

THE EXPERT'S VOICE® IN JAVA TECHNOLOGY

包含新的
NetBeans™
(J)Ruby/Rails IDE

Pro NetBeans IDE 6 Rich Client Platform Edition

NetBeans IDE 6

高级编程

使用开源的NetBeans IDE平台构建强大的
富客户端Internet应用程序



(美) Adam Myatt
刘志忠

等著
译



清华大学出版社

NetBeans IDE 6 高级编程

(美) Adam Myatt 等著

刘志忠

译



清华大学出版社

北 京

EISBN: 978-1-59059-895-5

Pro NetBeans™ IDE 6 Rich Client Platform Edition

Adam Myatt, et al

Original English language edition published by Apress, 2855 Telegraph Avenue, #600, Berkeley, CA 94705 USA. Copyright © 2008 by Apress L.P. Simplified Chinese-language edition copyright © 2008 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2008-2287

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

NetBeans IDE 6 高级编程/(美)马特(Myatt, A.)等著;刘志忠译.

—北京:清华大学出版社,2009.2

书名原文: Pro NetBeans™ IDE 6 Rich Client Platform Edition

ISBN 978-7-302-19466-8

I. N… II. ① 马… ② 刘… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 010653 号

责任编辑:王 军 郑雪梅

装帧设计:孔祥丰

责任校对:胡雁翎

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京市世界知识印刷厂

装 订 者:三河市兴旺装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:29.75 字 数:687 千字

版 次:2009 年 2 月第 1 版 印 次:2009 年 2 月第 1 次印刷

印 数:1~4000

定 价:68.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:029289-01

译者序

任何有过一定软件开发经验的人都知道：集成开发环境(Integrated Development Environment, IDE)对于程序开发人员来说就好比是勇士手中的利剑。好的集成开发环境往往能使得程序开发具有事半功倍的效果，而差的集成开发环境则会使得开发进展举步维艰。

NetBeans IDE 是一个针对 Java 的集成开发环境。它所提供的各种卓越特性使得越来越多的程序开发人员选择 NetBeans 这柄“利剑”来开发基于 Java 的应用。NetBeans IDE 所提供的特性涉及到了程序开发的整个声明周期，包括：良好的人机交互界面；智能的代码编程环境；集成的软件编译、构建以及部署环境；完善的性能分析、代码重构以及代码优化功能等。此外，NetBeans IDE 基本上支持所有的工程类型，在 NetBeans IDE 中可以构建简单的 Java 应用，也可以构建复杂的 Web 应用、Web 服务甚至富客户端应用。另外，通过安装各类插件，NetBeans IDE 还可以支持各种各样的第三方工具和类库等。

本书主要介绍了 NetBeans 6 为富客户端应用开发所提供的各种新特性，并在此基础上详细介绍了在 NetBeans 6 中开发各类应用的基本流程，包括 JRuby/Ruby on Rails、Web 应用、Web 服务、GUI 应用以及富客户端应用等。

本书主要介绍的是 NetBeans 6 的一些新特性，而对于 NetBeans IDE 以前版本中所提供的一些特性并未涉及，想了解 NetBeans IDE 的一般特性，可以阅读相关资料，或登录 NetBeans 的站点 netbeans.org。

作者简介

Adam Myatt 目前是 GE(通用电气公司)全球研究所软件开发室的主要技术人员。该研究所是通用电子的全球研发总部，位于纽约的 Niskayuna。Adam 是一位资深的 Java 开发人员，也是 Sun 微系统认证的 Java 程序员。他的工作贯穿了整个软件开发生命周期的始终，同时他也担负着领导全球 Java 软件和 Web 应用程序的开发、新技术的研发和建立长期战略的使命。

他是本地 Java 用户组的活跃成员，也是开源软件的狂热追求者。Adam 已经出版了 *Pro NetBeans 5.5 Enterprise Edition*(Apress, 2007)一书，该书主要介绍了 Java EE5 的相关技术及其在 NetBeans 中的应用。他最近作为 Tools & Languages Track Committee 的成员正在寻找 JavaOne 2008 的推荐标准。

