



SUN中国技术社区

推荐技术用书

developers.sun.com.cn

GlassFish

——开源的Java EE应用服务器

主编 蒋 健 王 昱
黄健昌 李 力
编著 叶 亮 颜曙光

- 由来自Sun工程研究院的资深Java工程师联手打造
- 创作原稿经过GlassFish核心开发团队人员细致审阅
- 基于书中内容开发的动手实验室入选JavaOne 2007



清华大学出版社



GlassFish——开源的 Java EE

应用服务器

主编	蒋 健	王 昱
	黄健昌	李 力
编著	叶 亮	颜曙光

清华大学出版社
北 京



这是一本关于开源的书，也是受益于开源而得以完成的书。没有开源，难以想象此书的出版能和产品的正式发布几乎同时进行；没有开源，难以想象一群不是来自产品开发团队的人会去讲述产品的实现和架构；没有开源，难以想象能获得如此丰富的资源并得到许多从未谋面的社区成员的热情支持；没有开源，难以想象这么多人愿意为之付出大量的时间和精力进行创作并使之完善。

——蒋 健

对于开发人员，开源项目最大的诱惑就是源代码，从源代码中可以学到书本上学不到的东西。当你静下心来，细细品味GlassFish源代码的时候，相信有很多让你心潮澎湃的地方，使你进入一个新的起点。

——王 昱

GlassFish是第一个实现Java EE 5规范的应用服务器，通过GlassFish，你可以快速学习和部署Java EE 5企业应用。在实现上，GlassFish采用了很多新技术和架构；同时，开放的源代码可以让你更深入地了解应用服务器运行的原理，从应用级开发深入到系统级开发。

——黄健昌

开源软件降低了开发者进入的门槛，也大大加快了新技术的普及和应用。开源软件不仅吸引着众多的开发者，对很多商业机构来说，开源软件也孕育着商业机会。加入开源运动必有收获。

——李 力

Java对开源世界的贡献有目共睹，许多Java的项目或框架也相当成功，并且在一定程度上已成为事实上的工业标准；本书的主要目的之一就是通过优秀的开源应用服务器GlassFish，让人们了解和认识最先进、最“时髦”的技术。

——叶 亮

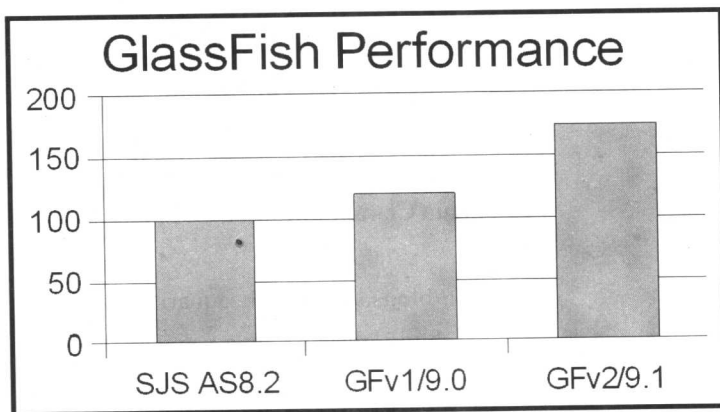
序 言

GlassFish 开始于 2005 年 6 月，或者说开始于 1999 年 6 月，这取决于你如何看待它。

2005 年 6 月 Sun 公司宣布将为 Java EE 5 的参考实现及其商业产品 Sun Java System Application Server 9.0 使用开放源代码许可(我们称之为 GlassFish 项目)。随着时间的推移，GlassFish 已成为社区及其所构建的应用服务器的代称。在 2006 年 5 月，我们发布了 Java EE 5 最终的参考实现和 GlassFish v1，以及有 Sun 公司商业支持的 SJS AS 9.0。目前我们计划在两个月之内，即 2007 年 9 月，发布 GlassFish v2(SJS AS 9.1)。

回到 1999 年 6 月，Sun 将其 Servlet 实现贡献给 Apache Software Foundation 并且加入 JServ 用户组联盟，从而启动了 Tomcat 项目。Tomcat 对行业产生了巨大的影响，它意味着 Java 更多地涉足服务器端应用并且在企业级应用领域逐步使用开源代码。如今，我和当时许多参与 Tomcat 创建的人正在为 GlassFish 社区工作，我们正在将自 1999 年 6 月以来所收获的经验教训贡献到 GlassFish 中去，我们希望整个社区能因此受益。在这个意义上，我们又可以说 GlassFish 开始于 1999 年 6 月。

