



OSGi

原理与最佳实践

林昊 曾宪杰 著



电子工业出版社
www.phei.com.cn

内容简介

本书基于作者使用OSGi的经验而编写，涵盖了OSGi从入门到深入的所有体系。从OSGi的简介开始，介绍OSGi的用途和基本概念，其后进入OSGi实践，结合案例详细讲解OSGi的架构和开发。如替代的普通Java应用，通过OSGi和Spring进行集成开发，通过OSGi构建和部署框架(Equinox, Felix, Spring-DM和Apache CXF)等，帮助读者深入理解OSGi。

OSGi

原理与最佳实践

图灵社区会员 (CIP) 数据

OSGi 原理与最佳实践 / 林昊, 曾志斌. — 北京: 电子工业出版社, 2009.8

ISBN 978-7-121-09242-8

I. O—II①林… III. 曾… IV. 737.5

中国版本图书馆CIP数据核字

第1版 (2009年8月)

第1次印刷

印 刷: 北京天竺星印刷厂

装 订: 三河市富康印刷厂

出版发行: 电子工业出版社

北京市西城区

开 本: 787×1092 1/16 印张: 17.2 字数: 320千字

印 次: 2009年8月第1次印刷

印 数: 4000册 定价: 39.80元

凡购买本书者, 均可享受售后服务。如有任何疑问, 请向本出版社或各分社联系。

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书基于作者多年使用 OSGi 的经验而编写,涵盖了 OSGi 从入门到深入的知识体系,从 OSGi 的简介开始,介绍 OSGi 的作用及基本概念;其后进入 OSGi 实战,结合实例讲解如何基于 OSGi 框架编写模块化、动态化的各种 Java 应用;最后对 OSGi 知识进行深入讲解,通过对 OSGi 规范和实现框架(Equinox、Felix、Spring-DM 和 Apache CXF)的分析,以及最佳实践的介绍,帮助读者更好地使用 OSGi。

本书适合希望了解、深入掌握 OSGi,以及编写模块化、动态化 Java 应用的 Java 架构师和开发人员阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

OSGi 原理与最佳实践 / 林昊, 曾宪杰著. —北京: 电子工业出版社, 2009.8

ISBN 978-7-121-09242-8

I. O… II. ①林…②曾… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 115898 号

策划编辑: 陈 琼 徐定翔

责任编辑: 陈元玉

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

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: 787×1092 1/16 印张: 17.5 字数: 350 千字

印 次: 2009 年 8 月第 1 次印刷

印 数: 4 000 册 定价: 39.80 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 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。

(序) 推荐序一

Peter Kriens

About two years ago I got a request from a guy called BlueDavy. He wanted to know if he could translate the bnd manual page into Chinese? He explained to me that there were lots of people using bnd in China and they really lacked some documentation. I was more than happy to give him a web page that he could use to provide the Chinese version. The result looked very strange to me. All these incomprehensible characters interspersed with well known words, like friends you serendipitously meet in a far away city. And now there is also an OSGi book in Chinese.

When we started working on the OSGi specifications in 1998 we did not foresee that one day there would be a Chinese book about our efforts. And now there is a Chinese book even before there is an English book! A number of English books are in the making, but BlueDavy has beaten them to the punch. And this is good. China is clearly emerging as an economic powerhouse and there must be many programmers that can benefit from OSGi. OSGi is the de-facto standard for modularity. Look at the major Java Application servers, they are all running on top of OSGi. These application servers are very large applications themselves, so large that modularity is not an option, it is mandatory. More and more applications are moving in the realm where modularity is a requirement to survive. This is caused by the success of open source projects that make it easy to build systems based on large swaths of functionality, as well of the natural progression, where over time, things always grow.

OSGi is a quest for reusable components, a dream of our industry since the dawn of computing. As one of the authors of the specifications, I am the first to admit we haven't found the grail yet. However, I am convinced that OSGi is further on this path than any other standard or product. We do need help in this quest. With the OSGi specifications you get the fruits of an effort that has costed tens of millions of dollars to produce, in return we are asking for your feedback, support, participation in the Chinese user group, or best, becoming an OSGi member. Having a single modularity system is crucial to move Java to the next level, and for that we need you.

