



Java 服务器 高级编程

(英) Danny Ayers 等著

曾国平 冯伟 王占峰 等译

Wrox程序员参考系列

Java服务器高级编程

(英) Danny Ayers 等著

曾国平 冯 伟 王占峰 等译



机械工业出版社
China Machine Press

本书详细深入地讲述了servlet和Web编程，展示了Java服务器端专业编程的新特性，内容覆盖XML、EJB、JBDC、JND、JavaMail、RMI、CORBA、Jini/JavaSpace和Apache，为在Java平台上编写n-层应用提供了最新的程序设计理论和技术，是一本用servlet、JSP、Jini/JavaSpace等新技术创建n-层Java应用的专业书籍。本书内容详尽，语言精练，编程技术先进，适用于有一定Web编程经验并熟悉Java语言 and 核心API的开发人员；同时也能为想要理解和掌握Java服务器端编程的读者提供帮助。

本书提供了5个编程实例和大量的源程序，所有源代码都可以从Internet上免费下载，其地址是<http://www.wrox.com>，同时也可以从这个站点获得对本书的技术支持。

Danny Ayers, et al: Professional Java Server Programming.

Authorized translation from the English language edition published by Wrox Press.

Original copyright © 2000 by Wrox Press. All rights reserved.

Chinese simplified language edition published by China Machine Press.

本书中文简体字版由英国乐思出版公司授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-2000-2197

图书在版编目（CIP）数据

Java服务器高级编程 /（英）艾尔斯（Ayers, D.）等著；曾国平等译. —北京：机械工业出版社，2001.5

（Wrox程序员参考系列）

书名原文：Professional Java Server Programming

ISBN 7-111-08857-3

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

中国版本图书馆CIP数据核字（2000）第19383号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：华章

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2001年5月第1版第1次印刷

787mm × 1092mm 1/16 · 57印张

印数：0 001-5 000册

定价：92.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译者序

随着因特网和万维网的发展，我们可以在世界各地共享信息，为客户提供服务，甚至进行电子商务活动。作为Web应用开发者，面临的挑战是需要进一步推动它的发展。随着技术的不断更新，需要开发出更快、更轻巧、更强壮的能通过Web进行发布的网上应用。在这方面，已经出现了许多功能强大的工具。过去，如果想通过Web发布一个数据库驱动的应用给客户，你只能写一些CGI程序处理格式化的数据并返回结果；但最近几年，出现了大量的新技术，像ISAPI和NSAPI这类API，它们比CGI的效率更高，但复杂性也随之提高；像Active Server Pages（即ASP）和PHP这类脚本解决方案有助于简化Web应用开发。这些方案已经为处理逻辑和数据资源通信的组件提供了Web接口，这是一种创建Web应用的集成技术。为了充分发挥这些技术的潜力和提供更好扩展、更好移植的服务器端解决方案，充分体现当今Web应用中流行的分离处理内容显示、业务逻辑和数据源的标准技术，Sun公司开发了一种新的称为servlet的技术；Java servlet非常高效，具有平台无关性，它为Web应用提供了一个面向对象的、扩展性好的中间层，能够访问像JNDI、JDBC、RMI这类企业版Java的API函数和Enterprise JavaBeans（EJB）。

本书将讨论Java servlet以及同服务器端相关的Java技术，将讲解如何使用最新的称为JavaServer Pages（JSP）的服务器端Java技术；讨论了如何将servlet融合到应用中，阐述了数据库访问、连接池（pool）、邮件的工作方式以及servlet、applet之间的通信；介绍了XML技术和用Java处理XML的方式；讨论了n-层应用的设计、国际化以及动态构造运行时行为的servlet，给出了如何利用JNDI命名和访问目录服务，描述了EJB的设计方式；最后，本书介绍了Jini和JavaSpace，展示了这些新技术的一些应用前景。

本书提供了Java服务器端编程的最新技术，同时还附有大量的代码，提供了丰富的Web资源信息，最后还给出了内容详尽的附录，附录包括了HTTP协议、几个servlet引擎的安装建议和在全书中所使用的企业版类和接口的详细API参考。本书内容丰富，编程技术先进，对于开发Web应用的程序设计人员具有很高的参考价值；同时也能为想要理解和掌握Java服务器端编程的读者提供帮助。

