



持续集成及CruiseControl的权威之作

CI理论与实践并重

如期交付高质量复杂软件产品的必由之路

敏捷持续集成 (CruiseControl版): 高效研发之道

罗时飞

飞思科技产品研发中心

编著

监制



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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



敏捷持续集成

敏捷持续集成

敏捷持续集成

敏捷持续集成 CruiseControl

高效研发之道



敏捷持续集成 (CruiseControl^版): 高效研发之道

罗时飞

飞思科技产品研发中心

编著

监制

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

· 内 容 简 介 ·

在并重“理论与实践”思想的指引下，作者踏上了持续集成及 CruiseControl 的写作行程。全书共分为 5 部分内容：第 1 部分（初涉持续集成）由持续集成概述（第 1 章）、实施持续集成（第 2 章）构成，这部分内容的着眼点在于阐述相关的背景知识及初步认识持续集成；第 2 部分（敏捷持续集成）由基于 Subversion 的持续集成实战（第 3 章）、基于 CVS 的持续集成实战（第 4 章）、基于 VSS 的持续集成实战（第 5 章）、基于 ClearCase 的持续集成实战（第 6 章）构成，这部分内容将实战摆在了首当其冲的位置；第 3 部分（持续集成权威指南）由持续数据库集成（第 7 章）、持续单元及集成测试（第 8 章）、持续评审（第 9 章）、持续部署（第 10 章）、持续功能及负载测试（第 11 章）、持续反馈（第 12 章）构成，研究实战背后的理论知识成为了这部分内容的重点；第 4 部分（CruiseControl 高级特性）由分布式集成构建（第 13 章）、CruiseControl 内核研究及扩展（第 14 章）构成，这是属于 CruiseControl 自身的盛宴；第 5 部分由维护 CC 配置文件（附录 A）构成，它是查找 CruiseControl 配置文件相关知识点的权威去处。

本书适合作为 CI 集成人员、Java/Java EE 开发者、架构师、项目及产品开发经理的参考书，同时，本书非常适合于高校相关专业的学生，以及对持续集成有兴趣的各类读者。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

敏捷持续集成：CruiseControl 版：高效研发之道 / 罗时飞编著. —北京：电子工业出版社，2008.6
（开源技术专家）

ISBN 978-7-121-06557-6

I. 敏… II. 罗… III. 软件质量—质量管理 IV. TP311.5

中国版本图书馆 CIP 数据核字（2008）第 058061 号

责任编辑：王树伟

印 刷：北京智力达印刷有限公司

装 订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京海淀区万寿路 173 信箱 邮编：100036

开 本：787×1092 1/16 印张：21 字数：537.6 千字

印 次：2008 年 6 月第 1 次印刷

印 数：4 000 册 定价：39.50 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

Part I “开源”希望

更广阔的开源世界

在已经过去的 2006 年，开源世界发生了三件大事：Microsoft 与 Novell 达成协议，增强了 Linux 产品与 Windows 产品之间的互操作性，这一协议的签署无疑震动了整个开源世界；甲骨文在 OpenWorld 大会上宣布将提供对 Red Hat 企业版 Linux (RHEL) 的支持，其实现形式之一就是发行一个甲骨文版的 RHEL 克隆版；还有就是 Sun 将 Java 开源。

而在这三大事件中有两件都直接关系到了“开源先锋官”——Linux。长期以来，大家说到“开源”，似乎仅仅指 Linux，然而事实远非如此。在今天，我们可以随意列出一大串的开源软件 (Opensoft) 与开源代码 (Opensource) 清单。

2007 年里，开源的精神日益成为无数程序员们探求的信念。无论是开源软件 (Opensoft) 还是开源代码 (Opensource)，无论开源软件的使用者，还是开源开发的开发者，都把这种精神作为一种不断奋斗的动力。这些，都不再是 Linux 一枝独秀时代可以包容的内涵了。

在“开源”的世界中挑战自我

人们对开源软件的界定主要有两层含义。第一层认为开源软件意味着免费。开源软件的另一种说法是自由软件 (Freesoft)，其实就是从这个角度来描述的。在英文中，Free 一词本身就意味着免费，因此 Freesoft 就让开源软件蒙上了免费软件的外衣；第二层含义意指开源软件的源代码是开放的，相对于商业软件的开放性。无疑，开源软件的出现让人们多了一种选择，也让人们意识到部分商业软件的高额垄断利润。从这层意义上说，有人把开源软件的精神内涵界定为挑战他人。