BlueDavy has done a tremendous job of making the OSGi Core specifications clear in Chinese. And though I see many of my friends in the text, unfortunately, the rest is Chinese to me. It looks like this book is a thing of beauty that I cannot see.

Peter Kriens, Beaulieu, July 2009

推荐序一（译）

Peter Kriens

大约两年前我收到了 BlueDavy 发来的一封邮件，询问我能否授权他将 bnd 手册翻译成中文，他解释说中国有很多人在使用 bnd，但是却非常缺乏相关的文档。我非常高兴地为 him 提供了一个网页，用来发布中文版本。翻译出来的内容看起来很奇怪，一大堆不认识的汉字中夹杂着几个熟悉的英文单词，这种感觉就好像在一个遥远陌生的城市中偶然碰到了你的朋友。现在又有了一本中文版的 OSGi 书籍。

1998 年我们开始从事 OSGi 规范的制定时，完全没有预料到会有一本中文版的 OSGi 书籍面世，而且先于英文版本。当英文版本还在编写时，BlueDavy 抢先一步完成了中文版，这太棒了。中国是一个新兴的经济大国，相信很多程序员能够从 OSGi 中获益。OSGi 是事实上的模块化标准，现在主要的 Java 应用服务器都运行在 OSGi 上。对于这些巨大的软件而言，模块化已经不是可选项，而是必须的。很多的领域软件都是基于开源产品而构建的，随着开源项目的不断成功，产品的功能也越来越复杂，就像自然界一样，事物总是在不断发展的，在这样的情况下，大家逐渐意识到模块化已经成为生存的必备条件。

OSGi 的目标是组件级的复用，这也是自计算机诞生以来我们行业的梦想。作为 OSGi 规范的作者之一，我首先承认我们目前还没有找到圣杯，但我坚信 OSGi 会比其他的规范或产品走得更远。达到这个目标，需要大家的帮助。OSGi 规范帮助软件行业创造了数千万美元的价值，反过来，我们也希望大家能够反馈意见到中国用户组，支持、参与中国用户组，当然最好能成为 OSGi 联盟的成员。采用统一的模块化系统对于将 Java 提升到一个新的水平而言是非常关键的，为此我们需要大家的帮助。

BlueDavy 为翻译 OSGi Core 英文规范做了大量的工作，虽然我在书中找到了许多熟悉的单词，但我的中文水平还不足以让我阅读其余部分。看起来，这本书是一种我没办法看懂的美好事物。

Peter Kriens

2009 年 7 月于英国博利厄市

关于 Peter Kriens

Peter Kriens 是 OSGi 联盟的技术负责人，曾在多家知名公司工作，包括 Intel、Ericsson、Motorola、Adobe、IBM 和 Nokia。

他的 blog 为：<http://www.osgi.org/blog> 及 <http://www.aqute.biz/Blog/HomePage>。

点财 推荐序二

与世界的接口

几年前，关注 OSGi 的人还很少的时候，林昊就和我讨论如何将 OSGi 应用在实际工作中。程序员在编写大量代码后，也会遇到复用和模块化的问题。相信每位认真的开发者都有过模块化的尝试，几次之后才会发现最大的挑战并非代码实现，而是来自于模块边界的定义。如何界定模块的边界，需要开发者有大量的经验和对体系的深刻理解。比较 JavaBean 这种 JVM 级别的编译式复用，OSGi 面对的无疑是更大的挑战，它给我的印象是希望达成健壮的、可以热插拔的、较粗粒度的模块化边界定义方案，幸运的是它成功了。针对这种边界方案的各种实现，几年之后，有了最著名的 Equinox 和 Felix。OSGi 也已经成为各大应用服务器的一致选择和实现标准，它成了你的代码与世界的接口。

在这次 Java 进化的过程中，林昊以技术人员的敏锐和韧性始终保持着对 OSGi 的持续关注，我相信他不止一次地体会到了兴奋和快乐。最难能可贵的是他希望将这种快乐传递给他人，因此，他以 BlueDavy 的网名编写了《OSGi 实战》和《OSGi 进阶》两份 Opendoc，很长时间内这是仅有的中文 OSGi 入门资料，他将很多人带入了 OSGi 的大门，而这本书的出版，也必将为 OSGi 在国内的研究和普及掀起一次高潮。

