

# 《Apache Tomcat 6高级编程》

## 图书基本信息

书名：《Apache Tomcat 6高级编程》

13位ISBN编号：9787115193520

10位ISBN编号：7115193525

出版时间：2009-3

出版社：人民邮电出版社

作者：Vivek Chopra, Sing Li, Jeff Genender

页数：560

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

# 《Apache Tomcat 6高级编程》

## 前言

本书讲解了Tomcat的配置、性能调整、系统安全和部署架构，并没有介绍如何使用Tomcat部署Web应用程序。如果读者想学习如何使用Tomcat部署Web应用程序，可以选择Beginning JavaServer Pages ( Wrox出版，ISBN 0-7645-7485-x )。本书可以满足Tomcat管理员的学习需求。本书是Apache Tomcat系列的第三版。第一版Professional Apache Tomcat主要讲解Tomcat 3和Tomcat 4。第二版Professional . Apache Tomcat 5主要讲解Tomcat 5。第二版以后的变化 与前两版相比，本书有众多改进之处。书中讲解了一个新的规范 ( Servlet 2 . 5、JavaServer Pages 2 . 1 ) 和实现该规范的全新的Tomcat版本 ( Tomcat 6 )。Tomcat 6优化了性能和内存，提供更快速的、更可靠的连接器，并改善了集群的实现。除了更新的内容以外，本书还包括以下内容。完整的Tomcat6开发指导。加入了关于性能调整和新的高性能的APR及NIC ) 连接器的章节。加入了关于日志的章节，Tomcat服务器日志和来自Web应用程序的日志，介绍日志文件管理策略和日志分析。修订关于使用其JMX支持管理和监视Tomcat的章节。修订关于集群的章节，Tomcat 6集群支持的改善，包括新的集群配置。修订关于安全的Tomcat安装和Web应用程序章节。Tomcat 6的Web服务器连接器——mod proxy和mod . ik。我们非常重视您的反馈，并改进了第二版中一些需要更改的地方。读者将发现一些章节已经根据反馈意见进行了改写，以使其组织更合理和内容更丰富。

# 《Apache Tomcat 6高级编程》

## 内容概要

《Apache Tomcat 6高级编程》全面介绍了安装、配置和运行Apache Tomcat服务器的知识。书中不仅提供了配置选项的逐行分析，还探究了Tomcat的特性和功能，可以帮助读者解决出现在系统管理的各个阶段的各种问题，包括共享主机、安全、系统测试和性能测试及调优。

《Apache Tomcat 6高级编程》重点讲解Tomcat 6的应用知识。从基本的Tomcat和Web应用程序配置，到用于集群、JDBC连接、日志和其他功能的更高级技术，这些内容将帮助读者高效地管理和执行Tomcat部署。

《Apache Tomcat 6高级编程》适合于负责Tomcat配置、性能调优、系统安全或部署架构的J2EE系统管理员和Java开发者阅读。

# 《Apache Tomcat 6高级编程》

## 作者简介

Vivek Chopra拥有13年以上的软件架构师、开发者和团队领导的经验，他在硅谷的许多公司工作过。他写的技术书籍很生动，他是这本关于Java、开源软件、XML和Web服务图书的主要作者。他已经申请了关于Web服务技术的专利。过去3年他是Java Community Process (JCP) 的会员，也是JSR 280 (Java ME的XML API) 专家组的成员。Sing Li (在20世纪70年代末就已经发现微机的bug) 是伴随着微处理器的发展长大的。他的第一台个人电脑价值99美元，自制的Netronics COSMIC ELF，具有256字节的内存，是向PopularElectronics杂志邮购的。他具有20年行业经验，是系统开发者、开源软件贡献者和Java技术、嵌入式及分布式系统架构方面的业余作家。他时常为一些流行的技术期刊和电子杂志写文章，同时是最早的Internet电话服务之一——Imemet Global Phone的创始人。他曾经编著和与人合著了许多技术书籍，包括Geronimo、Tomcat、JSP、servlet、XML、Jini、媒体流、设备驱动和JXTA。