参加本书翻译工作的人员都是多年从事计算机网络、数据库开发和翻译工作的专业人员，完成本书主要翻译工作的人员有曾国平、冯伟、王占峰、赵胜龙、王志军、黄永宏、杜皎、陈连俊；参加本书翻译和录校工作的还有付增少、张景生、赵祥龙、曹丽云、刘俊杰、赵宇龙、陶雪鹤、杨成、徐一鸣、易崇、懂泽明、罗俊、赵燕、蒋泽云、黄显春、杨华、陈清明、李斌、帅旭等人。

鉴于时间仓促，加之译者水平有限，书中难免有翻译不妥之处，恳请广大读者和同行批评指正。

2000年10月

作者简介

Danny Ayers

性格温和的Danny Ayers在维护位于Derbyshire Peak District的High Peak College（Derby大学）的网络期间，就已成为了自由作家和顾问工程师。他的兴趣包括神经网络、木雕、贝斯音乐。他的第一个PC是PET 2001 Series（8K）。可以通过danny_ayers@yahoo.com同他联系。

Hans Bergsten

Hans作为软件开发者，他拥有近20年的开发经验。在为Ericsson工作的17年时间中，他为每种计算机都开发过软件系统，这些计算机包括IBM的大型机、DEC微机和UNIX工作站及服务器以及PC。他在过去10年的时间里一直从事面向对象概念方面的工作，使用了Simula、Smalltalk、C/C++和Java。他从80年代末期起就一直开发数据库和基于网络的应用，并且当Java在1995首次公布时就开始对它进行研究。

在1997年，Hans创建了Gefion Software，以进一步发展他在基于网络、平台无关的软件系统方面的思想。Gefion Software是一种Java服务器端技术，是以下三种基于Java servlet的流行产品的先驱：用于简化开发动态Web站点的组件套件（InstantOnline Basic）、用于Netscape Web服务器的servlet引擎（WAICoolRunner）和用servlet支持的Java Web服务器（Lite WebServer）。

Hans是Servlet API和JavaServer Pages(JSP)规范工作组的成员，同时也是Java和站点设计方面的正规作家。

Mike Bogovich

Mike S. Bogovich是位于Illinois州Oak Brook城的BALR Corporation的顾问兼技术经理，在长达10年的时间中，他一直从事面向对象语言和关系数据库的工作。他是Chicagoland Java User Group的成员。

Jason Diamond

Jason Diamond（jason@injektilo.org）是一个软件工程师和音乐家，目前正在阳光明媚的CA的Oceanside享受生活。

Matthew Ferris

Matthew E. Ferris是位于Illinois州Oak Brook城的BALR Corporation的顾问，他在近5年的时间中一直从事面向对象软件的工作。他是Sun认证的Java程序员和Chicagoland Java User Group的主席。

Marc Fleury

Marc Fleury是硅谷中CA的Fostert City的独立承包商，Marc的研究集中于服务器端Java和电子商务的实现。在他早期的生活中，他获得了数学和理论物理学学位，结束了在物理博士方面的漫长学习。当他接触到软件bug问题时，他获得了新生。Marc加盟了Sun Microsystems，在那里他作为Java传播者和SAP专家度过了3年时间，然后他独立出来进行工作。在他的业余时间中，Marc喜欢参与各种开放源代码的项目，比如：EJBoss、Linux的EJB服务器。

Paul Houle

Paul最近从Cornell大学获得了博士学位，并且正在Dresden完成博士后研究。他喜欢UNIX、Java和他的两只猫。他正在进入Web顾问行业，并且目前正在研究大型超文本的智能Web蜘蛛和内容分析。

Piroz Mohseni

Piroz Mohseni是Bita Technologies的主席，Bita Technologie的工作集中在通过使用有效的技术来改进事务。他感兴趣的领域包括企业Java、XML和B2B商业应用。在这之前，他是Technical Staff的成员，同时还是Lucent Technologies Bell Laboratories中100%纯Java应用服务器的主要建筑师。Mr. Mohseni定期为各种Java出版物撰写文章，并且偶尔在会议上发言。可以通过pmohseni@iname.com同他联系。

