JDBC and Java* (O'Reilly)。如果你想对分布计算有更为深入的了解，我向你推荐 *Java Distributed Computing* (O'Reilly)。

本书内容是如何组织的

本书分为两大部分：技术篇，以及紧随其后的 JBoss 操作手册。技术篇解释了 EJB 是什么，EJB 如何工作，以及何时使用 EJB。JBoss 操作手册则对如何在 JBoss 4.0 应用服务器上安装、配置，以及运行前面章节中给出的示例提供了循序渐进的指导。

第 1 部分：EJB 3.0 标准

第 1 部分是 Bill Burke 在 Richard Monson-Haefel 所著的本书第 4 版的基础上改编而成的。Linda DeMichiel, EJB 3.0 规范的领导者, 为本书撰写了序。Jason T. Greene 改编了第 18 章和第 19 章。以下是技术篇各章内容的摘要:

第 1 章, 引介

这一章给出了组件事务监视器的定义, 并解释了它们如何构成 EJB 组件模型的底层技术。

第 2 章, 架构概览

这一章定义了 EJB 组件模型的体系架构, 并对 enterprise bean 三种基本类型之间的区别进行了分析, 它们分别是: entity bean, session bean, 以及 message-driven bean。

第 3 章, 资源管理与基本服务

这一章解释了 EJB 兼容的服务器如何在运行时管理 enterprise bean。

第 4 章, 编写你的第一组 Bean

这一章带领你学习了一些简单的 enterprise bean 与 entity bean 的开发过程。

第 5 章, 持久化服务 EntityManager

这一章解释了 entity bean 如何与新的 entity manager 服务进行交互。

第 6 章, 映射持久对象

这一章定义了由 Java Persistence 规范所提供的基本关系数据库映射。

第 7 章, 实体关联

这一章对第 6 章余下的议题做了讲解, 拓展了你对持久化和复杂 bean-to-bean 关联关系的理解。

第 8 章, 实体继承

这一章讨论了 entity bean 的继承, 以及将对象层次结构映射到关系数据库的方法。

第 9 章, 实体查询与 EJB QL

这一章提出了 Enterprise JavaBeans 查询语言 (EJB QL), 它被用来对 entity bean 进行查询, 并在 Java Persistence 中对特定的 entity bean 进行定位。

第 10 章, 实体回调与监听器

这一章涵盖了 entity bean 的生命周期, 并告诉你如何编写能够拦截实体生命周期事件的类。

第 11 章, Session Bean

这一章告诉你如何开发 stateless session bean 和 stateful session bean。