GlassFish 首先是一个社区，其所致力于发布的是开源应用服务器并将其作为 Java EE 的参考实现。这个应用服务器拥有所有同类商业软件中你所期望的特性：集群(Clustering)、高可靠性(High-availability)、高性能(近期通过了业界领先的 SPECjAppServer 2004 benchmark 测试)(见下图)、微软应用的互操作性、完善的管理特性和文档、商业支持和培训等。作为 Java EE 的参考实现，它是市场上第一个符合 Java EE 标准的实现。



GlassFish 还提供了一个成功的开源项目期望拥有的各种特性：免费使用、紧密的用户交互及快速的响应、非常敏捷的开发模式、规范化的构建发布、与其他开源项目开放透明的集成、对所有流行的开源架构的支持、热情的社区支持……

拥有所有这些特性的关键是“社区的参与”，这也是为什么说这本书有超乎想象的重要性。



考察 2007 年以来 GlassFish 被采用的情况时世界各地的不同地区之间表现出一定的差异性。虽然只有几个月，但 GlassFish 应用的增长令人印象深刻，一些地区的增长明显快于其他地区。欧洲非常坚实，中国也提升很多——很大程度上要感谢使这本书产生的人们。中国与开源有着密切的关联，它应当体现出更多的下载和地理上更广泛的分布。

GlassFish 应用的进一步增长将应当感谢本书和为之付出努力的人们。本书由 GlassFish 社区的成员完成。他们直接参与了开发或实际使用了 GlassFish。本书将从 GlassFish 基础讲起，覆盖实践和社区的许多方面。作者都是 Sun 公司的职员，他们每日所使用和参与构建的就是 GlassFish，我非常感谢他们的工作和对本书的贡献。

本书及由社区组织的其他活动将具有巨大的影响。这些影响可以通过 GlassFish 被采用的分布地图(<http://beta.glassfish.java.net:81/maps>)中看到。

我们相信 GlassFish 是最好的商业和开源世界的产品。我鼓励大家参与到社区中来，获取大量的机会。这些机会包括：使用最终的产品版本、对不同的里程碑(Milestone)产品进行测试、为社区的 Wiki(在线协作)做出贡献、参与到大学和研究中心的研发当中、贡献代码修正和补丁等。

我们希望能在正在创建的各类草根(Grass-root)用户讨论组中见到你。

Eduardo Pelegri-Llopert

GlassFish 项目总负责人

Distinguished Engineer
Sun Microsystems

2007 年 8 月 1 日

Santa Clara, California, U.S.A.

<http://blogs.sun.com/theaquarium>

<http://glassfish.java.net>

前 言

GlassFish 是一个以社区为基础,全面实现了 Java EE 5 各项标准并拥有众多独到特性的开源项目。正如 GlassFish(玻璃鱼)名字所寓意的,象玻璃一样透明开放,象鱼一样自由自在,GlassFish 项目正在把 Java 企业级开源应用推向前所未有的广度和深度。本书力图将这一技术尽快、尽可能全面地介绍给读者。

本书可以划分成 3 篇:基础篇、应用篇和开源篇,将由浅入深,由表及里地介绍 GlassFish 项目。其中基础篇将介绍 Java EE 的基础概念和 GlassFish 的构建方法,应用篇将介绍如何管理 GlassFish 和开发 GlassFish 平台上的应用,开源篇将选取 GlassFish 的特色技术,结合其代码实现加以介绍。GlassFish 对 Java EE 5 中新规范的支持以及其自身众多的独到特性是贯穿全书的两个主要线索。

本书不是 Java EE 5 大全,而是选取 Java EE5 的新技术和 GlassFish 的独到特性重点加以介绍;本书不是 GlassFish 的在线帮助,而是在归纳和总结的基础上辅以实例帮助读者理解和把握 GlassFish;本书不是开发指南,而是侧重于 Java EE 上的 Web 应用开发的新进展和 GlassFish 特有的支持,让大家体会 Java EE 给开发人员带来的新的感受;本书也不是源代码的 javadoc,而是通过诠释代码来说明代码背后的设计思想和理念。