Andrew Patzer

Andrew Patzer是New Resources Corporation的高级顾问，同时也是公司电子商务实践方面的执行经理。他是一个获得了合格证书的Java程序员，并且在过去的两年中一直从事开发服务器端Java应用的工作。Andrew已经出版了分布式体系结构方面的白皮书，并且向用户群和会议进行了多次演示。除Java之外，Andrew也十分喜爱从事Linux操作系统方面的工作，并且还学习过Tcl/Tk和Python。Andrew同他的妻子Beth及女儿Ashley、Emily一起居住在美国Wisconsin州的Oak Creek城，可以通过apatzer@usa.net同他联系。

Ron Phillips

Ron Phillips从事设计和开发商业软件工具的工作已超过10年，他尤其擅长于使用商业4GL环境、编程语言和软件设计工具。他在IT社团方面是很活跃的，并在美国和欧洲的几个工业会议上发表过演说。他目前的研究工作集中在分布式对象、n-层系统和移动代理。

Sing Li

Sing首次受到计算机bug的刺激是在1978年，并伴随着微处理器革命一起长大。他的第一台PC是99美元的自己组装的COSMIC ELF计算机，具有256字节内存和1比特LED显示。在20年中，Sing一直是活跃的作家、顾问、演说家、讲师和企业家，他的经验丰富，跨越了分布式体系结

构、多层Internet/Intranet系统、计算机电话学、呼叫中心技术和嵌入系统。Sing过去参加了几个Wrox项目，并且从这些项目的第一个 α 发行版开始一直从事Java和Jini的工作（并一直撰写这方面的文章）。他是Jini社团的一个积极参与者。

Krishna Vedati

Krishna Vedati是ValiCert Inc.的高级软件工程师。在工作于ValiCert之前，他是Rational Software Corporation的高级软件工程师，为X-windows和Web服务器设计和开发了负载测试工具。Vedati也参与了Wrox Press出版的《Beginning Linux Programming》一书的编写工作，并为Linux社团讲授Tcl和PGP方面的入门课程。他感兴趣的技术领域包括Internet安全、网络代理、Internet协议、脚本编程语言和图形用户界面。可以通过krishnav@valicert.com或者kvedati@yahoo.com同他联系。

Mark Wilcox

Mark是North Texas大学的Web管理员，也是由Wrox Press出版的《Implementing LDAP》的作者，还是Netscape的ViewSource杂志的正规专栏作家。

Stefan Zeiger

从1997年开始，Stefan Zeiger一直作为自由Java程序员进行工作，并从1996年开始在Darmstadt的Technical大学学习计算机科学。他是NetForge Web服务器软件和深受欢迎的在线servlet入门教程《Servlet Essential》的作者。

前言

本书展示了Java Servlets和JavaServer Pages(JSP)如何以可移植、安全、高效的方式向Web客户提供动态和可定制的内容。

由于Web应用实际上只相当于它们提供的功能操作，所以我们也介绍了将servlet和JSP对接到一起的Java Enterprise API。为了展示编写Java应用的实际情况，我们给出了5个正在使用的servlet和JSP的事例研究。

本书适用的读者

我们无需再介绍Java语言了。从发布那天起，Java就已被大家所熟知。本书不仅包括Java语言，而且还有Java 2平台，以及支持各种应用的类库，同时也说明了它们影响Web编程的方式。

Java 2平台为编出可移植的、面向对象的、安全的Internet应用提供了一条快速成熟的途径。在过去的两年里，Java对应用的支持获得了快速增长。你现在可以开发出以下动态Web服务器应用：

- 提供HTML、XML和GIF服务。
- 分离界面、逻辑和数据。
- 跟踪客户会话。
- 比CGI更容易扩展。
- 提供同数据库、其他Java应用、CORBA、目录和邮件服务的接口。
- 利用应用服务器中间件提供事务支持。

本书所讨论的API得到了广泛的工业支持，它们已被JavaSoft及其他专业合作伙伴所开发。其结果是，Java的这种可移植代码和开放API所带来的革命引起了现有产品的一些变化（包括数据库、应用、邮件和Web服务器）。能够在服务器上运行Java应用的产品的丰富性使得这项技术成为了快速发展和十分具有竞争力的市场，但同时通过规范、标准API和类库等手段保持了必要的兼容性，这使得服务器端Java成为了一个非常激动人心的领域。

