

## <<Java开发超级工具集>>

### 图书基本信息

书名：<<Java开发超级工具集>>

13位ISBN编号：9787302209713

10位ISBN编号：7302209715

出版时间：2009-11

出版时间：清华大学出版社

作者：斯马特

页数：762

译者：张文波

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：<http://www.tushu007.com>

## <<Java开发超级工具集>>

### 前言

所有真正的艺人都需要最好的工具来完成最佳工作，程序员也不例外。  
本书介绍了30种开源工具，这些工具被设计用于改进任何规模的团队或组织中的Java开发人员的开发实践。

书中每章包括多个小节，每章介绍了一种特定工具（无论该工具是用于构建系统、版本控制还是用于开发过程的其他方面），因此，读者相当于在一本书中获得了30本简短参考书。

无论选择何种开发方法（敏捷开发、RUP、XP、SCRUM或是其他方法），本书中的实用技术和工具都能帮助用户优化开发流程并将其自动化。

本书讨论了Java开发的关键问题领域和最佳实践，重点介绍了开源工具，这些工具能够提高开发周期中每个领域的生产效率，它们包括：

- 构建工具，如Ant和Maven 2。
- 版本控制工具，如CVS和Subversion。
- 质量度量工具，如CheckStyle、PMD、FindBugs和Jupiter。
- 文档工具，这些工具可以减少用于编写和维护文档的时间，同时又能生成良好文档。
- 单元测试工具，如JUnit 4、TestNG和开源测试覆盖工具Cobertura。
- 自动化集成、负载和性能测试工具；自动化web服务、Swing界面和web界面测试工具。
- 问题管理工具，如Bugzilla和Trac。
- 持续集成工具，如Continuum、CruiseControl、LuntBuild和Hudson。

这些工具可以改进开发实践并使开发过程更加轻松，本书是首席开发人员和软件架构师的必读读物，它可以使其职业生活井然有序。

John Ferguson Smart是Wakaleo咨询公司（[www.wakaleo.com/](http://www.wakaleo.com/)）的首席咨询顾问，该公司提供企业Java和敏捷开发方面的咨询、培训和指导服务。

## <<Java开发超级工具集>>

### 内容概要

作为一个Java软件开发人员，你正为某个问题而苦恼吗？

因为某个问题而影响开发进度，影响开发效率吗？

拿起这本书吧！

也许，你的问题的答案就在《Java开发超级工具集》中，你只需使用《Java开发超级工具集》介绍的一个小工具，所有问题就迎刃而解了！

Java是现今最为流行的软件开发语言了。

为了提高开发人员的效率，人们开发了大量的辅助工具，《Java开发超级工具集》介绍了Java语言的各种功能强大的辅助开发工具，包括构建程序与调试，技术文档的编写、单元测试及集成、负载和性能测试，质量度量与问题管理，版本控制与持续集成等，涵盖了Java软件开发的各个方面。