Jeff Genender拥有18年以上的软件架构师、团队领导，具有多家公司的开发经验，他是Apache Geronimo的Project Management Committee (PMC) 成员，也是关于OpenTerraCotta、OpenEJB、ServiceMix和Mojo (Maven插件) 的委员。作为Apache软件基金会的代表，他也是JSR.313 (Java平台，企业版6[Java EE 6规范]) 的Java Community Process (JCP) 专家组的成员。他通过努力成功地发展了开源软件，并被许多全球2000强公司分享，他的这些成功经验为这些机构节省了数百万的许可证费用。

## 书籍目录

### 第1章 Apache Tomcat

- 1.1 谦逊的开始：Apache项目
- 1.2 Apache软件基金会
- 1.3 Tomcat
  - 1.3.1 分发Tomcat：Apache的许可证
  - 1.3.2 与其他许可证的比较
- 1.4 蓝图：Java EE
  - 1.4.1 Java API
  - 1.4.2 Java EE API
  - 1.4.3 Java EE应用服务
  - 1.4.4 向标准化看齐，在实践中竞争
  - 1.4.5 Tomcat和应用服务器
- 1.5 Tomcat和Web服务器
- 1.6 本章小结

### 第2章 Web的应用：Servlet、JSP及其他

- 2.1 Web应用简史
  - 2.1.1 CGI脚本：第一个动态文本机制
  - 2.1.2 Java服务器端：Servlet
  - 2.1.3 JavaServer Page
  - 2.1.4 JSP标签库
  - 2.1.5 JSP EL
  - 2.1.6 MVC架构
  - 2.1.7 使用合适的Web技术
- 2.2 Web应用的创建和分类
- 2.3 本章小结

### 第3章 Tomcat安装

- 3.1 安装Java虚拟机
  - 3.1.1 在Windows上安装JVM
  - 3.1.2 在Linux上安装JVM
- 3.2 安装Tomcat
  - 3.2.1 选择安装类别
  - 3.2.2 检验下载文件
  - 3.2.3 Windows上安装Tomcat
  - 3.2.4 在Windows下使用ZIP文件安装Tomcat
  - 3.2.5 在Linux下安装Tomcat
- 3.3 使用源代码构建Tomcat
  - 3.3.1 是需要通过源代码构建Tomcat
  - 3.3.2 下载源代码版本
  - 3.3.3 subversion储存库
  - 3.3.4 构建一个源代码版本
- 3.4 Tomcat安装目录
- 3.5 安装APR
- 3.6 疑难解答和详细说明
  - 3.6.1 类版本错误
  - 3.6.2 使用中的端口号
  - 3.6.3 运行多重实例
  - 3.6.4 代理阻塞通道