1. 读者对象

本书针对以下 6 类用户的特定需求提供有价值的参考。

- (1) 在校师生和开源软件入门者(Java EE 基本概念的讲解和相关资源介绍)。
- (2) 开源社区成员(相关标准概况和技术趋势以及开源社区相关服务介绍)。
- (3) 软件开发人员(讲解开发并部署各类应用到 GlassFish 的步骤和技巧)。
- (4) 用户及维护人员(基于 GlassFish 应用中的典型问题研究)。
- (5) 企业信息主管(CIO)(介绍最新动态和发展趋势,帮助进行项目选型)。
- (6) 开源软件开发者(GlassFish 架构及典型代码研读,为开源软件开发提供借鉴)。

2. 内容安排

(1) **基础篇**(第 1~3 章)将概要介绍 GlassFish 的背景、关联技术和标准、开源社区及安装和构建。

第 1 章: Java EE 5 和 GlassFish。

第 2 章: 开源和社区。

第 3 章: 安装和构建 GlassFish。

(2) **应用篇**(第 4~14 章)将从应用 GlassFish 的角度来介绍 GlassFish,包括如何管理 GlassFish,如何开发基于 GlassFish 的应用并对这些应用进行优化和配置。

第 4 章: 管理控制台。



第 5 章: GlassFish 命令行。

第 6 章: Web 应用程序基础和 Servlet。

第 7 章: JSP 编程。

第 8 章: JSF 编程。

第 9 章: AJAX 技术。

第 10 章: Web 服务。

第 11 章: EJB 3.0 和 JPA 开发。

第 12 章: GlassFish 的安全性。

第 13 章: 国际化和本地化。

第 14 章: Java EE 应用性能调优。

(3) **开源篇**(第 15~20 章)面向中高级开发人员和技术主管,更加注重从源代码的角度来介绍 GlassFish 技术,探讨 GlassFish 中的设计思想和实现技巧。

第 15 章: JMX 在 GlassFish 中的应用。

第 16 章: 自管理和 CallFlow。

第 17 章: 高性能的 HTTP 引擎——Grizzly。

第 18 章: 异步请求处理和服务器推送技术。

第 19 章: EJB 容器实现。

第 20 章: 内部持久性实现和 Java DB。

3. 参编作者

本书作者包括蒋健、王昱、黄健昌、李力、叶亮和颜曙光。其中,蒋健编写第 1、2、3、4、5、10、15、16、20 章;王昱编写第 14、17、18 章;黄健昌编写第 11、19 章;李力编写第 9、10 章;叶亮编写第 2、6、7、8 章;颜曙光编写第 12、13 章。

4. 致谢

这里要特别感谢 Sun 公司的欧阳庆庆(Qingqing Ouyang)——GlassFish 中文博客“水族馆”的负责人,她为本书提供了许多建议并出色地组织了对本书的审阅。

本书由多位活跃在 GlassFish 社区开发第一线的技术专家审阅——Anissa Lam、Cheng Fang、张虹(Hong Zhang)、Jane Young、焦瑛(Dianne Jiao)、陈成威(Shing Wai Chan)、Kin-man Chung、冷林洁(Jie Lin Leng)、Yifeng Luo、Tony Zhang。他们的深刻见解、评论和建议使本书在各方面都得到了很大的提升。

Sun 中国技术社区的蒋清野(John Jiang)为本书提供了多方面的资源和支持。Sun 公司 AGC 部门的肖顺敏(Serena Xiao)和董清富(Simford Dong)阅读了本书的部分书稿并提供了宝贵的意见。在本书出版之际,我们向以上各位的支持致以最诚挚的感谢。

5. 本书的网站和联络信息

读者可以参阅 Sun 中国技术社区上的博客 <http://developers.sun.com.cn/blog/java/> 获得关于本书的最新信息。您的评论和建议可以通过邮箱 glassfish_press@sun.com 反馈给作者。