这些软件开发工具，都是作者从浩如烟海的大量工具中精挑细选出来的，并进行了分类，给出了详细的使用说明，帮助你提高开发效率，更出色地完成任务。

## <<Java开发超级工具集>>

### 作者简介

John Ferguson Smart, 是Wakaleo咨询公司的首席咨询顾问, 该公司提供企业Java和敏捷开发方面的咨询、培训和指导服务。

## &lt;&lt;Java开发超级工具集&gt;&gt;

## 书籍目录

第一部分 构建工具第1章 使用Ant设置项目1.1 构建过程中的Ant1.2 安装Ant1.3 Ant概述1.4 在Ant中编译Java代码1.5 使用属性自定义构建脚本1.6 在Ant中运行单元测试1.7 用Javadoc生成文档1.8 将应用程序打包1.9 部署应用程序1.10 引导构建脚本1.11 用Maven任务在Ant中使用Maven依赖1.12 在Eclipse中使用Ant1.13 在NetBeans中使用Ant1.14 用XMLTask操作XML1.15 小结第2章 使用Maven 2设置项目2.1 Maven和开发构建过程2.2 Maven和Ant2.3 安装Maven2.4 声明式构建和Maven项目对象模型2.5 理解Maven 2的生命周期2.6 Maven目录结构2.7 根据环境配置Maven2.8 Maven 2中的依赖管理2.9 用MvnRepository查找依赖2.10 项目继承和聚合 (Aggregation) 2.11 使用原型 (Archetype) 创建项目模板2.12 编译代码2.13 测试代码2.14 打包和部署应用程序2.15 使用Cargo部署应用程序2.16 在Eclipse中使用Maven2.17 在NetBeans中使用Maven2.18 使用插件定制构建过程2.19 用Archiva设置企业存储库2.20 使用Artifactory设置企业存储库2.21 在Maven中使用Ant2.22 高级原型2.23 使用组件 (Assembly) 第二部分 版本控制工具第3章 使用CVS设置版本控制3.1 CVS概述3.2 设置CVS仓库3.3 在CVS中创建新项目3.4 检出项目3.5 处理文件 - - 更新和提交3.6 解决仓库锁定问题3.7 使用关键字替换3.8 处理二进制文件3.9 CVS标记3.10 在CVS中创建分支3.11 从分支中合并更改3.12 查看更改历史3.13 还原更改3.14 在Windows中使用CVS第4章 用Subversion设置版本控制4.1 Subversion概述4.2 安装Subversion4.3 Subversion仓库类型4.4 设置Subversion仓库4.5 设置新的Subversion项目4.6 检出工作副本4.7 将现有文件导入到Subversion4.8 理解Subversion仓库的URL4.9 使用文件4.10 查看当前状态: Status命令4.11 解决冲突4.12 使用标记、分支和合并4.13 回滚到以前的修订版本4.14 对二进制文件使用文件锁定4.15 打破和窃取锁定4.16 用svn: needs-lock属性使锁定文件成为只读4.17 使用属性4.18 Subversion中的更改历史记录: 日志和Blaming4.19 用svnserve设置Subversion服务器4.20 设置安全的svnserve服务器4.21 设置支持WebDAV/DeltaV的Subversion服务器4.22 设置安全的WebDAV/DeltaV服务器4.23 用钩子脚本定制Subversion4.24 将Subversion安装为Windows服务4.25 备份和还原Subversion仓库4.26 在Eclipse中使用Subversion4.27 在NetBeans中使用Subversion4.28 在Windows中使用Subversion4.29 缺陷跟踪和变更控制4.30 在Ant中使用Subversion4.31 小结第三部分 持续集成第5章 用Continuum设置持续集成服务器5.1 Continuum概述5.2 安装Continuum服务器5.3 手工启动和停止服务器5.4 检查服务器状态5.5 以Verbose模式运行Continuum服务器5.6 添加项目组5.7 添加Maven项目5.8 添加Ant项目5.9 添加外壳脚本项目5.10 管理项目构建5.11 管理用户5.12 设置通知方法 (Notifier) 5.13 配置和制订构建计划5.14 调试构建5.15 配置Continuum邮件服务器5.16 配置Continuum网站端口5.17 用Continuum自动生成Maven网站5.18 配置手工构建任务5.19 小结第6章 用CruiseControl设置持续集成服务器6.1 CruiseControl概述6.2 安装CruiseControl6.3 配置Ant项目6.4 用Publisher (发布者) 通知开发人员6.5 在CruiseControl中设置Maven 2项目6.6 CruiseControl操作面板6.7 第三方工具6.8 小结第7章 LuntBuild - - 基于Web的持续集成服务器7.1 LuntBuild概述7.2 安装LuntBuild7.3 配置LuntBuild服务器7.4 添加项目7.5 为版本编号使用项目变量7.6 构建结果诊断7.7 与Eclipse一起使用LuntBuild7.8 在Luntbuild中使用Cobertura报告测试覆盖7.9 将Luntbuild与Maven集成7.10 小结第8章 使用Hudson进行持续集成8.1 Hudson概述8.2 安装Hudson8.3 管理Hudson的主目录8.4 安装升级8.5 配置Hudson8.6 添加新构建任务8.7 组织任务8.8 监控构建8.9 查看和提升特定构建8.10 管理用户8.11 认证与安全8.12 查看更改8.13 Hudson插件8.14 记录测试结果8.15 记录代码度量8.16 报告代码覆盖第9章 用Openfire设置即时消息平台9.1 开发项目中的即时消息软件9.2 安装Openfire9.3 在Openfire上设置用户账户9.4 认证外部数据库中的用户9.5 针对POP3服务器认证用户9.6 用群聊天召开虚拟团队会议9.7 用Openfire插件扩展功能9.8 与Continuum一起使用Openfire9.9 与CruiseControl一起使用Openfire9.10 与Luntbuild一起使用Openfire9.11 使用Smack API从Java应用程序中发送Jabber消息9.12 用Smack API检测用户是否在线9.13 使用Smack API接收消息第四部分 单元测试第10章 用JUnit测试代码10.1 JUnit 3.8和JUnit 410.2 用JUnit 4进行单元测试10.3 设置和优化单元测试用例10.4 用Timeout进行简单的性能测试10.5 轻松地检查异常10.6 使用带参数的测试10.7 使用assertThat和Hamcrest库10.8 JUnit 4的理论机制10.9 与Maven 2一起使用JUnit 410.10 与Ant一起使用JUnit 410.11 在Ant中有选择地运行JUnit 4测试10.12 集成测试10.13 在Eclipse中使用JUnit 4第11章 用TestNG进行下一代测试11.1 TestNG概述11.2 用TestNG创建简单单元测试11.3 定义TestNG测试套件11.4 Eclipse的TestNG插件11.5 在Ant中使用TestNG11.6 与Maven 2一起使用TestNG11.7