## 3.7 本章小结

## 第4章 Tomcat架构

### 4.1 Tomcat目录概述

#### 4.1.1 bin目录

#### 4.1.2 conf目录

#### 4.1.3 lib目录

#### 4.1.4 logs目录

#### 4.1.5 temp目录

#### 4.1.6 webapps目录

#### 4.1.7 work目录

### 4.2 Tomcat架构概述

#### 4.2.1 服务器

#### 4.2.2 服务

#### 4.2.3 Tomcat架构中的其余类

### 4.3 连接器架构

#### 4.3.1 通信路径

#### 4.3.2 连接器协议

#### 4.3.3 选择连接器

### 4.4 生命周期

#### 4.4.1 Lifecycle接口

#### 4.4.2 LifecycleListener接口

### 4.5 根据架构进行配置

## 4.6 本章小结

## 第5章 Tomcat的基本配置

### 5.1 Tomcat 6配置的要点

### 5.2 \$CATALINA\_HOME/conf中的文件

### 5.3 服务器的基本配置

#### 5.3.1 通过server.xml默认文件配置服务器

#### 5.3.2 应用服务器配置下的Tomcat运行

#### 5.3.3 Web应用软件的Context定义

#### 5.3.4 默认context.xml文件

#### 5.3.5 认证和tomcat-users.xml文件

#### 5.3.6 默认的部署描述符——Web.xml

#### 5.3.7 servlet.xml、Context描述符和web.xml怎样协同工作

#### 5.3.8 严密的访问控制：catalina.policy

#### 5.3.9 Catalina.properties：建立在访问查询上的严密的访问控制

#### 5.3.10 引导配置

#### 5.3.11 配置和管理的不同点

### 5.4 Tomcat 6基于Web的GUI配置器

## 5.5 本章小结

## 第6章 高级Tomcat特性

### 6.1 阀——截取Tomcat样式

### 6.2 标准阀

### 6.3 访问日志实现

### 6.4 日志文件范围

### 6.5 单点登录实现

#### 6.5.1 不使用单点登录阀实现多次登录

#### 6.5.2 配置单点登录阀

### 6.6 Form认证器阀

## 6.7 通过请求过滤器限制访问

### 6.7.1 远程地址过滤器

### 6.7.2 远程主机过滤器

### 6.7.3 配置请求过滤器阀

### 6.7.4 请求记录器阀

## 6.8 持久会话

### 6.8.1 持久会话的必要性

### 6.8.2 配置Persistent Session Manager

## 6.9 JNDI资源配置

### 6.9.1 什么是JNDI

### 6.9.2 Tomcat和JNDI

### 6.9.3 典型的Tomcat JNDI资源

### 6.9.4 通过JNDI配置资源

### 6.9.5 配置JDBC DataSource

### 6.9.6 配置Mail会话

## 6.10 配置Lifecycle listener

### 6.10.1 Tomcat组件发送的生命周期事件

### 6.10.2 Listener元素

### 6.10.3 Tomcat 6 Lifecycle listener配置

## 6.11 本章小结

## 第7章 Web应用程序配置

### 7.1 理解Web应用程序内容

#### 7.1.1 公共资源

#### 7.1.2 Web-INF目录

#### 7.1.3 META-INF目录

### 7.2 理解部署描述符

#### 7.2.1 Servlet 2.3类型部署描述符

#### 7.2.2 Servlet 2.4/2.5类型部署描述符

## 7.3 本章小结

## 第8章 Web应用程序管理

### 8.1 Web应用程序示例

### 8.2 Tomcat管理员程序

#### 8.2.1 允许访问管理员程序

#### 8.2.2 配置管理员程序

### 8.3 Tomcat管理器：网络接口

#### 8.3.1 显示Tomcat服务器状态

#### 8.3.2 管理Web应用程序

#### 8.3.3 部署Web应用程序

### 8.4 Tomcat管理器：用Ant管理应用程序

### 8.5 Tomcat管理器：使用HTTP请求

#### 8.5.1 列出已部署的应用程序

#### 8.5.2 部署新的应用程序

#### 8.5.3 在Tomcat 6中安装/部署应用程序

#### 8.5.4 远程部署新的应用程序

#### 8.5.5 从本地路径部署新的应用程序

#### 8.5.6 重载现有的应用程序

#### 8.5.7 列出可用的JNDI资源

#### 8.5.8 列出OS和JVM属性

#### 8.5.9 停止现有的应用程序

- 8.5.10 启动被停止的应用程序
- 8.5.11 反部署Web应用程序
- 8.5.12 显示会话统计
- 8.5.13 使用JMX代理Servlet查询Tomcat内核
- 8.5.14 使用JMX代理Servlet设置Tomcat内核
- 8.6 可能发生的错误
- 8.7 安全性考虑
- 8.8 Tomcat部署器
- 8.9 本章小结
- 第9章 类装载器
  - 9.1 类装载器概述
    - 9.1.1 标准的Java SE类装载器
    - 9.1.2 关于类装载器的更多行为
    - 9.1.3 创建自定义类装载器
    - 9.1.4 为什么Tomcat需要自定义的类装载器
  - 9.2 安全性和类装载器
    - 9.2.1 类装载器委派
    - 9.2.2 核心类限制
    - 9.2.3 独立的类装载器命名空间
    - 9.2.4 SecurityManager
  - 9.3 Tomcat与类装载器
    - 9.3.1 系统类装载器
    - 9.3.2 Endorsed Standards Override机制
    - 9.3.3 通用类装载器
    - 9.3.4 Web应用程序类装载器
  - 9.4 动态类重载
  - 9.5 通用类装载器缺陷
    - 9.5.1 在不同的类装载器中分割包
    - 9.5.2 Singleton
    - 9.5.3 XML分析器
  - 9.6 本章小结
- 第10章 HTTP连接器
  - 10.1 HTTP连接器
    - 10.1.1 Tomcat 6 HTTP/1.1连接器
    - 10.1.2 高级NIO连接器
    - 10.1.3 Comet异步IO支持
    - 10.1.4 本地APR连接器
  - 10.2 配置Tomcat支持CGI
  - 10.3 配置Tomcat以支持SSI
    - 10.3.1 配置Tomcat 6 SSI Servlet
    - 10.3.2 配置Tomcat 6 SSI过滤器
  - 10.4 在代理服务器后运行Tomcat
  - 10.5 性能调整
    - 10.5.1 可调整的配置属性
    - 10.5.2 TCP/IP栈调整技巧
  - 10.6 前端Tomcat 6和Web服务器
  - 10.7 本章小结
- 第11章 Tomcat与Apache HTTP服务器
  - 11.1 AJP连接器体系结构