目 录

基 础 篇

第 1 章 Java EE 5 和 GlassFish.....3	
1.1 Java EE 5 概述.....3	
1.2 Java EE 5 架构.....4	
1.2.1 容器(Container).....6	
1.2.2 应用组件(Application Component).....6	
1.2.3 Java EE 5 的标准服务.....7	
1.3 Java EE 5 的新特性.....9	
1.3.1 标注.....9	
1.3.2 依赖性注入.....10	
1.3.3 EJB 3.0 和 JPA.....10	
1.3.4 Web 服务.....11	
1.3.5 JSF.....11	
1.4 Java EE 5 相关的 JSR.....11	
1.4.1 JCP 介绍.....12	
1.4.2 Java EE 5 相关的 JSR.....13	
1.5 关于 GlassFish.....14	
1.5.1 GlassFish 项目概述.....15	
1.5.2 GlassFish 生态.....15	
1.5.3 GlassFish 的技术特性.....17	
1.6 GlassFish 和 Java EE 5 SDK.....18	
1.6.1 Java EE 5 SDK.....18	
1.6.2 基于 GlassFish 代码构建的 3 个产品.....20	
1.7 小结.....21	
第 2 章 开源和社区.....22	
2.1 开源概述.....22	
2.1.1 FOSS.....22	
2.1.2 开源带来的变革.....23	
2.1.3 开源软件许可类型.....24	
2.2 Java 开源与 GlassFish 开源.....24	
2.2.1 Java 的 GPL v2 开源.....24	
2.2.2 GlassFish 的双开源许可.....25	
2.3 GlassFish 社区.....26	
2.3.1 参与 GlassFish 社区.....26	
2.3.2 FishEye.....27	
2.3.3 Issue Tracker.....28	
2.3.4 GlassFish 子社区.....28	
2.4 java.net 社区.....29	
2.4.1 参与 java.net 社区.....30	
2.4.2 java.net 指导原则.....32	
2.5 SDN 和 SDN China.....33	
2.5.1 SDN.....33	
2.5.2 Sun 中国技术社区 (SDN China).....34	
2.6 小结.....35	
第 3 章 安装和构建 GlassFish.....36	
3.1 基于二进制文件安装 GlassFish.....36	
3.1.1 预备条件.....36	
3.1.2 安装二进制存档.....37	
3.1.3 启动服务器.....37	
3.2 基于源代码构建 GlassFish.....38	
3.2.1 构建预备工作.....38	
3.2.2 构建步骤.....39	
3.2.3 快速测试.....43	
3.3 使用 NetBeans(IDE)构建模块.....43	
3.4 小结.....45	



应 用 篇

第 4 章 管理控制台	49	5.2.2 asant 扩充的 Ant 任务	80
4.1 管理控制台概述	49	5.3 命令行架构	82
4.1.1 启动和登录	49	5.3.1 命令的定义	83
4.1.2 域的管理	50	5.3.2 命令的实现	84
4.1.3 管理控制台的组成	50	5.3.3 命令的交互	85
4.2 应用部署	52	5.4 命令定制	86
4.2.1 部署应用程序	52	5.4.1 增加命令	86
4.2.2 部署 Web 服务	53	5.4.2 定制远程命令	89
4.2.3 部署 MBean	54	5.5 小结	92
4.3 资源管理	55	第 6 章 Web 应用程序基础和 Servlet	93
4.3.1 JDBC 资源	56	6.1 HTTP 简介	93
4.3.2 JMS 资源	56	6.1.1 HTTP 的特点	93
4.3.3 JavaMail 会话	58	6.1.2 URL(统一资源定位符)	94
4.3.4 JNDI 资源	58	6.1.3 HTTP 请求(Request)	94
4.3.5 连接器	59	6.1.4 POST 和 GET 方法	96
4.4 系统配置	60	6.1.5 HTTP 响应(Response)	97
4.4.1 Java EE 容器	60	6.2 Web 应用程序基础	98
4.4.2 设置 HTTP 服务	63	6.2.1 Web 应用程序分类	98
4.4.3 ORB(对象请求代理)	66	6.2.2 Web 容器(Web Container)	98
4.4.4 线程池(Thread Pool)	67	6.2.3 Web 模块(Web Module)	99
4.4.5 系统 JMX 连接器	67	6.2.4 在 GlassFish 上打包、部署、 配置和运行	100
4.5 系统监控	67	6.2.5 动态重载	101
4.5.1 日志管理	68	6.3 Servlet 基本概念	102
4.5.2 监控组件和服务	69	6.3.1 什么是 Servlet	102
4.5.3 诊断服务	71	6.3.2 Servlet 的常用接口	102
4.5.4 JVM 和高级设置	71	6.3.3 Servlet 的生命周期	104
4.6 小结	72	6.4 Servlet 中的对象	105
第 5 章 GlassFish 命令行	73	6.4.1 请求对象	105
5.1 命令行工具 asadmin 的应用	73	6.4.2 响应对象	106
5.1.1 语法	73	6.4.3 Session 对象	107
5.1.2 asadmin 子命令	74	6.5 Filter(过滤器)	109
5.1.3 域(domain)子命令	75	6.5.1 定制 Filter	109
5.1.4 本地命令和远程命令	78	6.5.2 Filter 声明	111
5.1.5 asadmin 执行方式	79	6.5.3 Filter 的生命周期	113
5.2 asant 介绍	80	6.6 请求转发	114
5.2.1 asant 的配置	80		