本书是林昊多年研究的心血结晶，非常高兴能看到本书最终定稿出版。值得一提的是，满江红组织在审校范里程翻译的《OSGi R4 规范》中文版的过程中，林昊出任过 Leader 一职，为 OSGi 核心文档的翻译贡献了心力。

梅花香自苦寒来，向林昊表示由衷的祝贺。

满江红开放技术研究组织 曹晓钢

2009 年 7 月于上海

联系博文视点

联系我们

您可以通过如下方式与本书的出版方取得联系。

读者信箱: reader@broadview.com.cn

投稿信箱: bvtougao@gmail.com

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

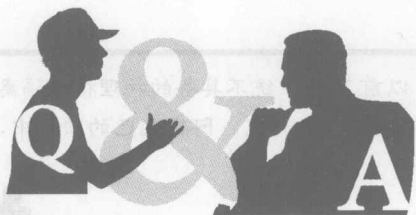
传 真: 027-87690595

欢迎您访问博文视点官方博客: <http://blog.csdn.net/bvbook>

附赠图书 博文视点图书发行目录

2009年7月

编著往来



林昊与曾宪杰先生的新书《OSGi 原理与最佳实践》即将出版，本书策划编辑徐定翔和林先生以对话的形式交流了一些技术话题，从这些交流中能看到一本原创著作的问世，包含了作者多少深入的思考。



博文编辑

林先生，我们知道您 2007 年底加入淘宝网，任职淘宝平台架构部架构师，可否先请您向读者介绍一下您在淘宝网的工作情况。

林昊



加入淘宝后主要负责淘宝分布式服务平台方面的工作。分布式服务平台承担了淘宝几乎所有 Java 远程调用的支撑及管理，因此其在通信的性能、稳定性及服务的治理上都有很高的要求。



博文编辑

您是国内推广 OSGi 的第一人，是 OSGi 联盟官方授权的 China User Group Director。两年曾经编写了《OSGi 实战》和《OSGi 进阶》两个文档，分享了您学习和使用 OSGi 的经验，在技术社区受到了广泛的欢迎。可否请您介绍一下这本《OSGi 原理与最佳实践》与之前两个文档的区别与联系。

林昊



为了避免自己陷入之前编写 OSGi Opendoc 的惯性思维，《OSGi 原理与最佳实践》这本书特意邀请了我的同事，同样是淘宝平台架构部架构师的曾宪杰来编写，这样保证了这本书与之前两个 OSGi Opendoc 的不同。《OSGi 原理与最佳实践》相对两个 Opendoc 而言，更加系统和全面地介绍了 OSGi 的入门知识和 OSGi 的深入知识，而 OSGi 的深入知识是之前两个 Opendoc 中提及甚少的部分。对于 OSGi 及 OSGi 实现框架的深入掌握，加上笔者在使用 OSGi 方面的经验，相信能够对大家更好地基于 OSGi 编写模块化、动态化的 Java 系统提供帮助。另外在例子方面，提供了大家更加熟悉的 petstore 例子的实现，以及有不少需求的分布式场景的例子。

来封著录



博文编辑

OSGi 框架被认为通过组件 (bundle) 的形式, 实现了以前 Java 系统不具备的物理模块隔离, 具有模块化、动态化和可扩展的特点。就您所知, 在淘宝、支付宝、阿里巴巴的项目中, OSGi 框架是否真的发挥了这些优势呢?

林昊



淘宝、支付宝及阿里巴巴在其服务平台中均使用了 OSGi 框架, 以这三家公司的使用情况来看, OSGi 确实对构造模块化、动态化的系统起到了重大作用, 例如模块化带来的 classloader 的隔离, 对于避免服务平台和应用系统依赖的 jar 版本冲突带来了很大的帮助。



博文编辑

请您展望一下 OSGi 未来的发展趋势。

林昊



目前 OSGi 的发展状况远好于两年前对它的期待, Java 主流应用服务器的采用, Java7 从语言级对模块化的支持, 这些都将使得 OSGi 逐渐成为热门技术, 甚至是 Java 语言的必备技术。



博文编辑

您目前正在撰写另一部作品《构建高性能的大型分布式 Java 应用》, 可否向读者透露一下这本书的内容定位和目前的进展?