Adam 在加入通用电气公司之前，在多家海外软件公司就职。他毕业于波茨坦纽约州立大学的计算机科学系。

在有限的空闲时间内，Adam 喜欢到新的有趣的地方旅行、钓鱼和打牌。最近，Adam 和他的妻子驱车穿梭于美国，行程达 6500 公里，遍布 20 多个州，在整个过程中没有收到任何超速罚单，也未发生任何交通事故。

前言

最初，代码都是用简单的基于文本的工具来编写的，如 Notepad。出于讨论的目的，本书将“最初”定义为 20 世纪 90 年代中期，也就是 Java 刚刚开始流行的时候。结合一个文本编辑器和命令提示行一起使用，用户可以编写并编译代码。

很快，人们就觉得该方法并不能提供一个最为高效的开发环境。例如，如果在文本编辑器中产生了一个代码语法错误，那么该错误要到保存并编译文件时才能被发现，而在此之前没有其他的方法可以确定这个问题。然后，查看编译错误，在代码中定位出错的代码行，并试图确定错误的原因。编译错误在诊断代码中的问题时并不总是完全有帮助。

很多新的程序员最初都是使用 Notepad 加命令行的编程环境。该方法本身并没有任何错误，因为很多专业程序员仍然使用相同的方式。对于一个纯粹的 Java 初学者而言，使用一个普通的文本编辑器是最容易也是最快的方法。但是，文本编辑并不提供语言语法的帮助，也不会提供编译器集成、智能反向工程支持以及其他编写代码的支持功能。

大多数文本编辑器处理中比较有用的一个特性就是“查找和替换”。通过这个简单的功能，程序员能够用一个单词替换某个单词或者短语的出现。这一点在一些情况下是有用的，但是也可能导致一些问题。假定创建了如下的类：

```
public class SomeCode {

    public void myMethod1(String var) {

        String FirstName = var.toUpperCase();

        // do something with FirstName
    }

    public void myMethod2(String var) {

        String FirstName = var.toLowerCase();

        // do something else with FirstName
    }

}
```

SomeCode 类包含两个方法：myMethod1 和 myMethod2。如果以后需要重命名 myMethod1 中的 FirstName 变量，就需要手动地编辑每一行代码来改变该方法名。显然，这是一个非常简单的例子。但是，如果 myMethod1 方法碰巧有数百行，并且 FirstName 在多个地方出现，那么手动编辑代码将花费大量的时间。也可以使用文本编辑器的“查找和替换”功能来用新的变量名快速替换所有出现的 FirstName。但是，最初的变化需求仅仅指定了在 myMethod1 方法中的 FirstName 需要改变，而 myMethod2 中的该方法则不需要改变。使用查找和替换将会错误地替换所有 myMethod1 和 myMethod2 中出现的 FirstName。当然，逐个替换出现的 FirstName 也是有可能的，但是这也是非常耗时的，并且可能人为出错。

许多文本编辑器都为编程语言提供了更好的支持。较为流行的、基于 Unix 的工具 Emacs 提供了很多有趣的特性，包括高级文本匹配和替换能力。通过安装插件，这些编辑器也可以提供 Java 语法强调、代码缩排、简单调试以及编译等支持。这些插件功能比较强大，但是这些编辑器仍然不能提供最灵活和高效的编程环境。

使用 Emacs 和文本编辑器的程序员问的第一个问题就是“为什么要使用 IDE”。有些程序员往往都会依赖于一个特定的工具集或者编程语言，并拒绝做任何改变。在当今日新月异的世界中，程序员一个非常重要的能力就是适应新技术的发展。

新的工具集在很多方面都可以给专业的编程人员提供帮助。作为一个程序员，应该将时间花在编写代码、重写代码以及测试代码上，而不是将时间浪费在试图解决如何重命名代码中的方法名、生成工程文档、或者正确地编译一个包中的所有类上。一旦确定了所要执行的动作，工具将非常快地为您完成这些任务。