本书适用于那些有一定Web编程经验的专业开发人员，同时还假定他们熟悉Java语言和核心的API，可参阅其他相关内容的书籍获得这些知识。我们将回顾服务器端编程方面不断成熟的主要领域。

本书内容

简要地说，本书讲述servlet和Web编程，其重点集中在输入/输出流、请求和响应方面。servlet是任何能使用HTTP协议的客户端同Java代码所控制的数据或服务之间进行通信的桥梁。

本书的结构如下：

- 作为一本servlet编程指南，讨论了类、servlet的生命周期、错误处理、会话和servlet环境，并列举了一些例子。
- 介绍了JSP，说明了它与servlet的关系，给出了使用标记（tag）的方式，并创建了bean（可重用对象）用于封装事务逻辑和保持页面的简洁。
- 在前面讨论的基础上，接下来的几章即讨论了如何将servlet融合到应用中，还用了一些章节讨论数据库访问、连接池（pool）、邮件的工作方式、servlet（小服务程序）和applet（小应用程序）以及servlet和servlet之间的通信。
- 在Web上使用XML（可扩展标记语言）是这些应用的即将到来而又至关重要的部分，我们将简介用Java处理XML和一个向浏览器提供XML数据的实例。
- 接下来的章节讨论了n-层（n-tier）应用的设计、国际化以及根据输入动态地构造运行时行为的servlet，如何利用JNDI来命名和访问目录服务，讨论了EJB的设计并给出了一个servlet spider研究实例。
- 最后，我们介绍Jini和JavaSpace，并展示了这些新技术的一些应用前景。

附录给出了编程方面的补充材料、关于servlet引擎的安装建议和全书所使用的企业版类和接口的详细API参考。

使用本书的准备工作

本书代码在Java 2 Platform Standard Edition SDK上（JDK 1.2.1）通过测试。另外，我们还需要安装其他一些库，这些库及版本如下：

- servlet。我们使用JSDK 2.1和JSWDK 1.0 EA，这里包含的servlet引擎适用于大多数软件测试。
- JSP 1.0。我们使用JavaSoft公司的JSP 1.0参考实现的发行版。
- XML语法分析器。我们使用Sun的ProjectX类（<http://java.sun.com/xml>）。
- GIF编码器。可以从<http://www.acme.com> 获得。
- 数据库。任何JDBC/ODBC兼容数据库，我们使用Access和MySQL。
- 邮件服务器。同JavaMail 1.1.2和JAF 1.0.1连接在一起的任何SMTP邮件服务器。
- JNDI/LDAP服务器。Netscape目录服务器、JNDI 1.2 Beta(1.1.2)。
- 国际字体。
- Jini和JavaSpace。我们使用这两个套件的1.0版。
- Enterprise Java Beans 1.1。
- Web Servers/Servlet Runners。使用Apache 1.3.6、Apache JServ1.0、JRun2.3、Servlet Exec和Java Web Server 2.0。

如果计算机与网络相连（即能够通过本地浏览器看到<http://localhost>），各章的大多数代码都可在单机上运行。Jini章节实际上需要3台联网的机器以使代码能够完成。同时请注意，Jini代码还需要UDP支持，一些Windows用户需要更新Winsock的版本。

本书所有源代码都可以从如下地址下载：<http://www.wrox.com>。

约定

- 粗体代码段属于Java类库的定义。
- 键盘控制键表示为：Ctrl-A。
- 文件和代码表示为：doGet()
- 用户界面上的URL和文本表示为：<http://www.wrox.com>。

联系方法

我们勤奋工作是为了尽可能让本书对你有用，因此我们十分希望知道你的想法。我们总是热切希望了解你需要掌握的东西。我们衷心希望得到反馈，欢迎你提出意见，以便今后的工作。有什么想法，请通过如下联系方式告诉我们：