- 11.1.1 Apache的本地代码模块
- 11.1.2 Apache JServ协议
- 11.1.3 AJP连接器
- 11.2 Apache Web服务器前端或独立的Tomcat
- 11.3 了解Tomcat Worker
  - 11.3.1 多个Tomcat Worker
  - 11.3.2 使用workers.properties文件配置Apache服务器 与多个Tomcat Worker协同工作
- 11.4 连接Tomcat与Apache
  - 11.4.1 Tomcat 6配置
  - 11.4.2 Apache Web服务器配置
  - 11.4.3 使用mod\_jk模块
  - 11.4.4 使用mod\_proxy模块
- 11.5 为Apache Web服务器配置SSL
  - 11.5.1 为Apache配置mod\_ssl
  - 11.5.2 测试启用SSL的Apache设置
  - 11.5.3 启用SSL的Apache-Tomcat设置
- 11.6 Tomcat负载均衡与Apache
  - 11.6.1 更改Tomcat启动文件中的CATALINA\_HOME
  - 11.6.2 设置不同的AJP连接器端口
  - 11.6.3 设置不同的服务器端口
  - 11.6.4 禁用默认的HTTP/1.1连接器
  - 11.6.5 在Standalone Engine中设置jvmRoute
  - 11.6.6 注释 Catalina Engine
  - 11.6.7 httpd.conf中的指令
  - 11.6.8 workers.properties中的worker配置
- 11.7 测试负载均衡器
  - 11.7.1 测试Sticky会话
  - 11.7.2 测试Round-Robin行为
  - 11.7.3 在不同的负载系数下测试
- 11.8 本章小结
- 第12章 Tomcat与IIS
  - 12.1 ISAPI插件的角色
  - 12.2 连接Tomcat和IIS
    - 12.2.1 检查Tomcat与IIS的安装
    - 12.2.2 配置JK连接器
    - 12.2.3 安装ISAPI插件
    - 12.2.4 配置Tomcat worker
    - 12.2.5 配置转发请求规则
    - 12.2.6 可选的URL重写规则配置
    - 12.2.7 为ISAPI插件升级Windows注册表
    - 12.2.8 IIS 5独立模式(只针对IIS 6)
    - 12.2.9 在IIS下创建虚拟目录
    - 12.2.10 添加作为IIS过滤器的ISAPI插件
    - 12.2.11 授权作为Web应用程序扩展的ISAPI插件(只针对IIS 6)
    - 12.2.12 测试最后的设置
  - 12.3 故障解决技巧
  - 12.4 使用SSL
  - 12.5 IIS和Tomcat的可扩展架构
    - 12.5.1 分发Web和应用程序部署