集成开发环境(IDE)为开发工作提供了一个完整的环境。它将不同的工具以一种关联的方式集成到了一起。因此所需要的服务和动作都无缝地集成到了一起。

IDE 包含了很多的技术优点，如下所示：

- 执行动作的图形化用户界面(GUI)
- 将源代码和配置文件放入一个工程
- 与编译器紧密集成
- 与源代码库耦合
- 性能调整、分析和加载测试代码的能力
- 集成的可重用测试框架
- 使用第三方插件和工具的能力
- 通过一次执行一行代码来调试代码的能力
- 快速访问工程文档和易于生成工程文档

另外一些使用 IDE 的非常实用的业务优点如下所示：

- 减少开发的周期时间
- 提高代码的质量和可靠性
- 标准化软件开发过程
- 为编程人员提供一个通用的平台以减少培训时间

这些优点中有一些确实存在争议，并且有些可能需要通过仔细分析、实施和执行之

后才能真正实现。虽然受多方面因素的影响,但就像前面举的例子一样,一个真正好用的 Java IDE 工具是取得重要进展的基础。

NetBeans 是笔者对 Java IDE 的选择。笔者喜欢并使用 NetBeans 的理由有很多。笔者已经使用过很多开发工具,包括 Notepad、TextPad、Emacs、vi、Macromedia UltraDeveloper、Macromedia Dreamweaver、Oracle JDeveloper、IntelliJ IDEA、Borland JBuilder、Microsoft Visual Studio 和 Eclipse。

这些工具都有其优缺点。它们都希望所有用户和整个团体以它们为中心。在不久的将来,要区分这些工具就非常困难了,因为它们提供了许多相似的特性。笔者过去一直在犹豫是采用 IntelliJ IDEA 还是采用 Eclipse,但是在使用了 NetBeans 之后,发现 NetBeans 确实是一个不错的工具,其所有特性及功能都和所期望的一样,并且这些特性很少有或者根本没有配置问题。

编程人员愿意使用 NetBeans 而不是其他 Java IDE 的前 10 大理由包括如下方面,这些理由没有先后之分:

为 Swing 开发提供了直观且易于使用的 Matisse GUI 设计器:即使对 Swing 了解不多,甚至是一无所知,用户都可以立刻将元素拖拉到一个所见即所得(WYSIWYG)的设计窗口。Matisse GUI 设计器实际上产生了真正的 Swing 代码,而不再是其他一些工具所创建的样本文件代码。在最近举办的 JavaOne 会议上,笔者坐在一个使用 JBuilder GUI 设计功能的绅士旁边,该绅士在看笔者使用了 Matisse 几分钟后,就完全被它征服了,并立刻下载了该 GUI 设计器。

强大的重构支持:这一点对于 Jackpot 引擎是非常真实的。该引擎允许使用一个类似于正则表达式的查询语言来实现 Java 类型感知的重构。该查询语言由 James Gosling 设计,非常容易使用,并允许实施模式匹配和替换。而且,这些查询可以测试,以匹配特定的 Java 类型或者对象实例。

最优秀的代码分析器之一:在众多代码分析器中,NetBeans Profiler 是最好的一个。用户可以分析内存、CPU、性能以及监视线程。NetBeans 6 Profiler 引入了分析点的概念。该 Profiler 可以绑定到一个当前运行的进程或者应用,也可以和当前运行进程或应用取消绑定。它还提供了 32 位和 64 位支持,并允许对 Enterprise JavaBeans(EJB)模块以及企业应用进行分析。对于喜欢 Mac 系统的人员而言,它 also 支持分析 Mac OS X Intel 系统。

UML 工程支持:为了对代码、进程步骤或者是设计模式进行建模,程序员可以创建统一建模语言(Unified Modeling Language, UML)工程。UML 工程可以直接链接到 Java 工程。当用户创建并修改 UML 对象和 UML 图时,相应的 Java 代码将自动生成。如果在链接的 Java 工程中的源代码发生了改变,那么 UML 图也将自动地相应更新。由于具有导出 UML 图、生成代码以及创建基于 Web 的工程报告的能力,UML 工程特性是 NetBeans 最好的特性之一。