林昊



《构建高性能的大型分布式 Java 应用》旨在深入介绍构建高性能、大型分布式 Java 应用所须掌握的知识体系, 包含了分布式 Java 应用的通信、远程调用涉及的知识体系; JVM 的深入分析, 包括 JVM 的内存机制、内存回收机制 (例如现在的 CMS, 引人注目的 G1) 及线程机制; JDK 中与高性能分布式 Java 应用有关的包的深入分析; 性能测试及调优的知识; 软件负载、硬件负载涉及的相关知识体系等。目前完成了大概 30% 的内容, 预计于 11 月中旬完成全部初稿。

序 一

两年前，曾向周筠老师提及想出本 OSGi 方面的书，那时还没有多少人对 OSGi 有学习的需求。技术图书出版的多少能在一定程度上说明这门技术在业界的接受程度，如 Struts、Hibernate 等技术都是如此。但在 2007 年，OSGi 能否被业界接受却还没有定论，当时 OSGi 的使用者增幅也有限。到了 2009 年，OSGi 终于浮出水面。使用者也从当初仅有的 Eclipse 到现在已是所有主流的 Java 应用服务器、各种模块化框架、淘宝、支付宝、阿里巴巴等都在使用。在出版方面，不论是知名的 IT 英文出版社，还是 IT 中文出版社，也都已开始在策划 OSGi 方面的书籍，我很幸运得到了周筠老师的信赖，受托撰写这本 OSGi 的书，而这本书也是目前市面很少见的一本能够领先于英文相关书籍出版的中文技术原创书，略感遗憾的是这本书没有成为全球第一本 OSGi 书籍。据 Peter Kriens 的消息，德国已经出版了一本 OSGi 的书籍，因此本书也就只能获得全球第二本 OSGi 书籍或国内第一本中文 OSGi 书籍这样的称号了。当然，这些都是虚名。我撰写本书的目的一方面是圆自己出版一本 OSGi 书的梦想，另一方面也是希望国内有越来越多的开发者们能够开始使用 OSGi，甚至参与到 OSGi 框架的实现及 OSGi 的改进中来。

我接触 OSGi 是在 2005 年，2006 年开始在商业产品中实际应用它，说起来也并不算很早，在使用 OSGi 过程中碰到的问题及中文文档的缺乏让我付出了很大的学习代价，于是先后编写了《OSGi 实战》和《OSGi 进阶》两个 Opendoc，提供给开发者们参考，想尽可能地加快后来者们学习的进度，以及减少后来者们所走的弯路。但随着时间的推移，由于 OSGi 的高速发展，这两篇 Opendoc 都已在一定程度上过时。恰好博文视点给了我写书的委托，于是就有了本书。本书对 OSGi 的知识体系进行了由浅入深的介绍，希望对有意愿了解、学习和使用 OSGi 的，以及想深入掌握 OSGi 的读者们有一定的帮助。

为了避免自己陷入之前写 Opendoc 的惯性思维，特邀请了曾宪杰一起编写本书，而正是因为他不同的思维角度，给本书带来了更清晰的讲解及更多更好的知识点，非常感谢他的执着努力和认真参与。

最后要感谢我的女友：宗伟，正是因为她的理解我才能在下班后和周末抽出足够的时间来编写本书，谢谢！

淘宝网 架构师 林昊（网名：BlueDavy）

2009 年 6 月于杭州

序 二

第一次接触 OSGi 大约是在 2007 年的下半年，当时公司的一个系统考虑采用基于 OSGi 的架构来实现。那时自己才第一次接触 OSGi，对 OSGi 开始有了一些初步的认识，也大体了解了 OSGi 是怎么一回事。虽然那时就对 OSGi 提供的特性很感兴趣，但还没有深入去了解和使用它。真正深入了解 OSGi 是在和林昊成为同事之后，是从他的两篇 Opendoc 入门的。坦白来说，OSGi 的资料和文档真的比较少，多亏了林昊的 Opendoc 和附带的例子让我入了门（相信很多人跟我有同感吧）。之后，开始在自己负责的产品中尝试应用 OSGi。