- 12.5.2 多个Tomcat Workers
- 12.5.3 负载均衡的AJP Worker
- 12.6 本章小结
- 第13章 JDBC连接
- 13.1 JDBC基础
  - 13.1.1 建立和终止RDBMS连接
  - 13.1.2 JDBC版本演化
  - 13.1.3 JDBC驱动类型
  - 13.1.4 数据库连接池
  - 13.1.5 连接池的问题
  - 13.1.6 Tomcat和JDBC演化
- 13.2 Tomcat 6中的JNDI模拟和连接池
- 13.3 首选的配置：JNDI资源
  - 13.3.1 Resource标签
  - 13.3.2 继承JNDI资源配置
  - 13.3.3 测试JNDI资源配置
- 13.4 可选的JDBC配置
- 13.5 可选的连接池管理器
  - 13.5.1 关于c3p0池管理器
  - 13.5.2 部署c3p0池管理器
  - 13.5.3 无需JNDI查找获得JDBC连接
  - 13.5.4 使用c3p0测试对非JNDI池的访问
  - 13.5.5 使用JNDI映射获得连接
  - 13.5.6 使用Tomcat 6 JNDI-comp-atible查找测试c3p0
  - 13.5.7 部署第三方连接池
- 13.6 本章小结
- 第14章 Tomcat安全
- 14.1 核查下载的Tomcat的完整性
  - 14.1.1 核查MD5摘要
  - 14.1.2 使用PGP签名来核查下载的文件
- 14.2 确保Tomcat服务器安全安装
  - 14.2.1 去除默认的应用程序
  - 14.2.2 ROOT和tomcat-docs
  - 14.2.3 系统应用程序——manager和host-manager
  - 14.2.4 约束系统应用程序访问安全性
  - 14.2.5 删除JSP和Servlet的例子
  - 14.2.6 更改SHUTDOWN命令
- 14.3 使用特殊账户运行Tomcat
  - 14.3.1 创建一个无特权的Tomcat用户
  - 14.3.2 使用Tomcat用户运行Tomcat
- 14.4 确保文件系统安全
  - 14.4.1 Windows文件系统
  - 14.4.2 Linux文件系统
- 14.5 确保Java虚拟机的安全
  - 14.5.1 Security Manager概述
  - 14.5.2 使用Tomcat的Security Manager
  - 14.5.3 推荐的Security Manager练习
- 14.6 确保Web应用程序的安全
  - 14.6.1 “认证”和“域”

- 14.6.2 域安全性
- 14.7 SSL加密
  - 14.7.1 JSSE
  - 14.7.2 使用SSL保护资源
- 14.8 保证DefaultServlet的安全
  - 14.8.1 禁用目录列表
  - 14.8.2 禁用Invoker Servlet、SSI和CGI Gateway
- 14.9 主机限制
- 14.10 本章小结
- 第15章 共享的Tomcat主机
  - 15.1 虚拟主机概念
  - 15.2 Apache中的虚拟主机
    - 15.2.1 实例部署方案
    - 15.2.2 Apache中基于IP的虚拟主机
    - 15.2.3 Apache中基于名称的虚拟主机
  - 15.3 Tomcat中的虚拟主机
    - 15.3.1 实例部署方案
    - 15.3.2 作为独立服务器的Tomcat
    - 15.3.3 使用Apache的Tomcat
    - 15.3.4 配置Apache
  - 15.4 Tomcat Host-Manager应用程序
  - 15.5 虚拟主机问题：稳定性、安全和性能
  - 15.6 调整Tomcat中的虚拟主机设置
    - 15.6.1 为每个虚拟主机创建独立的JVM
    - 15.6.2 在Tomcat JVM上设置内存资源限制
    - 15.6.3 使用Java Security Manager限制
  - 15.7 本章小结
- 第16章 使用JMX监视和管理Tomcat
  - 16.1 管理需求
  - 16.2 关于JMX
    - 16.2.1 JMX体系结构
    - 16.2.2 工具层
    - 16.2.3 代理层
    - 16.2.4 分布式服务层
  - 16.3 JMX远程API
  - 16.4 MBean介绍
    - 16.4.1 标准MBean
    - 16.4.2 动态MBean
    - 16.4.3 模型MBean
    - 16.4.4 开放MBean
  - 16.5 JMX在Tomcat 6中的可管理元素
    - 16.5.1 可管理的Tomcat 6体系结构组件
    - 16.5.2 可管理的嵌入式组件
    - 16.5.3 可管理的运行时数据对象
    - 16.5.4 可管理的资源对象
  - 16.6 通过Manager Proxy访问Tomcat 6的JMX Support
    - 16.6.1 使用JMX Proxy
    - 16.6.2 修改MBean属性
    - 16.6.3 使用jconsole GUI监视 Tomcat