集成 Ant:在 NetBeans 中,Java 工程是使用 Ant 构建文件来结构化的。当一个工程第一次被创建时,IDE 生成该构建脚本和相应的目标。然后,用户可以触发特定的目标或者是完全自定义它们的构建文件结构以适应工程的需求。即使对于不熟悉 Ant 的用户而言,也不存在任何影响,因为 Ant 目标的执行直接链接到了 NetBeans 的菜单和按钮。

许多用户也发现从外部工程中导入已有的构建文件将大大地加快软件开发的速度。对于初学者而言，它非常容易使用。对于专家而言，它非常容易实现自定义。

J2ME 移动应用支持：即使没有进行过太多的移动应用开发，在查看一些示例，并阅读了介绍性指南后，就可以感受到 NetBeans 移动工具的能力，其所提供的示例应用令人深刻印象。由于支持 Java 2 Micro Edition (J2ME) Mobile Information Device Profile (MIDP) 2.0，且提供可视化的移动设计器、无线连接向导以及无线的下载测试，移动应用开发人员具有一些很好的、功能很强大的工具。

开发人员协同工具：开发人员可以登录到一个公开或者私有的环境，以共享代码。也可以加入一些公开会话，或者启动带有一些限制的私有会话。现在所能看到的最好的特性是能够将代码或者是整个工程拖到聊天窗口，并和一个或者多个程序员共享代码。NetBeans 支持多用户团队编码。当某个用户开始改变一个代码块时，该代码块被突出显示，并被锁定以和其他用户进行共享。在目前的全球经济环境下，开发团队散布在多个不同地方，因此该工具非常有用。

易用的 Update Center：NetBeans Update Center 允许快速地选择从哪个发布站点来检查改变、更新以及创新的模块。也可以选择安装之前下载尚未安装的模块。Update Center 比其他 Java IDE 的更新工具更加直观，并使得更新 NetBeans 更加方便。

固有的 JSP 和 Tomcat 支持：NetBeans 包含了 Apache Tomcat。只要使用了 New Project 向导来创建 Web 应用工程，就可以创建 JavaServer Pages(JSP)文件。然后可以右击任意 JSP 文件并选择 Run File。绑定的 Tomcat 服务器立即就会启动，默认的 Internet 浏览器将打开，并且在 Tomcat 中执行的 JSP 文件将会显示。NetBeans 甚至具有足够智能，可以激活 HTTP Monitor。

NetBeans HTTP Monitor：在参与了很多 Web 相关的 Java 开发后，笔者认为这是目前市面上所有 Java IDE 中最好的，也是独一无二的特性。HTTP Monitor 在调试或者执行 Web 应用时能够被激活。该工具允许监视请求、HTTP 请求头、Cookies、会话、servlet 场景以及客户端/服务器参数。并且也不再需要编写服务器端代码来读取这些变量，将变量输出到日志，并查看该日志文件。在 NetBeans 中，可以调试代码、逐行运行代码，并监视所需要监视的属性。

这些特性仅仅是 NetBeans 所提供特性的一部分。其他 Java IDE 可能也提供了这里所描述的一些功能，但是没有哪个 Java IDE 能够提供 NetBeans IDE 所提供的直观的界面和集成工具集。为了学习 NetBeans 所提供的所有功能，请继续阅读本书的其他章节。

本书将重点讨论 NetBeans IDE 6 所提供的一些新特性。当学习 NetBeans 时，可以重点学习某些类型的技术和领域。在最新的发行版本中，开发人员已经可以访问一些令人印象深刻的新特性了，包括如下特性，但并不限于此：

- 一个新的、改进过的 Source Editor
- 改进的重构能力
- 改进的代码补全
- 经过全面改进的 Profiler，包括分析点和 HeapWalker
- Maven 支持

- JUnit 4 支持
- Ruby 和 JRuby 支持

本书适用于各个层次的开发人员。不管是 NetBeans 的新手，或者是初级的程序员，还是具有丰富经验的资深程序员，都会发现本书将提供各个特性的直接解释以及简单示例。本书还重点介绍了 NetBeans 的一些核心特性以为专业的软件开发人员提供帮助，如 Ant、JUnit、CVS、Subversion 以及一些静态分析工具等。