2008 年，林昊邀请我一起撰写本书。自己感到高兴的同时，也希望为更多想了解、学习和使用 OSGi 的朋友尽一份力。写作的过程还是蛮辛苦的，花了很多时间在例子的构思和代码的编写上，其中也遇到过 Apache CXF-DOSGi 的 Bug 等问题。但是当本书草稿完成的时候，回过头去看自己完成的部分，还是非常欣慰的。想到自己书写的这些内容及例子能够帮助很多人少走弯路，或者说帮助很多人能更快地掌握 OSGi，觉得付出的辛苦还是非常有意义的。

本书从 OSGi 的简介开始，到 OSGi 的展望收尾。其中介绍了 OSGi 的框架及使用，解读了 OSGi 的规范本身，分析了 OSGi 框架的源码，应该说涵盖了比较丰富的内容。本书不仅能让读者会用 OSGi 来开发应用，也能让读者了解许多框架之后的东西，能够理解实现本身，也希望更多的人能够通过本书进入 OSGi 的世界。

最后，更多的是感谢：

感谢博文、感谢周筠老师给了我们这样一个机会，能够参与编写这样一本 OSGi 的书。

感谢林昊的邀请及在写作过程中的帮助。

感谢老爸老妈精神的鼓励和支持。

最后，要感谢我的女友，感谢她对我的理解和支持，让我能够把大部分的业余时间都用来专心写书。

淘宝网 架构师 曾宪杰

2009 年 6 月于杭州

前言

Java 7 的发布日期临近, 模块化是 Java 7 中最重要的特性之一。在 Java 语言级对模块化提供支持之前, OSGi 已经是业界中最知名的 Java 模块化规范。OSGi 联盟成立于 1999 年, 发展到今天已经得到了众多企业、厂商、开源组织的支持, 尤其当主流的 Java 应用服务器 (Oracle 的 Weblogic、IBM 的 Websphere 及 Sun 的 Glassfish 等) 都采用 OSGi 时, OSGi 作为 Java 模块化标准已成为事实。掌握 OSGi 是实现模块化 Java 应用的必备技能, 在将来甚至会成为 Java 语言中必须学习的技能之一, 就像现在 Java 中的泛型一样, 而动态化也是 OSGi 的另一特性。OSGi 对于动态化的支持能够帮助开发者更好地实现“即插即用”、热部署及“即删即无”的系统。

本书作为一本早于同类技术英文书而编写的 OSGi 中文书籍, 旨在为希望实现模块化、动态化 Java 系统的架构师和开发工程师提供 OSGi 入门知识, 同时也为希望深入掌握 OSGi 的架构师、开发工程师提供 OSGi 知识的深入讲解。本书内容从 OSGi 的简介开始, 到 OSGi 框架的使用, 再到 OSGi 规范的掌握, 最后到 OSGi 框架的实现分析, 阐述了基于 OSGi 编写模块化、动态化的 Java 系统须要掌握的知识体系, 希望本书能给读者带来一次愉快的 OSGi 之旅。

本书第 1 章为读者揭开了 OSGi 的面纱, 以帮助读者了解 OSGi、其发展历程及掌握 OSGi 的基本概念。

第 2 章、第 3 章、第 4 章、第 5 章结合简单例子及经典的 PetStore 例子对 OSGi 框架 (Equinox、Felix 及 Spring-DM) 的使用进行了介绍, 同时介绍了 OSGi 在 B/S、分布式等多种典型场景中的使用方法。

第 6 章、第 7 章、第 8 章、第 9 章及第 10 章对 OSGi 规范进行了详细分析, 以帮助读者掌握 OSGi 的本质, 对典型 OSGi 框架的实现进行了详细的分析, 并结合作者的经验对 OSGi 框架的使用提供了一些实践范例, 以帮助读者在实际的项目中更加合理地使用 OSGi 框架, 进而编写模块化、动态化的 Java 应用系统。

第 11 章和第 12 章对 OSGi 的未来做了一些阐述, 包括即将发布的 OSGi R 4.2 规范, 以及根据作者的经验对 OSGi 的未来做了一些大胆的推测。