6.6.1 include 方法	114	7.6.2 XML 标记库	140
6.6.2 forward 方法	115	7.6.3 SQL 标记库	141
6.7 生命周期事件	115	7.7 小结	142
6.7.1 监听器类型	116	第 8 章 JSF 编程	143
6.7.2 部署监听器	116	8.1 JSF 概述	143
6.8 Servlet 2.5 规范的新特性	117	8.1.1 RAD 与 Web 应用	143
6.8.1 Servlet 2.5 中的标注	117	8.1.2 什么是 JSF	144
6.8.2 web.xml 的改变	118	8.1.3 开发工具对 JSF 的支持	144
6.8.3 其他	119	8.2 创建 JSF 应用程序	145
6.9 小结	119	8.2.1 创建一个使用 JSF 框架的 Web 应用程序	146
第 7 章 JSP 编程	120	8.2.2 创建页面	147
7.1 JSP 概述	120	8.2.3 页面导航	148
7.1.1 JSP 发展历史	120	8.2.4 受管 Bean	149
7.1.2 JSP 的由来	120	8.2.5 部署运行	150
7.1.3 JSP 与 Servlet 的转换	120	8.3 完善 JSF 应用程序	151
7.1.4 JSP 的生命周期	123	8.3.1 示例分析	151
7.1.5 架构中的应用	123	8.3.2 创建验证器(Validator)	151
7.1.6 JSP 页面的构成	124	8.3.3 创建转换器(Converter)	152
7.2 JSP 指令	124	8.3.4 错误信息处理	152
7.2.1 page 指令	124	8.3.5 自定义错误信息	153
7.2.2 taglib 指令	125	8.3.6 自定义验证器	154
7.2.3 include 指令	126	8.4 JSF 中的重要组件	156
7.3 JSP 动作	126	8.4.1 UI 组件(UI Component)	156
7.3.1 <jsp:useBean>动作	127	8.4.2 绘制器(Renderer)	156
7.3.2 <jsp:setProperty>和 <jsp:getProperty>动作	128	8.4.3 验证器(Validator)	156
7.3.3 <jsp:include>动作	129	8.4.4 受管 Bean(Managed Bean)	156
7.3.4 <jsp:forward>动作	129	8.4.5 转换器(Converter)	157
7.3.5 其他 JSP 动作	130	8.4.6 事件(Event)和 监听器(Listener)	157
7.4 统一表达式语言(Unified Expression Language)	130	8.4.7 消息(Message)	157
7.4.1 隐含对象(Implicit Objects)	131	8.4.8 导航(Navigation)	157
7.4.2 对象的访问及其操作	132	8.5 JSF 的工作方式	158
7.5 表达式	133	8.5.1 配置 JSF 应用	158
7.5.1 立即表达式和延迟表达式	133	8.5.2 JSF 的生命周期	159
7.5.2 值表达式和方法表达式	134	8.6 重要的类和接口	162
7.5.3 自定义标记属性类型	135	8.6.1 FacesContext 类	162
7.5.4 禁用表达式	136	8.6.2 ExternalContext 类	162
7.6 JSP 标准标记库	137	8.6.3 Application 类	162
7.6.1 Core 标记库	138	8.6.4 UIComponent 接口	163