16.6.4 为远程监视配置Tomcat

16.7 本章小结

第17章 集群

17.1 集群益处

17.1.1 可扩展性与集群

17.1.2 高可靠性的必要性

17.2 集群基础

17.2.1 主-备用拓扑模式

17.2.2 故障转移行为模式

17.3 Tomcat 6集群模型

17.3.1 负载均衡

17.3.2 会话共享

17.4 使用Tomcat 6集群

17.4.1 Tomcat 6中的会话管理

17.4.2 Cookie和现代浏览器的角色

17.4.3 配置一个Tomcat 6集群

17.4.4 通用前端：通过Apache mod\_jk使负载均衡

17.4.5 准备使用不同的后端会话共享

17.4.6 后端1：内存复制配置

17.4.7 后端2：使用共享文件存储的永久会话管理器

17.4.8 后端3：使用JDBC存储的永久会话管理器

17.4.9 测试使用JDBC永久会话管理器后端的Tomcat集群

17.5 集群的复杂性

17.5.1 集群和性能

17.5.2 集群和响应时间

17.5.3 使用集群解决性能问题

17.6 本章小结

第18章 嵌入式Tomcat

18.1 现代系统设计中嵌入式Tomcat的重要性

18.1.1 典型地嵌入式应用程序

18.1.2 使用嵌入式Tomcat开发

18.2 本章小结

第19章 日志

19.1 从Tomcat 5的变革

19.2 log4j

19.2.1 log4j体系结构

19.2.2 log4j安装和配置

19.2.3 log4j入门指南

19.2.4 log4j高级用法

19.2.5 log4j性能建议

19.3 JULI

19.3.1 Java日志体系结构

19.3.2 JULI入门指南

19.4 日志文件分析

19.5 本章小结

第20章 性能测试

20.1 性能概念

20.1.1 评估什么

20.1.2 可扩展性和性能

- 20.1.3 理解用户角度的性能
- 20.1.4 评估性能
- 20.2 JMeter
  - 20.2.1 安装和运行JMeter
  - 20.2.2 设计和理解使用JMeter的测试计划
  - 20.2.3 JMeter特性
  - 20.2.4 分布式负荷测试
  - 20.2.5 解释测试结果
- 20.3 替代JMeter的工具
- 20.4 性能测试后的事项
- 20.5 本章小结
- 第21章 性能调优
  - 21.1 性能调优的最佳实践
    - 21.1.1 步骤1：建立测试台
    - 21.1.2 步骤2：性能测试和确定基线
    - 21.1.3 步骤3：诊断性能瓶颈
  - 21.2 诊断Tomcat的性能问题
  - 21.3 Tomcat性能调优提示
    - 21.3.1 JVM的参数调优
    - 21.3.2 预编译JSP
    - 21.3.3 Tomcat配置调优
    - 21.3.4 适当的时候使用Web服务器的静态内容
  - 21.4 本章小结
- 附录A Tomcat和集成开发环境
  - A.1 Eclipse
    - A.1.1 调试Eclipse中的远程Web应用程序
    - A.1.2 使用Sysdeo Tomcat插件部署和调试本地Web应用程序
    - A.1.3 使用Web工具平台部署和调试Web应用程序
    - A.1.4 使用Apache Ant和Eclipse管理Web应用程序部署
  - A.2 NetBeans
    - A.2.1 调试NetBeans中的远程Web应用程序
    - A.2.2 调试NetBeans内的Web应用程序
  - A.3 小结
- 附录B Apache Ant
  - B.1 安装Ant
  - B.2 Ant介绍
  - B.3 Ant诀窍
    - B.3.1 使用Ant创建Web应用程序
    - B.3.2 编译JSP
    - B.3.3 利用属性文件和命令行参数的可重用Ant脚本
    - B.3.4 创建日志
    - B.3.5 通过电子邮件创建通知
    - B.3.6 Ant和源控制系统
    - B.3.7 自动测试
  - B.4 持续集成
  - B.5 Ant任务参考
  - B.6 小结