feedback@wrox.com或者<http://www.wrox.com>

目 录

译者序	
作者简介	
前言	
第1章 Web应用开发	1
1.1 Web结构	1
1.1.1 Web浏览器	2
1.1.2 Web服务器	3
1.2 n-层应用	4
1.3 Web应用的体系结构	5
1.3.1 收集数据	6
1.3.2 向Web服务器发送请求	6
1.3.3 执行服务器脚本	7
1.3.4 将结果返回浏览器	8
1.4 开发服务器应用	8
1.4.1 服务器端的技术	8
1.4.2 Java的特点	10
1.5 小结	11
第2章 servlet简介	12
2.1 servlet容器	12
2.1.1 servlet API	14
2.1.2 servlet的构造	18
2.2 简化HTML的生成过程	21
2.3 关于性能的几点注释	23
2.4 处理表单数据和发送邮件	23
2.4.1 建立HTML表单	24
2.4.2 编写servlet	25
2.5 小结	29
第3章 错误处理和事件记录	30
3.1 异常	30
3.1.1 运行时刻和编译时刻错误	30
3.1.2 在Java中处理异常	31
3.1.3 在servlet中处理异常	33
3.1.4 发送HTTP错误	38
3.1.5 记录事件	39
3.1.6 使用servlet异常	40
3.2 小结	41
第4章 会话与会话跟踪	42
4.1 会话	42
4.2 使用传统方法进行会话跟踪	43
4.2.1 重写URL	43
4.2.2 隐藏表单字段	43
4.2.3 cookies	44
4.3 使用Java Servlet API进行会话跟踪	44
4.3.1 获得一个会话对象	45
4.3.2 读写会话数据	45
4.3.3 使会话对象失效	46
4.3.4 一个使用会话的简单购物车	46
4.3.5 方法的选择	50
4.4 小结	50
第5章 使用servlet环境	51
5.1 “聊天”应用概述	51
5.2 servlet环境的配置	55
5.2.1 servlet环境的唯一URI路径前缀	55
5.2.2 servlet环境的servlet名称和URI路径映射	56
5.2.3 优点	56
5.3 servlet环境状态	58
5.3.1 初始化servlet环境状态	58
5.3.2 访问环境状态	59
5.4 改变环境状态	66
5.5 运行聊天应用	68
5.5.1 配置servlet引擎	68
5.5.2 添加聊天室	70
5.5.3 聊天	70

5.6 小结	71	7.7 小结	117
第6章 生成动态内容	73	第8章 连接数据库	118
6.1 MIME类型	73	8.1 向servlet增加数据	118
6.1.1 Image/jpeg	73	8.1.1 Java数据库连接性	118
6.1.2 Image/gif	74	8.1.2 影片数据库	123
6.1.3 Image/png	74	8.2 用预备式语句优化性能	131
6.2 返回二进制数据	74	8.3 合并Connection对象	135
6.3 创建GIF图像	75	8.4 小结	140
6.3.1 创建一个不显示的图像	75	第9章 连接池	142
6.3.2 对图像进行编码	76	9.1 池化资源	142
6.4 统计图servlet	77	9.2 传统的连接池	143
6.4.1 实用类	77	9.2.1 ConnectionPool类	144
6.4.2 servlet本身	80	9.2.2 PoolManager类	150
6.4.3 图形环境	81	9.2.3 例子客户	157
6.4.4 创建颜色	82	9.3 JDBC 2.0 Standard Extension连接池	158
6.4.5 绘制饼形图	84	9.3.1 主要概念	159
6.4.6 绘制条形图	85	9.3.2 带有连接池的DataSource	160
6.4.7 创建一个嵌入图像的HTML页面	87	9.4 将我们的连接池修改得像JDBC 2.0 SE	162
6.4.8 反向链接到页面	90	9.4.1 ConnectionWrapper类	162
6.5 小结	91	9.4.2 ConnectionPool的修改	163
第7章 JavaServer Pages介绍	92	9.4.3 例子客户	164
7.1 体系结构概述	92	9.5 小结	165
7.2 一个简单的JavaServer Page	93	第10章 servlet链接	166
7.2.1 JavaServer Page的编程元素	94	10.1 使用别名链接servlet	166
7.2.2 JSP伪指令	95	10.1.1 配置servlet引擎	167
7.2.3 JSP声明	96	10.1.2 使用别名的servlet链接	170
7.2.4 JSP scriptlet	96	10.2 使用MIME类型过滤servlet结果	174
7.2.5 JSP表达式	98	10.2.1 配置Web服务器	174
7.2.6 编写JSP页面的代码	98	10.2.2 使用MIME-Type过滤内容	174
7.3 使用具有JSP的JavaBean组件	98	10.3 小结	177
7.4 文件浏览器	103	第11章 servlet通信	178
7.5 浏览和查询数据库	105	11.1 从applet中调用servlet	178
7.5.1 询问正确的问题	106	11.2 RequestDispatcher 接口	184
7.5.2 查询数据库	107	11.3 小结	186
7.5.3 浏览数据库	112	第12章 利用servlet进行分布式计算	188
7.5.4 混合servlet和JSP	116	12.1 远端方法调用	188
7.6 用applet和bean增强用户界面	117	12.1.1 远端对象	189