8.7 JSF 组件的层次结构和标识	163	10.2 安全的 Web 服务	204
8.7.1 JSF 组件的层次结构	163	10.2.1 在 GlassFish 上实现	
8.7.2 用标识符标识组件	165	· 全局的 WSS	205
8.8 小结	166	10.2.2 在 GlassFish 上实现	
第 9 章 AJAX 技术	167	局部的 WSS	206
9.1 AJAX 概述	167	10.3 JAXB	207
9.1.1 为什么会出现 AJAX	167	10.3.1 JAXB 历史	207
9.1.2 一个简单的 AJAX 验证的		10.3.2 XML 与 Web 服务	208
例子	167	10.3.3 JAXB 所描述的绑定	209
9.1.3 AJAX 技术的优势	175	10.3.4 JAXB 命令行工具	210
9.1.4 AJAX 技术的一些限制	176	10.3.5 JAXB 应用开发	210
9.2 Dojo Toolkit	176	10.4 StAX	212
9.2.1 用 Dojo Toolkit 实现 AJAX		10.4.1 拉式解析和推式解析	212
验证	176	10.4.2 StAX 接口	212
9.2.2 AJAX Widget	178	10.5 小结	217
9.2.3 用 Dojo 实现的 Fisheye		第 11 章 EJB 3.0 和 JPA 开发	218
List	178	11.1 EJB 3.0 特点综述	218
9.2.4 Dojo 的包	181	11.1.1 为什么要 EJB 3.0	218
9.3 jMaki	182	11.1.2 EJB 3.0 特点综述	218
9.3.1 容器——jMaki	182	11.2 Session Bean 开发示例	219
9.3.2 jMaki 在 NetBeans IDE 中的		11.2.1 定义业务接口	219
安装和使用	182	11.2.2 实现业务接口	220
9.3.3 用 jMaki 实现 Fisheye	183	11.2.3 打包	221
9.4 AJAX 与 JSF	184	11.2.4 发布	221
9.4.1 实例：自动完成文本字段	184	11.2.5 查找和调用 EJB 3.0	222
9.4.2 Sun BluePrints AJAX 组件	188	11.3 依赖注入示例	224
9.5 AJAX 与 GlassFish	189	11.3.1 引用 EJB	224
9.5.1 Phobos	189	11.3.2 引用 Resource	225
9.5.2 Woodstock	190	11.4 Entity 与 Java Persistence 示例	226
9.6 小结	191	11.4.1 定义 EJB 3.0 Entity	226
第 10 章 Web 服务	192	11.4.2 使用和管理 EJB 3.0	
10.1 Web 服务原理	192	Entity	230
10.1.1 Web 服务体系结构	192	11.4.3 Entity Manager 事务控制	232
10.1.2 Web 服务的实现技术	193	11.4.4 Persistence.xml	234
10.1.3 一个使用天气预报		11.5 小结	234
Web 服务的实例	196	第 12 章 GlassFish 的安全性	235
10.1.4 用 JAX-WS 实现		12.1 GlassFish 安全性基本概念	235
Web 服务	199	12.2 密码安全的管理	236
10.1.5 SAAJ	202	12.2.1 密码文件权限	236

12.2.2 主密码管理.....	237	12.8 小结	261
12.2.3 用户和资源密码管理	237	第 13 章 国际化和本地化	262
12.3 身份认证和授权	238	13.1 Java 国际化/本地化基本概念	262
12.3.1 身份认证	238	13.1.1 本地化环境(Locale).....	263
12.3.2 身份授权	238	13.1.2 字符编码(Character Encoding).....	265
12.3.3 管理控制台对授权的 管理	239	13.1.3 资源包(Resource Bundle).....	267
12.3.4 自定义的认证和授权 模块	242	13.2 Java EE 应用开发中的国际化	270
12.4 数字证书及工具	243	13.2.1 Servlet 的国际化	270
12.4.1 数字证书的基本概念	243	13.2.2 JSTL 中的国际化标记	271
12.4.2 GlassFish 证书文件	244	13.2.3 JSF 应用的国际化	276
12.4.3 更改 GlassFish 证书文件的 位置	244	13.3 GlassFish 的国际化特性	282
12.4.4 keytool 工具概述	245	13.4 小结	282
12.4.5 使用 keytool 生成 GlassFish 证书	246	第 14 章 Java EE 应用性能调优	283
12.4.6 使用 keytool 为数字证书 签名	246	14.1 Java EE 应用的性能调优简介	284
12.5 单点登录	247	14.1.1 什么是 Java 应用的性能	284
12.6 Web 服务安全性	248	14.1.2 Java EE 应用性能调优的 复杂程度	286
12.6.1 Web 服务安全性规范	248	14.1.3 Java EE 应用性能调优的 方法论	287
12.6.2 GlassFish 消息安全 提供者	249	14.2 被动的性能调优	288
12.6.3 GlassFish 消息保护策略	249	14.2.1 硬件层和操作系统层次的 监控	288
12.6.4 GlassFish 消息加密	252	14.2.2 Java 虚拟机层次的监控	294
12.6.5 WSS 消息安全实例	253	14.2.3 应用服务器层次的监控	298
12.7 GlassFish 应用程序安全设置	256	14.2.4 应用代码的跟踪	298
12.7.1 了解应用程序和系统 安全性	256	14.2.5 被动性能调优的案例分析	299
12.7.2 角色、主体以及 主体到角色的映射	257	14.3 主动的性能调优	302
12.7.3 领域设置	259	14.3.1 应用层的性能调优	302
12.7.4 设置 server.policy 文件	260	14.3.2 应用服务器的性能调优	305
		14.3.3 Java 虚拟机的性能调优	308
		14.3.4 操作系统的性能调优	308
		14.4 小结	309