## <<Java开发超级工具集>>

管理测试生命周期11.8 使用测试组11.9 管理依赖11.10 并行测试11.11 测试参数和数据驱动的测试11.12 检查异常11.13 处理部分失败11.14 重新运行失败的测试第12章 用Cobertura 最大化测试覆盖12.1 测试覆盖12.2 从Ant中运行Cobertura12.3 检查TestNG测试的代码覆盖12.4 理解Cobertura报告12.5 实施高水平代码覆盖12.6 在Maven中生成Cobertura报告12.7 将覆盖测试集成到Maven构建过程中12.8 Eclipse中的代码覆盖12.9 小结第五部分 集成、功能、负载和性能测试第13章 用StrutsTestCase测试Struts应用程序13.1 概述13.2 测试Struts应用程序13.3 StrutsTestCase概述13.4 使用StrutsTestCase 进行模拟测试13.5 测试Struts错误处理13.6 定制测试环境13.7 一级 (First-level) 性能测试13.8 小结第14章 用DbUnit进行数据库集成测试14.1 引言14.2 概述14.3 DbUnit的结构14.4 示例应用程序14.5 准备数据库14.6 验证数据库14.7 替换值14.8 其他数据集格式14.9 处理自定义数据类型14.10 其他应用第15章 用JUnitPerf进行性能测试15.1 JUnitPerf概述15.2 用TimedTest测量性能15.3 用LoadTest模拟负载15.4 对非线性安全的测试进行负载测试15.5 在Ant中分离性能测试和单元测试15.6 在Maven中分离性能测试和单元测试第16章 用JMeter进行负载和性能测试16.1 概述16.2 安装JMeter16.3 测试简单的web应用程序16.4 组织测试用例16.5 记录和显示测试结果16.6 使用JMeter代理服务器记录测试用例16.7 使用变量进行测试16.8 在多台计算机上进行测试第17章 用SoapUI测试Web服务17.1 概述17.1 SoapUI概述17.2 安装SoapUI17.3 安装本地web服务17.4 用SoapUI测试web服务17.5 用SoapUI进行负载测试17.6 从命令行运行SoapUI17.7 从Ant中运行SoapUI17.8 从Maven中运行SoapUI17.9 持续测试17.10 小结第18章 用Sun JDK工具监视和分析Java应用程序的性能18.1 Sun JDK性能分析和监视工具18.2 用jConsole连接并监视Java应用程序18.3 用jConsole监视远程Tomcat应用程序18.4 用JDK工具检测和识别内存泄露18.5 用堆转储、jmap和jhat诊断内存泄露18.6 检测死锁第19章 在Eclipse中分析Java应用程序的性能19.1 在集成开发环境中分析应用程序的性能19.2 Eclipse测试和性能工具平台19.3 安装TPTP19.4 TPTP和Java 619.5 使用TPTP进行基本性能分析19.6 用基本内存分析结果分析内存使用19.7 分析执行时间19.8 显示代码覆盖统计19.9 使用过滤器优化结果19.10 分析web应用程序的性能19.11 小结第20章 测试用户界面20.1 概述20.2 用Selenium测试web应用程序20.3 用FEST测试Swing图形用户界面20.4 小结第六部分 质量度量工具第21章 用Checkstyle检测和实施编码标准21.1 用Checkstyle实施编码标准21.2 在Eclipse中使用Checkstyle21.3 在Eclipse中定制Checkstyle规则21.4 使用XML配置文件定制Checkstyle规则21.5 定制Checkstyle：可以舍弃和使用的常见规则21.6 用Checkstyle定义源代码文件头规则21.7 禁用Checkstyle测试21.8 