12.1.2 stub 和skeleton	191	14.5.3 注释	223
12.1.3 RMI注册表	191	14.5.4 包含/忽略	223
12.1.4 Java客户	191	14.5.5 数据处理指令	224
12.1.5 实现RMI的步骤总结	192	14.5.6 码元声明	224
12.2 Java 2的安全机制	192	14.5.7 属性字声明	226
12.3 建立一个远端对象	193	14.5.8 实体声明	226
12.3.1 创建远端接口	194	14.5.9 符号声明	228
12.3.2 实现远端接口	194	14.6 对有效XML进行分析	229
12.3.3 编译并运行代码	195	14.7 关于分析的更多内容	231
12.3.4 创建RMI客户应用	195	14.8 SAX	232
12.4 建立RMI客户的servlet	196	14.9 DOM	236
12.4.1 安装安全管理器	196	14.10 Pizza servlet	238
12.4.2 获得远端接口的引用	196	14.11 小结	249
12.4.3 调用远端方法	197	14.12 XML的资源	250
12.5 小结	198	第15章 站点Weeds of El Limon 2	253
第13章 JavaMail和servlet	199	15.1 站点介绍	253
13.1 安装JavaMail	199	15.2 工作原理	256
13.2 提供者注册表	200	15.3 不友好的网络	258
13.3 使用JavaMail	200	15.4 工具	259
13.3.1 JavaMail会话	201	15.5 XML、SQL和Java对象	259
13.3.2 构造一个报文	202	15.5.1 XML文件	259
13.3.3 传输服务	203	15.5.2 XML到SQL的映射	262
13.4 SendMail servlet	204	15.6 Weeds类	268
13.5 存储转发服务	206	15.6.1 将关系数据库映射到对象	270
13.6 WebMail servlet	208	15.6.2 创建并装入数据库	273
13.7 小结	213	15.7 URL重写和文件的布局	275
第14章 XML介绍	214	15.7.1 Web浏览器中的网页	276
14.1 什么是XML	214	15.7.2 服务器上实际有些什么	276
14.1.1 可扩展性	215	15.8 JSP和JavaBean	278
14.1.2 可结构化	216	15.8.1 Index.jsp和Common.jsp	280
14.1.3 可确认性	216	15.8.2 Weed.jsp	281
14.2 XML像什么	216	15.8.3 Error.jsp	283
14.3 用Java进行分析	217	15.9 bean	284
14.4 有效的XML文档	220	15.10 从数据库中提取图像	287
14.5 DTD像什么	221	15.10.1 ViewWeed类	287
14.5.1 名称	222	15.10.2 InsertImages类	288
14.5.2 数据类型	222	15.11 小结	289