开 源 篇

第 15 章 JMX 在 GlassFish 中的应用..... 313	第 16 章 自管理和 CallFlow346
15.1 JMX 概述..... 313	16.1 自我管理346
15.1.1 JMX 的背景..... 313	16.1.1 自我管理概述346
15.1.2 选择 JMX 的理由..... 314	16.1.2 管理规则与 JMX 消息 机制347
15.2 JMX 的基本概念..... 314	16.1.3 管理规则的定义和实例.....347
15.2.1 JMX 的 3 层架构..... 314	16.1.4 自我管理的事件(Event).....349
15.2.2 MBean 组件..... 315	16.1.5 创建管理规则350
15.2.3 MBean 服务器..... 317	16.1.6 自我管理的实现代码.....351
15.2.4 JMX 消息模型..... 319	16.2 自我管理应用开发实例.....351
15.2.5 代理服务(Agent Service)..... 320	16.2.1 创建自我管理应用353
15.2.6 连接器(Connector)..... 320	16.2.2 运行所创建的自我管理应用.....359
15.3 JMX 整合实例..... 321	16.3 CallFlow(调用流程监控).....363
15.3.1 运用 JMX 技术管理 Java 样例 ArcTest..... 321	16.3.1 CallFlow 功能概述.....363
15.3.2 MBean 的实现..... 322	16.3.2 CallFlow 架构.....364
15.3.3 JMX 代理的实现..... 324	16.3.3 CallFlow 应用.....365
15.3.4 运行实例..... 325	16.3.4 CallFlow 开发.....368
15.3.5 分布式管理应用的实现..... 326	16.4 小结371
15.4 Java EE 中的 JMX 应用..... 328	第 17 章 高性能的 HTTP 引擎—— Grizzly.....372
15.4.1 Java EE 管理模型(Management Model)..... 328	17.1 NIO 简介372
15.4.2 可管理对象..... 329	17.1.1 NIO 的基本概念373
15.4.3 监控接口..... 329	17.1.2 NIO 之前的 Server 程序的 架构375
15.4.4 MEJB 330	17.1.3 使用 NIO 来提高系统 扩展性376
15.5 GlassFish 对 JMX 的应用..... 331	17.1.4 使用 NIO 来制作 HTTP 引擎的最大挑战378
15.5.1 GlassFish 中的 MBean 331	17.2 Grizzly 简介379
15.5.2 AMX 概述 332	17.2.1 Grizzly 的基本架构.....379
15.5.3 动态客户代理(Dynamic Client Proxy)..... 333	17.2.2 源码阅读指南381
15.5.4 AMX 对 JMX 的扩展..... 337	17.3 Grizzly 的特点383
15.5.5 AMX 实现 338	17.3.1 如何处理慢速的连接.....383
15.6 开发 User MBean..... 343	17.3.2 如何避免内存泄漏.....387
15.6.1 User MBean 的生命周期..... 343	17.3.3 使用多个 Selector393
15.6.2 GlassFish 中 MBean 相关的 操作..... 343	17.3.4 Grizzly 其他的特点.....396
15.7 小结..... 345	