与Ant一起使用Checkstyle21.9 与Maven一起使用Checkstyle第22章 用PMD预先检测错误22.1 PMD和静态代码分析22.2 在Eclipse中使用PMD22.3 在Eclipse中配置PMD规则22.4 PMD规则集22.5 编写自己的PMD规则集22.6 在Eclipse中生成PMD报告22.7 禁用PMD规则22.8 用CPD检测剪切和粘贴22.9 在Ant中使用PMD22.10 在Maven中使用PMD第23章 用FindBugs预先检测错误23.1 FindBugs：专业的程序错误检测工具23.2 在Eclipse中使用FindBugs23.3 用FindBugs过滤器有选择地禁用规则23.4 使用FindBugs注释23.5 在Ant中使用FindBugs23.6 在Maven中使用FindBugs23.7 小结第24章 检查结果 - 用Jupiter进行半自动化代码评审24.1 Jupiter概述 - 用于Eclipse的代码评审工具24.2 在Eclipse中安装Jupiter24.3 理解Jupiter的代码评审流程24.4 进行个人代码审查24.5 配置24.6 设置默认配置值24.7 单独评审24.8 团队评审24.9 返工阶段24.10 Jupiter的后台处理24.11 小结第25章 用Mylyn突出工作重点25.1 Mylyn概述25.2 安装Mylyn25.3 跟踪任务和问题25.4 与任务仓库进行交互25.5 用上下文管理将工作重点集中在任务上25.6 使用Eclipse更改集25.7 与其他开发人员共享上下文25.8 小结第26章 监视构建统计信息26.1 概述26.2 QALab26.3 用StatSCM度量源代码管理26.4 在Ant中用StatSVN提供统计信息第七部分 问题管理工具第27章 Bugzilla27.1 Bugzilla概述27.2 安装Bugzilla27.3 设置Bugzilla环境27.4 管理用户账户27.5 使用用户组来限制访问27.6 配置产品27.7 用里程碑跟踪进度27.8 用分类管理产品组27.9 搜索程序错误27.10 创建新程序错误27.11 Bugzilla程序错误的生命周期27.12 安排通知 ( Whining ) 27.13 在Bugzilla中定制字段27.14 小结第28章 Trac - 轻量级项目管理28.1 Trac概述28.2 安装Trac28.3 设置Trac项目28.4 在独立服务器上运行Trac28.5 将Tracd设置为Windows服务28.6 在Apache服务器上安装Trac28.7 管理Trac网站28.8 管理用户账户28.9 剪裁Trac网站：使用Wiki功能28.10 使用Trac Ticket管理系统28.11 从Subversion中更新Trac问题28.12 定制Trac Ticket的字段28.13 设置电子邮件通知28.14 使用Trac查询和报告28.15 用Trac路线图 ( Roadmap ) 和时间线 ( Timeline ) 管理进度28.16 浏览源代码库28.17 使用RSS和iCalendar28.18 使用Python定制wiki页面28.19 小结第八部分 技术文档工具第29章 用Maven 2项目网站进行团队沟通29.1 作为沟通工具的Maven 2项目网站29.2 设置Maven网站项目29.3



## <<Java开发超级工具集>>

将报告集成到网站中29.4 创建专用Maven网站项目29.5 定义网站框架29.6 Maven网站生成架构29.7 使用代码片断（Snippet）29.8 定制网站外观29.9 发布网站第30章 自动生成技术文档30.1 概述30.2 用SchemaSpy直观查看数据库结构30.3 用Doxygen生成源代码文档30.4 用UmlGraph在Javadoc中嵌入UML图表30.5 小结参考资料