笔者的个人网站 www.ProNetBeans.com 提供了大量信息，包含 Java 和 NetBeans 的新闻、文章和指南等。该站点还包含本书的更新、修改以及勘误。如果您有任何问题或者想提供反馈的话，可以发送电子邮件到 wkservice@vip.163.com。

目 录

第 1 章	NetBeans 的下载、安装 以及自定义设置	1
1.1	下载文件	1
1.2	安装 NetBeans IDE	2
1.3	自定义 NetBeans JVM 启动选项	4
1.4	管理插件和更新	5
1.5	设置代理	10
1.6	自定义 IDE	11
1.6.1	设置内部 Web 浏览器	11
1.6.2	设置代码编辑器缩排	11
1.6.3	选择字体和颜色	15
1.6.4	配置快捷键映射	17
1.6.5	设置 Advanced 选项	18
1.7	导航并了解 IDE 布局	21
1.7.1	初始布局	21
1.7.2	窗口	22
1.8	小结	24
第 2 章	Source Editor	27
2.1	在 Projects 窗口中工作	27
2.1.1	Source Packages	27
2.1.2	Test Packages	28
2.1.3	Libraries	29
2.1.4	Test Libraries	29
2.2	在 Files 窗口中工作	29
2.3	排列文件和导航文件	29
2.3.1	在 Source Editor 中 排列文件	30
2.3.2	在 Source Editor 中 导航文件	31
2.4	在 Source Editor 窗口中工作	32
2.4.1	打开文件	32
2.4.2	行号	32
2.4.3	代码折叠	32
2.4.4	当前行突出显示	33
2.4.5	语法和错误突出显示	33
2.4.6	标注槽和错误条	34
2.4.7	代码缩排	37
2.4.8	识别起始和结束括号	39
2.4.9	识别未使用的导入	40
2.5	Source Editor 菜单	41
2.5.1	上下文菜单	41
2.5.2	Editor 菜单(工具栏)	45
2.6	Source Editor 快捷键	47
2.7	支持的特性和工具	48
2.7.1	宏	48
2.7.2	组件 Palette	51
2.8	小结	53
第 3 章	代码补全和模板	55
3.1	代码补全	55
3.2	使用代码补全功能	58
3.2.1	包(导入)	58
3.2.2	方法	59
3.2.3	类成员	60
3.2.4	构造函数	61

3.2.5	super 和 this	61	5.2	Profiler 校准	103
3.2.6	new 操作符	62	5.3	分析 Java 应用的性能	104
3.3	代码模板	63	5.3.1	将 Profiler 绑定到工程上	104
3.3.1	使用代码模板	63	5.3.2	理解 Profiler Control Panel 窗口	104
3.3.2	自定义模板	65	5.3.3	CPU 性能分析	106
3.4	文件模板	68	5.3.4	分析 CPU 性能的例子	111
3.4.1	使用文件模板	68	5.3.5	用分析点来分析 CPU 性能	115
3.4.2	处理文件模板	70	5.3.6	内存分析	120
3.4.3	添加和创建模板	71	5.4	理解 Profiler Telemetry	131
3.5	小结	72	5.4.1	查看 Telemetry 概览	131
第 4 章	调试	73	5.4.2	查看主 VM Telemetry 窗口	132
4.1	什么是 IDE 调试	73	5.5	分析外部和远程应用	133
4.2	工程级的调试设置	74	5.5.1	分析运行在 NetBeans 上 的远程 Java 应用服务器	133
4.3	断点	76	5.5.2	用 Profiler Remote Pack 来分 析远程 Java 应用服务器	134
4.3.1	添加断点	76	5.5.3	分析外部 Java 应用	135
4.3.2	禁用断点	82	5.6	小结	137
4.3.3	删除断点	82	第 6 章	管理版本控制	139
4.3.4	自定义断点	82	6.1	使用 CVS	139
4.3.5	Breakpoints 窗口	83	6.1.1	配置工程使用 CVS	140
4.3.6	分组断点	84	6.1.2	执行常用的 CVS 操作	143
4.4	调试 Java 代码	85	6.2	使用 Subversion	153
4.4.1	启动工程调试会话	85	6.2.1	安装 Subversion	153
4.4.2	启动文件调试会话	86	6.2.2	执行常用的 Subversion 操作	154
4.4.3	停止调试会话	86	6.3	使用 Local History	157
4.5	单步调试代码	87	6.3.1	配置 Local History 属性	158
4.5.1	Step Into	88	6.3.2	使用 Local History	158
4.5.2	Step Out	89	6.3.3	标注版本	160
4.5.3	Step Over	90	6.3.4	删除版本	161
4.5.4	Step Over Expression	91	6.3.5	恢复版本	161
4.5.5	Run to Cursor	92	6.4	小结	162
4.5.6	Run Into Method	92			
4.5.7	Evaluate Expression	93			
4.6	用 Watch 调试	96			
4.7	Local Variables 窗口	98			
4.8	小结	99			
第 5 章	性能分析	101			
5.1	配置 Profiler 属性	101			