## 章节摘录

Tomcat就是这样一個Servlet容器。它为Servlet，提供了执行环境以及系统资源（如文件系统），并维护客户的身份验证。正如在第1章所提到的，这也用在Servlet规范的参考实现中。虽然Servlet的规范允许使用除了HTTP以外的其他传输工具，但在实践中，Servlet几乎是专用于提供因特网上应用功能和为HTTP请求服务的。像CGI一样，Servlet规范设计为静态连接和创建Web应用程序之外扩展网站服务器提供了一种标准途径。与CGI区别的是，Servlet仅限于Java语言，尽管这使它因平台独立性而受益。如同Java语言，Servlet规范创造的目的是使第三方提供容器，以便在价格、性能和易用性上具有竞争力。原则上，因为这些容器是标准的，所以这些第三方的客户们可以在它们之间自由选择相对开销较小的移植。然而，在实践中，厂商的Servlet容器也与超出规范的服务竞争。此外，在几个领域内实现规范的准确途径是实行开放式阐述（open to interpretation）。在容器中运行的类装载器（负责容器中激活类以使其可以被应用程序所用）就是一个这样的例子。Tomcat的类装载器将在第9章中阐述。然而，比起重新编程及重新编译应用程序来，移植通常更像是一个容器配置问题。这使得我们猜想，程序员对使用非标准服务的Servlet容器和跨容器兼容编程不感兴趣。

## 编辑推荐

通过阅读本书，读者将掌握安装、配置和运行Apache Tomcat服务器的全部知识。本书不仅提供了配置选项的逐行分析，还探究了Tomcat的特性和功能。因此读者将能获得解决各种问题的技巧，这些问题出现在系统管理的各个阶段，包括共享主机、安全、系统测试和性能测试及调优。本书重点讲解Tomcat 6的应用知识，介绍了新的架构和性能更改。从基本的Tomcat和Web应用程序配置到用于集群、JDBC连接、日志和其他功能的高级技术，这些内容将帮助读者高效地管理和执行Tomcat部署。

**主要内容：** 在Windows和UNIX / Linux系统上安装JVM和Tomcat的方法； 打包和部署Web应用程序的步骤； 配置Tomcat的内部HTTP协议栈，包括新的APR和NIO连接器； 使用Apache Web服务器或Microsoft IIS作为前端Web月E务器来部署Tomcat的方法； 负载均衡和大量Tomcat服务器的集群； 在虚拟主机环境下安装Tomcat的方法； 加载部署在Tomcat中的测试Web应用程序的方法； 使用内部的维护统计功能实时监控Tomcat服务器的方法； 提供Web应用程序的扩展性和高可用性的技术； 提供Tomcat 6的性能技巧和最佳实践。

**本书读者对象：** 本书适于负责Tomcat配置、性能调优、系统安全或部署架构的J2EE系统管理员和Java开发者阅读。本书由在职程序员组织和撰写，能够充分满足程序员、开发人员和IT专业人士的实际需求。本书内容定位于技术专家日常面对的各种技术问题，提供了示例、实用解决方案和专家级的新技术剖析，这些内容都可以帮助程序员更好地完成工作。

**作者简介：** vivek Chopra拥有13年以上的软件架构师、开发者和团队领导的经验，他在硅谷的许多公司工作过。他写的技术书籍很生动，他是这本关于Java、开源软件、XML和Web服务图书的主要作者。他已经申请了关于web服务技术的专利。过去3年他是Java Community Process ( JCP ) 的会员，也是JSR 280 ( Java ME的XML API ) 专家组的成员。

# 《Apache Tomcat 6高级编程》

## 精彩短评

- 1、看原版吧，我一口气读完了，翻译的太差了，真的，从未读过如此差劲的翻译书籍，但不可避免的是，英文原版还是比较有技术含量的~
- 2、了解tomcat 架构,但是书中介绍的大多数基于JNI和JVAEE的介绍有些过时



# 《Apache Tomcat 6高级编程》

## 精彩书评

1、讲的还不错，虽然翻译不是特别给力，但是整体还是面面俱到，基本都能理解，然后对以前用过的东西做个回忆就理解会更加深入。tomcat集群部分一直都没去了解过，今天看了终于理解的公司的某些产品在集群方面为何这样配置，基本能知道了采用的策略，自己再对比下各种本身提供的策略还是有许多感触的

## 章节试读

### 1、《Apache Tomcat 6高级编程》的笔记-第1页

目前正在读，还不错，喜欢这个系列。

# 《Apache Tomcat 6高级编程》

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu000.com](http://www.tushu000.com)