本书包含了众多的实例, 由于篇幅关系, 书中仅列出了代码的片断, 如需完整代码, 请到 <http://china.osgiusers.org> 中下载。

由于本书编写有些仓促, 有些知识点来不及纳入书中, 这些知识点都会在 <http://www.blogjava.net/bluedavy> 上进行发布, 感兴趣的读者可以移步到此网站进行更多的阅读。

目录

第 1 章 OSGi 简介.....	1
1.1 梦想中“即插即用”的系统.....	1
1.2 OSGi, 拯救 Java 模块化的规范.....	1
1.3 厚积薄发的 OSGi.....	2
1.4 OSGi 基本概念.....	3
1.4.1 Bundle.....	3
1.4.2 Service.....	4
1.4.3 Service-Oriented Component Model (SOCM).....	4
1.4.4 Declarative Service.....	6
1.4.5 部署.....	6
第 2 章 OSGi 框架简介.....	7
2.1 Equinox.....	7
2.1.1 简介.....	7
2.1.2 环境搭建.....	7
2.1.3 HelloWorld.....	10
2.1.4 开发传统类型的应用.....	17
2.1.5 从外部启动 Equinox.....	35
2.2 Felix.....	38
2.2.1 简介.....	38
2.2.2 环境搭建.....	38
2.2.3 应用的部署.....	38
2.2.4 在 Eclipse 中调试 Felix.....	39
2.3 Spring-DM.....	43
2.3.1 简介.....	43
2.3.2 环境搭建.....	44
2.3.3 HelloWorld.....	48
2.3.4 Web 版 HelloWorld.....	51
第 3 章 基于 Spring-DM 实现 Petstore.....	57
3.1 “即插即用”的 Petstore.....	57

3.1.1	Petstore 的功能需求.....	57
3.1.2	OSGi 框架的功能和设计思想.....	58
3.1.3	Petstore 的设计.....	60
3.2	新一代 Petstore 的实现.....	70
3.2.1	环境准备.....	70
3.2.2	Utils 模块.....	72
3.2.3	Bootstrap 模块.....	74
3.2.4	ProductDal 模块.....	79
3.2.5	ShoppingCartDal 模块.....	80
3.2.6	ProductList 模块.....	81
3.2.7	ShoppingCart 模块.....	85
3.2.8	ProductManagement 模块.....	85
3.3	部署.....	85
3.4	Petstore 的扩展.....	85
第 4 章	基于 Apache CXF 实现分布式 Petstore.....	87
4.1	分布式 OSGi.....	87
4.2	分布式 Petstore 的设计.....	87
4.3	Apache CXF 简介.....	90
4.3.1	CXF-DOSGi HelloWorld.....	90
4.3.2	集成 Spring-DM 的 CXF-DOSGi HelloWorld.....	95
4.4	分布式 Petstore 的实现.....	101
4.5	部署.....	105
第 5 章	构建 OSGi Bundle Repository.....	107
5.1	OSGi Bundle Repository 的使用.....	107
5.2	Apache Felix OBR 的使用.....	110
5.2.1	命令行方式的用法.....	110
5.2.2	代码方式使用 OBR.....	113
5.3	构建 OSGi Bundle Repository.....	117
第 6 章	OSGi 规范解读.....	119
6.1	OSGi 规范发展历程.....	119
6.2	Core 规范解读.....	120
6.2.1	Module Layer.....	120
6.2.2	LifeCycle Layer.....	129
6.2.3	Service Layer.....	132
6.3	OSGi Service 规范解读.....	133
6.3.1	Log Service.....	134
6.3.2	Http Service.....	137