第 7 章 生成和访问 Javadoc	163		
7.1 Javadoc 元素	163		
7.1.1 类描述	163		
7.1.2 类标签	164		
7.1.3 类成员变量	165		
7.1.4 构造函数	165		
7.1.5 方法	166		
7.2 在 NetBeans 中创建 Javadoc	166		
7.2.1 配置 Javadoc 提示设置	167		
7.2.2 配置工程 Javadoc 设置	169		
7.2.3 生成工程的 Javadoc	171		
7.3 访问 Javadoc	171		
7.3.1 额外库和平台	171		
7.3.2 查看上下文敏感的 Javadoc	173		
7.3.3 搜索 Javadoc	174		
7.4 小结	176		
第 8 章 用 Ant 和 Maven 构建软件的管理	177		
8.1 Ant 任务和目标	177		
8.2 在 NetBeans 中配置 Ant 属性	181		
8.3 NetBeans 工程构建文件	182		
8.3.1 build.xml 文件	182		
8.3.2 build-impl.xml 文件	184		
8.3.3 build-before-profiler.xml 文件	185		
8.3.4 profiler-build-impl.xml 文件	186		
8.3.5 project.properties 文件	187		
8.4 使用目标	187		
8.4.1 运行目标	187		
8.4.2 调试目标	188		
8.4.3 停止和重运行目标	189		
8.4.4 为 Ant 目标创建快捷键	190		
8.5 Maven 简介	191		
8.6 使用 Maven 工程	191		
8.6.1 配置 Maven 属性	192		
8.6.2 创建 Maven 工程	192		
8.6.3 配置 Maven 工程属性	194		
8.6.4 添加类库依赖关系	195		
8.7 小结	195		
第 9 章 JUnit 测试	197		
9.1 创建 JUnit 测试案例	197		
9.1.1 创建新的测试类	197		
9.1.2 为现有类创建测试	200		
9.1.3 查看测试	202		
9.1.4 修改测试	204		
9.2 运行 JUnit 测试	205		
9.2.1 查看测试结果	205		
9.2.2 生成测试案例报告	207		
9.3 在 NetBeans 中配置 Junit 属性	208		
9.4 小结	209		
第 10 章 重构	211		
10.1 NetBeans 重构选项	211		
10.2 Move Class 重构	213		
10.3 Rename 重构	214		
10.4 Safe Delete 重构	214		
10.5 Use Supertype Where Possible 重构	216		
10.6 Move Inner to Outer Level 重构	216		
10.7 Encapsulate Fields 重构	218		
10.8 Pull Up 重构	222		
10.9 Push Down 重构	223		
10.10 Convert Anonymous to Inner 重构	225		
10.11 Introduce Method 重构	228		
10.12 Extract Interface 重构	229		
10.13 Extract Superclass 重构	230		
10.14 Change Method Parameters 重构	232		
10.15 重构的键盘快捷键	233		