第16章 错误跟踪器事例研究	290	第18章 从CGI转移到servlet	357
16.1 工程介绍	290	18.1 概述	357
16.2 开发过程的选择	291	18.2 获得环境变量	357
16.3 合理统一过程	291	18.3 使用cookie	359
16.3.1 开始阶段	292	18.3.1 用Java来处理cookies	360
16.3.2 详细描述阶段	292	18.3.2 举例——用cookies创建一个购物车	361
16.3.3 构建阶段	293	18.4 正则表达式	366
16.3.4 转移阶段	294	18.5 小结	372
16.3.5 RUP内的活动	294	第19章 国际化Web站点	374
16.4 错误跟踪器的开始阶段	295	19.1 字符集	374
16.5 下一步工作	301	19.2 字符编码	374
第17章 错误跟踪器事例研究：详细描述、 构建和转移阶段	303	19.3 在Java核心API中支持的字符编码	375
17.1 详细描述	303	19.4 内容协商	376
17.1.1 原型设计	303	19.5 Servlet API中支持的字符编码	381
17.1.2 低精度的用户界面原型设计	303	19.5.1 国际化servlet	381
17.1.3 技术选择——信息输出	304	19.5.2 返回文本文件的变体	386
17.1.4 技术选择——对象的分布	306	19.6 小结	393
17.1.5 技术选择——报文处理	308	第20章 智能servlet	395
17.1.6 定义应用的体系结构	309	20.1 混合的网络配置	395
17.1.7 充实对象模型	311	20.2 混合的数据库平台	396
17.2 质量工程	318	20.3 混合的处理和操作系统平台	396
17.3 构建	321	20.3.1 servlet的解决方案:one-to-one	398
17.3.1 用IDL定义组件	325	20.3.2 servlet链接的解决方案	399
17.3.2 构建业务模型组件	326	20.3.3 智能servlet解决方案	399
17.3.3 业务模型对象的定义及持续性	328	20.4 请求的处理过程	399
17.3.4 建立映射组件	331	20.5 进程授权	402
17.3.5 构建业务逻辑组件	335	20.6 在企业中实现“智能servlet”	403
17.3.6 构建用户界面组件	340	20.6.1 servlet作为任务管理者	403
17.3.7 构建错误跟踪器的servlet	342	20.6.2 授权请求的力量	405
17.3.8 建立错误跟踪器	351	20.6.3 将进程与servlet分离	405
17.4 转移阶段	351	20.6.4 尽可能地抽象servlet和lancester类	405
17.4.1 配置运行时刻选项	351	20.7 智能servlet的扩展性	405
17.4.2 配置和建立数据库	351	20.7.1 用Hashtable排除方法参数	405
17.4.3 运行错误跟踪器	352	20.7.2 不需重新编译servlet而添加功能	406
17.4.4 运行Web客户	355	20.7.3 使用预先做好的HTML文件进行 动态更改	406
17.5 小结	356	20.8 小结	409

第21章 使用JNDI进行服务器编程	411	23.1 Web需要可移植性	459
21.1 命名和目录服务	411	23.1.1 Java和Web	459
21.1.1 网络就是计算机	411	23.1.2 服务端移植性	460
21.1.2 命名服务	412	23.1.3 用于Web应用的服务	462
21.1.3 目录服务	412	23.1.4 Enterprise JavaBeans和WebOS	465
21.2 为什么使用LDAP	413	23.2 EJB规范	465
21.3 介绍JNDI	415	23.2.1 在EJB规范中的角色	466
21.4 使用JNDI	417	23.2.2 indirection观点	468
21.4.1 安装JNDI	417	23.2.3 三种规约定义一个容器	473
21.4.2 JNDI程序包	418	23.3 会话bean	474
21.5 JNDI服务提供者和AKA JNDI驱动 程序	418	23.3.1 引用和句柄是活动的	476
21.5.1 如何得到JNDI服务提供者	419	23.3.2 会话bean的生命周期	477
21.5.2 开发自己的服务提供者	419	23.3.3 事务性和非事务性方法	478
21.6 基本的LDAP操作	420	23.3.4 钝化和激活	479
21.6.1 标准的LDAP操作	420	23.4 实体bean	480
21.6.2 用JNDI连接到LDAP服务器	420	23.4.1 实体bean的持续性	480
21.6.3 绑定	421	23.4.2 实体bean的身份	483
21.6.4 检索LDAP服务器	423	23.4.3 实现主关键字	484
21.6.5 确定LDAP的检索范围	423	23.4.4 搜索器	485
21.6.6 检索操作	424	23.4.5 激活和钝化	486
21.7 LDAP服务器的修改	430	23.4.6 实体bean的生命周期	487
21.7.1 增加条目	430	23.5 事务管理	489
21.7.2 修改条目	433	23.6 调度描述符	491
21.7.3 删除条目	435	23.7 小结	495
21.8 小结	436	第24章 索引和检索	496
第22章 使用LDAP和Java	437	24.1 spider	496
22.1 在LDAP里存储和取回Java对象	437	24.1.1 信息搜集	497
22.2 分散存储和集中存储	439	24.1.2 附加信息	498
22.3 使用JNDI/LDAP改进顾客服务	440	24.1.3 分析器	498
22.3.1 介绍基本对象	441	24.1.4 数据库	502
22.3.2 获得数据	441	24.2 GUI	511
22.3.3 屏幕镜头	442	24.3 开始爬行	518
22.3.4 LDAP的目录信息树	444	24.4 我, 机器人	520
22.3.5 顾客服务代码	447	24.5 客户	520
22.4 小结	458	24.6 优化和扩展	526
第23章 Enterprise JavaBeans	459	24.7 链接	527
		第25章 未来服务器Jini和JavaSpace	528