17.3.5 Grizzly 的性能.....	398	过程分析	432
17.4 小结.....	398	19.2.3 EJB 3.0 业务方法的调用	
第 18 章 异步请求处理和服务器推送技术	399	过程分析	434
18.1 异步请求处理.....	399	19.2.4 依赖注入的实现	434
18.1.1 Grizzly 异步请求处理的实现.....	401	19.2.5 回调和拦截器实现的分析.....	436
18.1.2 “新邮件提醒”异步请求处理.....	407	19.3 小结	438
18.1.3 Grizzly 异步请求处理的特点.....	411	第 20 章 内部持久性实现和 Java DB	439
18.2 服务器推送技术.....	411	20.1 基于文件的持久性实现.....	439
18.2.1 Comet 实现的分析	413	20.1.1 域配置文件 domain.xml	439
18.2.2 Comet 实例讲解——“聊天室”应用	416	20.1.2 读写 domain.xml	440
18.3 小结.....	419	20.2 基于数据库的持久性实现.....	441
第 19 章 EJB 容器实现	420	20.2.1 Java DB 的来源.....	441
19.1 GlassFish EJB 容器	420	20.2.2 Java DB 的特点	442
19.1.1 EJB 远程视图	420	20.2.3 使用 Java DB	443
19.1.2 EJB 容器的实现	422	20.2.4 Java DB 工具.....	444
19.1.3 EJB 远程方法调用过程分析.....	425	20.2.5 GlassFish 管理控制台有关 Java DB 的配置.....	445
19.2 GlassFish EJB 3.0 实现.....	429	20.2.6 GlassFish 中利用 Java DB 的持久性实现	448
19.2.1 EJB 3.0 中 Home 接口和 Component 接口的实现	429	20.2.7 Java DB 的应用	451
19.2.2 EJB 3.0 远程对象的查找		20.3 小结	457
		结束语	459
		附录 参考资源	461
		参考文献	462

基础篇

基础篇将从较高层面考察 GlassFish 项目，概要介绍 GlassFish 的背景、关联技术和标准，GlassFish 相关的开源社区以及如何构建 GlassFish

内容提要

一个健康的行业一定是被一系列制定良好的标准规范引导着。本篇第 1 章将介绍与 GlassFish 密切关联的 Java EE 5 标准，包括 Java EE 5 SDK 的组成、架构和新特性，该章接着将特别介绍制定 Java 标准的组织、工作流程和 Java EE 5 所包含的 JSR。第 2 章将介绍开源的基本概念以及 Java 主要社区的内容和组成。第 3 章将讨论如何构建 GlassFish，包括构建步骤和相关工具(Maven 和 NetBeans)的使用方法。

第 1 章 Java EE 5 和 GlassFish

Java EE 是最新的 Java 企业应用的工业标准, GlassFish 项目与 Java EE 平台紧密关联。本章将概要介绍 Java EE 5 平台、GlassFish 项目和二者的关系。关于 Java EE 5 平台, 本章先由 Java EE 5 的架构说起, 以便对 Java EE 5 平台的整体轮廓有些了解, 再进一步介绍 Java EE 5 中新的技术和特性, 然后将从 Java 标准制定的角度来了解 Java EE 5 平台。关于 GlassFish 项目的介绍, 将包括 GlassFish 的背景和历史以及 GlassFish 自身拥有的众多独到的特性。

本章重点:

- Java EE 5 架构
- Java EE 5 的新特性
- GlassFish 的技术特性

1.1 Java EE 5 概述

Java 企业平台一直在迅速地发展着, 每隔 2~3 年就有新的版本推出, 最新的 Java 企业平台是 Java EE 5, 见图 1-1。

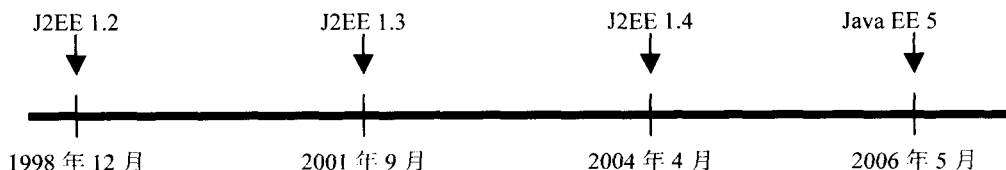


图 1-1 Java 企业平台的演变

Java EE 5 规范连同其 SDK 于 2006 年 5 月发布。Java EE 5 是 Java 企业开发平台对两年前发布的 J2EE 1.4 所做的一次重大升级。与 J2EE 1.4 相比, 可以发现升级到 Java EE 5 所带来的变化是巨大的, 这些变化体现了对 J2EE 不足之处的改进, 也体现了近年来 Java 企业应用技术的热点和趋势。

J2EE 自 1998 年问世以来, 在其强大功能被认可并广泛使用的同时, 也因为繁琐的开发步骤、依赖性强的调试开发环境和由此而产生的较高的学习门槛, 使不少开发人员, 尤其是开发中小企业应用的开发人员望而却步。在此背景下, 大量轻量级的框架被广泛接受和使用, 如 Spring 和 Hibernate。针对这些问题, Java EE 5 将简化开发当作首要使命, 吸纳并扩展了各类流行的轻量级框架的思想, 并将之规范化和标准化。此外, 随着 SOA