6.3.3	Configuration Admin Service	139
6.3.4	Declarative Services.....	142
6.3.5	Event Admin Service	145
第 7 章	Equinox 实现分析	149
7.1	模块化	149
7.1.1	加载并解析 jar 为 Bundle.....	149
7.1.2	Bundle 类共享及类加载机制.....	152
7.2	动态化	159
第 8 章	Felix 实现分析.....	173
8.1	模块化	173
8.1.1	加载并解析 jar 为 Bundle.....	173
8.1.2	Bundle 类共享及类加载机制.....	174
8.2	动态化	177
第 9 章	Spring-DM 实现分析	189
9.1	Spring Dynamic Module 概述.....	189
9.2	Bean 发布为 OSGi 服务实现分析	190
9.2.1	Extender Bundle 启动.....	190
9.2.2	Bundle 中的 NamespaceHandler 处理.....	192
9.2.3	ExtenderConfiguration 处理.....	194
9.2.4	ApplicationContext 处理	200
9.3	将 OSGi Service 引入为 Bean	209
9.3.1	osgi:service 的处理	209
9.3.2	osgi:set 和 osgi:list 的处理.....	211
9.4	将 Bundle 部署到 Web 服务器.....	214
9.4.1	WebExtender 的启动	214
9.4.2	部署 Bundle 到 Web 服务器	215
第 10 章	CXF 实现分析	217
10.1	Apache CXF-DOSGi 实现分析	217
10.2	SingleBundle 发行包.....	217
10.2.1	SingleBundle 发行包的启动.....	218
10.3	把 Bundle 暴露为远程服务	221
10.3.1	DSW Bundle 启动	221
10.3.2	RegisterDistributionProviderService	222
10.3.3	创建 Hook 对象及注册 Listener.....	224
10.3.4	处理当前框架中的 Bundle	225
10.3.5	发布远程服务.....	225
10.4	使用远程服务	231

139	10.4.1	Discovery Bundle 启动	231
142	10.4.2	处理当前框架中的 Bundle	232
142	10.4.3	注册远程服务的本地存根	234
140	第 11 章	先睹为快: OSGi R 4.2 草稿版	237
149	11.1	Core	238
149	11.1.1	Module Layer	238
152	11.1.2	Lifecycle Layer	238
159	11.1.3	Service Layer	242
173	11.1.4	Framework API	243
173	11.2	Compendium	243
173	11.2.1	Declarative Services Specification	243
174	11.2.2	Deployment Admin Specification	244
177	11.2.3	Monitor Admin Service Specification	247
180	第 12 章	OSGi 展望	251
180	12.1	Java 企业应用领域	251
190	12.2	Java 标准领域	252
190	12.3	OSGi R5	253
194	索引		255

第 1 章 OSGi 简介

1.1 梦想中“即插即用”的系统

第一次听到插件应该是当初用 WinAMP 播放器的时候，记得当时 WinAMP 有很多的插件，可以方便地下载、安装、使用。在实际的工作中，有过引入 COM 组件技术使系统模块化的经历，后来自己也做过一个产品用的插件管理器（实在不能说是插件平台），实现了基本的插件远程下载、安装、启动、停止等管理工作，也有插件间的消息通知。但是这个管理器更像是对模块的动态管理，并且在实现上也发现它远没有最初设想的简单。还有一个问题是，这个插件管理器是用在公司的第一款产品上，目标是做到公司内通用，但是最终发现其实并不是那么通用，并且这个管理器也有些相关平台。当时就在想，有没有一个通用的框架，实现一个插件体系结构，使用者遵循规范，开发自己的插件，使用已有的插件，而插件又能控制对外暴露的内容，在运行时，可以动态地安装、启动、停止、卸载插件，每个插件提供一个或多个服务，其他的插件只是根据接口来获取服务提供者，而完全不知道提供者是谁。如果有了这样的规范和框架，我们就可以用已有的积木搭出各种造型，也能够提供积木给自己、给别人来使用。所以，我们很希望能够有一个稳定的平台，管理规范的插件，插件可以热插拔，并且在运行期能够改变行为。

1.2 OSGi，拯救 Java 模块化的规范

在 Java 中，是没有模块的概念的。我们不能把一些代码组成一个模块，然后控制哪些是可以外部使用的，哪些是内部使用的。我们能用的主要有两个办法，一个是把一些内部使用的类赋予包内访问权限，但是这样在内部使用上会很不方便，因为使用这个类的类和本类本身要在同一个包下；另外一个办法是通过文档告知它是内部类，不能在外部使用，但是这种限制很容易被忽略。而内部类被外部使用的后果是，不能轻易调整接口，因为不知道内部类是否被外部使用了。如果调整，可能会导致外部系统受很大影响。难道就没有一种机制来保证我们只暴露想暴露的部分给外部么？