25.1 警告：即将发生的变化	528	26.2.4 read()方法	587
25.2 展望未来：Jini事例研究	528	26.2.5 take()方法	587
25.2.1 在线贷款许可Web站点	528	26.2.6 通知的注册：Jini分布式事件	588
25.2.2 仔细考虑的问题	530	26.2.7 条目提交的优化	589
25.3 驱动分布式设备	530	26.2.8 用JavaSpace实现工作流程	590
25.3.1 分布式计算	530	26.2.9 将并行性加入到JavaSpace系统中	591
25.3.2 自然不协调性	531	26.3 将JavaSpace添加到我们的事例研究中	592
25.4 计算上的革命	532	26.3.1 系统中的服务	592
25.4.1 集中式控制与动态分组分裂	532	26.3.2 系统中的JavaSpace	593
25.4.2 Jini是我们共同工作的方式！	533	26.3.3 系统的示意图	594
25.5 剥离Jini层	538	26.3.4 精炼我们的设计	595
25.5.1 发现和加入协议	538	26.3.5 准备实现	599
25.5.2 查找服务——代理	541	26.3.6 JavaSpace小结	600
25.5.3 Sun提供的实现	552	第27章 编写基于Jini的站点的代码	601
25.6 投入Jini的海洋	557	27.1 模拟业务逻辑	601
25.6.1 为探测准备系统	558	27.2 本地优化	601
25.6.2 接触REGGIE——一个参考查找 服务	560	27.3 附加模块	601
25.6.3 其他基本的Jini部分	566	27.4 编写定制条目	602
25.6.4 编写三个Jini服务的代码	567	27.4.1 UserEntry	602
25.6.5 编写一个Jini客户的代码	575	27.4.2 UserEntryBean	603
25.6.6 测试自己的djinn	577	27.5 编写ServerNode	604
25.6.7 随着Jini技术一同前进	579	27.6 编写JSpaceServlet	608
第26章 同JavaSpace一起工作	581	27.7 编写服务节点	614
26.1 什么是JavaSpace	581	27.7.1 NameVerifierNode	614
26.1.1 JavaSpace是Java接口	581	27.7.2 CreditCheckerNode	616
26.1.2 JavaSpace是Jini服务	582	27.7.3 HumanVerificationNode	618
26.1.3 JavaSpace像查找服务	582	27.8 编写测试工具	621
26.1.4 JavaSpace是真正的对象数据库	583	27.8.1 DataPut——一个空间写入器	622
26.1.5 JavaSpace是你的MOM	583	27.8.2 流量生成器	624
26.1.6 JavaSpace是steroid上的一种排队 服务	585	27.8.3 DumpSpace——一个空间转储器	625
26.1.7 看看飘在天空中的云	585	27.8.4 编译代码	627
26.2 个人的JavaSpace构造集合	585	27.9 创建JavaSpace	628
26.2.1 和你需要的空间一样多	586	27.9.1 mahalo——Sun的事务服务	628
26.2.2 JavaSpace里的条目	586	27.9.2 outrigger——Sun的JavaSpace服务	628
26.2.3 write()方法	586	27.9.3 演奏JavaSpace交响乐	629
		27.10 设置系统	630
		27.10.1 在三个机器组成的网络上的实现 步骤	630

27.10.2 一步一步的启动过程	632	附录H JRun的配置	742
27.10.3 测验我们的分布式系统	634	附录J JSDK API 参考	748
27.10.4 我们已经成功地完成了的工作	635	附录K JavaServer Pages API参考	760
附录A HTTP	636	附录L JNDI API参考	765
附录B Java对象流及串行化	652	附录M 核心JavaMail/JAF API参考	803
附录C 密码术和servlet	669	附录N 核心Jini API 参考	832
附录D LogWriter类	699	附录O JavaSpaces API 参考	852
附录E UML符号	701	附录P Enterprise JavaBeans API 参考	854
附录F JServ的配置	705	附录Q JDBC API参考	862
附录G ServletRunner和Java Web Server的 配置	737	附录R 支持与勘误表	890