10.16	小结	234
第 11 章	代码优化工具	235
11.1	使用 Checkstyle	235
11.1.1	Checkstyle 检查点	236
11.1.2	示例 Checkstyle 配置文件	241
11.1.3	在 NetBeans 中使用 Checkstyle	242
11.2	使用 PMD	248
11.2.1	PMD 检查点概述	248
11.2.2	示例 PMD 配置文件	251
11.2.3	在 NetBeans 中使用 PMD	252
11.3	使用 SQE	256
11.3.1	安装 SQE 插件	256
11.3.2	配置 SQE 插件	257
11.3.3	运行 SQE 插件	258
11.4	小结	260
第 12 章	在 Rail 上开发 JRuby/Ruby 应用	261
12.1	安装 Ruby 环境	261
12.2	配置运行环境	262
12.3	创建 Ruby 应用工程	263
12.3.1	Ruby 应用	264
12.3.2	基于现有源代码的 Ruby 应用	265
12.4	创建 Ruby on Rails 工程	265
12.4.1	Ruby on Rails 应用	265
12.4.2	基于现有代码的 Ruby on Rails 应用	267
12.5	添加文件到工程中	267
12.6	Ruby Editor	270
12.6.1	代码补全	270
12.6.2	代码模板	273
12.7	运行 Rake 任务	274
12.8	自定义 Ruby 工程	275
12.8.1	Ruby 工程属性	275

12.8.2	Ruby on Rails 工程属性	276
12.9	Ruby Gem Manager	276
12.10	管理 Rails 插件	277
12.11	测试 Ruby 工程	279
12.11.1	创建测试	279
12.11.2	运行测试	280
12.12	调试工程	281
12.13	IRB 和 Rails Console	281
12.14	JRuby	282
12.14.1	从 Ruby 调用 Java	282
12.14.2	在 Servlet 容器中运行 Rails	283
12.15	综合应用	285
12.15.1	创建数据库	285
12.15.2	创建工程	285
12.15.3	运行工程	286
12.16	小结	286
第 13 章	开发 Web 应用	289
13.1	创建 Web 应用工程	289
13.2	导航 Web 应用工程	291
13.2.1	Web Pages	292
13.2.2	Configuration Files	292
13.2.3	Server Resources	292
13.3	JavaScript 和 CSS 文件支持	293
13.3.1	处理 CSS 文件	293
13.3.2	处理 JavaScript 文件	295
13.4	构建 Web 应用	298
13.4.1	清除并构建一个工程	298
13.4.2	编译 JSP 文件	298
13.4.3	重复构建和停止构建	299
13.5	运行 Web 应用	300
13.6	定义 Java 应用服务器	301
13.6.1	使用 Tomcat	301
13.6.2	使用 GlassFish	304
13.6.3	为某个工程设置应用服务器	307

13.7	HTTP 监视.....307	14.5	创建 RESTful Web 服务.....384
13.7.1	启用 HTTP Monitor.....307	14.5.1	安装 RESTful 模块.....384
13.7.2	使用 HTTP Monitor.....308	14.5.2	根据模式创建 RESTful Web 服务.....385
13.8	使用 Web 应用框架.....310	14.5.3	创建客户端以读取服务.....388
13.8.1	使用 Struts.....310	14.6	小结.....389
13.8.2	使用 Struts 2.....316	第 15 章	开发 GUI 应用.....391
13.8.3	使用 Visual JavaServer Faces.....317	15.1	创建简单的 GUI 应用.....391
13.8.4	使用 jMaki 框架.....340	15.1.1	使用 Palette 窗口.....391
13.9	小结.....345	15.1.2	创建工程.....394
第 14 章	开发 Web 服务: JAX-WS、 SOA、BPEL 和 RESTful ...347	15.1.3	创建初始的 JFrame 类.....395
14.1	安装 Web 服务模块.....347	15.1.4	使用 Form Editor.....396
14.2	创建 Web 服务.....348	15.1.5	使用 FreeTTS.....405
14.2.1	创建 Web 服务.....349	15.2	使用 Swing Application Framework.....407
14.2.2	创建 Web 服务客户端.....355	15.2.1	创建 Java Desktop Application 工程.....407
14.2.3	根据 WSDL 文件创建 一个 Web 服务.....360	15.2.2	使用动作.....410
14.2.4	创建消息处理器.....361	15.2.3	使用 Application Actions 窗口.....415
14.2.5	创建逻辑处理器.....363	15.3	使用 Beans Binding.....416
14.3	处理 SOA 和 BPEL.....364	15.3.1	创建数据库.....417
14.3.1	创建 BPEL Module 工程.....364	15.3.2	创建工程.....419
14.3.2	创建 BPEL 流程文件.....365	15.3.3	分析生成的应用.....421
14.3.3	导航 BPEL Design 窗口.....366	15.3.4	理解 Update Source When 域.....425
14.3.4	创建 WSDL 文件.....368	15.3.5	编写自定义的 Validator.....426
14.3.5	导航 WSDL Editor.....371	15.4	小结.....429
14.3.6	使用 BPEL Designer 和 BPEL Mapper.....374	第 16 章	开发富客户端应用.....431
14.4	创建 Composite 应用.....381	16.1	NetBeans Platform 提供 的特性.....431
14.4.1	创建 Composite Application 工程.....381	16.2	入门.....432
14.4.2	设置 Composite Application 工程属性.....381	16.3	术语.....433
14.4.3	添加 JBI 模块.....382	16.4	NetBeans Platform SDK.....434
14.4.4	测试 Composite Application 和 BPEL 流程.....383	16.4.1	工程模板.....435
		16.4.2	文件模板.....436