## <<Java开发超级工具集>>

### 章节摘录

第一部分 构建工具 “这正好说明，麻烦多一些也能带来好事，” 咿呦说，“你明白吗，噗熊维尼？”

你明白吗，皮杰？

先动脑筋，再下功夫，你们看！

这才是盖房子的方法。

——“在维尼角落，为咿呦修建了一座房子”，《维尼角落的家》，A.A.米尔恩 从一开始就投入精力来规划构建过程，这样，在以后工作困难重重而面临压力时，就会得到充分回报，这就是精心设计的构建过程和优化的构建工具其价值所在。

与IT和其他领域中的许多事物一样，构建工具主要是人类懒惰的结果。

从命令行编译C或C++（或Java，就本书而言）代码是非常乏味的，在充斥着脚本的Unix世界中，下一步就很自然了：为什么不编写脚本来为自己完成这项工作呢？

用户编写的用于编译少量C语言源代码文件的基本外壳脚本，可能是现代Java构建工具（如Ant和Maven）最古老的祖先。

对于少量源代码文件，外壳脚本工作良好，但这种方法难以扩展到更大的应用程序，这就是Make发挥作用的时候了。

Make是重要的Unix构建工具，熟悉Linux或Unix的用户一定会在某个阶段遇到它。

生成文件（makefile，Make运行的脚本名称）本质上就是用于编译应用程序的一组指令，其想法是，通过找出需要被编译的文件和编译顺序，将构建过程自动化。

可以通过定义依赖规则来实现这一点，依赖规则告诉Make应该在何时编译特定文件以及如何编译。



## <<Java开发超级工具集>>

### 媒体关注与评论

这是一本相当实用的Java技术图书，旨在帮助读者提高自己和团队的生产力。这些工具的法并不复杂，但很多开发人员因为不了解它们背后的动机和理念，对这些工具始终停留在“观望”阶段，想用却一直不知道应该怎么用才算用对、用好。本书语言精练、内容丰富，不仅提供了这些工具的“全景图”，告诉读者如何把这些工具组织在一起发挥出最大的作用，还给读者留下了一定的选择空间，让他们能在同类产品中根据自己的喜好和项目的特点选择适合自己的工具，而没有将作者主观的偏好强加于读者。愿这本书的出版能给广大Java程序员和项目经理带来生产力的飞跃。

——Rujia

## <<Java开发超级工具集>>

### 编辑推荐

所有真正的手艺人都需要最好的工具来完成最佳工作，程序员也不例外。本书介绍了30种开源工具，这些工具被设计用于改进任何规模的团队或组织中的Java开发人员的开发实践。

书中每章包括多个小节，每章介绍了一种特定工具(无论该工具是用于构建系统、版本控制还是用于开发过程的其他方面)，因此，读者相当于在一本书中获得了30本简短参考书。

无论选择何种开发方法(敏捷开发、RUP、XP、SCRUM或是其他方法)，本书中的实用技术和工具都能帮助用户优化开发流程并将其自动化。

本书讨论了Java开发的关键问题领域和最佳实践，重点介绍了开源工具，这些工具能够提高开发周期中每个领域的生产效率，它们包括：

- 构建工具，如Ant和Maven 2。
- 版本控制工具，如CVs和subversion。
- 质量度量工具，如CheckStyle、PMD、FindBugs和Jupiter。
- 文档工具，这些工具可以减少用于编写和维护文档的时间，同时又能生成良好文档。
- 单元测试工具，如JUnit 4、TestNG和开源测试覆盖工具Cobertura。
- 自动化集成、负载和性能测试工具；自动化Web服务、Swing界面和Web界面测试工具。
- 问题管理工具，如Bugzilla和Trac。
- 持续集成工具，如Continuum、CruiseControl、LuntBuild和Hudson。

这些工具可以改进开发实践并使开发过程更加轻松，本书是首席开发人员和软件架构师的必读读物，它可以使其职业生活井然有序。

## <<Java开发超级工具集>>

### 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:<http://www.tushu007.com>