16.4.3	NetBeans Platform Manager	437	16.6.2	使用 Window System API	449
16.4.4	Project Properties 对话框	439	16.6.3	整合应用	451
16.4.5	上下文菜单项	440	16.6.4	运行此应用	453
16.5	了解 Platform 之外的 API	442	16.6.5	发布应用	454
16.5.1	开始创建	443	16.6.6	更新应用	455
16.5.2	使用 Explorer & Property Sheet API	443	16.7	延伸阅读	455
16.5.3	使用 Nodes API	446	16.8	主要的 NetBeans API 小结	456
16.5.4	运行此应用	447	16.9	小结	457
16.6	组装一个 NetBeans Platform 应用	448			
16.6.1	开始组装	448			

第 1 章

NetBeans 的下载、安装以及自定义设置

自 NetBeans 5.5 发布以后，NetBeans 内核做了很多改进。其中一个重要方面就是安装过程。在过去的几年中，NetBeans 家族具有了很多不同的安装可选项，例如打包下载、独立的扩展包(add-on pack)以及独立的下载模块。

要使用 NetBeans，需要下载并安装该软件。本章除了将介绍如何下载并安装该软件外，还将介绍如何检查更新，以及根据个人的喜好自定义 NetBeans 设置。最后，将大致看一下今后开发中经常要用到的 NetBeans 窗口。

1.1 下载文件

首先讨论 NetBeans 6 的发行版，有多种可用于安装 IDE 的打包下载类型：基本包、标准包和完全包。

基本包(Basic Pack)：基本包由精简的 IDE 和基本的 Java Standard Edition 功能组成，包括性能分析工具和 GUI 构建能力。

标准包(Standard Pack)：标准包在基本包的基础上添加对移动性、Web 和 Java EE 特性的支持。它还包括了几个 Java 应用服务器，例如 Apache Tomcat 6 和 GlassFish 的最新版本。

完全包(Full Pack)：完全包除了包括前面提到的功能以外，还包括 UML、SOA 和 Ruby 模块。这是最大的下载包。不过如果不是实在需要简化版的 IDE，建议下载此包。在这个版本的安装过程中，可以选择需要的特性进行安装。而如果下载小一些的包，将没有这些选择。

请到 netbeans.org 上下载 NetBeans 包。此网站提供了多个下载 NetBeans 软件和相应工具的不同链接。

选择不同的路径下载，将可以下载不同版本的 NetBeans。通常由于您已经选定了操作系统，因此必须选择特定操作系统下的版本。但是如果未选定操作系统，则可以选择 Windows、Linux、Solaris 和 MacOS X 中的某个操作系统。在本书撰写时，Linux 和 Solaris 已经有 64 位可选了，但情况可能会发生变化。选择需要的下载包并单击 Download 按钮，