可以说挑战自我是开源软件体现的第一重精神境界，也可以说是开源软件开发者的终极追求之一；开源软件的第二重精神境界则是对团队精神的追求。

“开源”坚守的团队精神，在国外以开源社区的形式存在，在国内虽然也有，但数量远远不够，还不足以形成规模。开源社区的不足是制约中国开源软件业发展的桎梏。国内开源软件的发展需要一个健全的社区开发机制，通过国内联合与国际合作，才有可能实现与国际接轨，避免低水平的重复，培养出我们自己的软件大师，形成具有竞争力的国际性的开源项目。这其实也解释了开源软件业对合作和团队精神的需求。

Part II “开源”动力

“开源”原动力

北京易飞思信息技术有限公司（简称飞思）是中央一级优秀出版社电子工业出版社最大的出版实体，是电子工业出版社第一个倡导和实施以产品全程策划和品牌营销思路进行图书立体出版的单位。飞思承担着电子工业出版社计算机图书的策划、研发和引进工作。2003年获电子工业出版社“先进集体”，2006年获信息产业部“青年文明号”光荣称号。

开源的世界纷繁精彩，我们无法用类似其他系列图书的规划方式来固化这个图书体系的架构，我们只会凭借多年积累的专业眼光与开源世界中努力拼搏的技术专家一起打造每一次的作品。

“开源技术专家”系列的关注点

正如在前面所说到的，无论是开源软件（Opensoft）与开源代码（Opensource），无论是开源软件的使用者，还是开源开发的开发者，都是我们关注的焦点。

开源世界，应用为先，关注开发者和应用者的每一个需求，是我们的努力方向。这个系列刚刚开始，它也将永远是“在路上”的状态，还希望得到大家的关心和支持。

飞思科技产品研发中心

联系方式

咨询电话：(010) 68134545 88254161

电子邮件：support@fecit.com.cn

服务网址：http://www.fecit.com.cn http://www.fecit.net

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

1990 年，我开始编写商业代码；大约 6 年前，我不再有机会去写代码了；4 年前，我甚至失去了作为成员参与项目一线实践的机会了。我要说，做一名程序员是我一生中最幸福的选择。这些年，在朋友和同事之间，我一直以程序员自居，只是越来越没有底气，甚至渐渐有些惭愧了。因此，给时飞的新作写序的心情是忐忑不安的，不安的是我并没有亲历他的持续集成实践，即便如此，作为一个旁观者，能参与时飞的 Java EE 架构实践讨论、持续集成实践讨论，并分享着复杂软件产品如期高质量交付的快乐，这是我的荣幸。

认识时飞并成为同事是 2002 年的事情，那时候时飞在网络上的 J2EE 布道已经很有影响了。2004 年，我们先后离开了那家企业，此后，时飞继续着他的软件实践—软件布道—软件实践；我则去经营一家银行应用软件企业。

2006 年年底，我和时飞又见面了，那时时飞已经是多本畅销著作和译作的作者了，我则在筹备中国建设银行管理应用集成开发环境（RIDE）项目。这个项目大约只有 5 个月的工期，涉及 200 多个复杂功能点、近 10 个软件工具和环境的中底层集成、大量的既有产品积累和开源实践，开发人员最高峰时有 50 多人。显然，这个项目的成败关系到企业的声誉甚至存亡，我邀请时飞加入该项目并担任架构师，然后我便如鸵鸟躲避攻击般地等待着“Death March”的宣判。2007 年 4 月 17 日，RIDE 如期交付了！今天，这个产品已成为中国建设银行管理应用的统一标准前端开发平台，并被数以千计的专业开发人员使用着。显然，时飞的加入也是这个产品成功的要素，这个产品也成就了时飞持续集成实践的新高度。再后来，我们承担了中国建设银行统一数据服务环境（MUSE）项目，其复杂度和难度远超过前者，时飞担任 MUSE 的架构师和集成负责人，我们同样如期高质量地交付了产品，MUSE 再次成为中国建设银行数据线的标准平台，并得以在全行推广。2008 年年初，时飞决定闭关修行，他要布道了，现在，我们看到了这本新作。

这些年的实践中，我对软件之道的理解越来越“忽悠”：架构优先、坚持重用、持续迭代、以人为本。持续集成推崇的敏捷软件生产实践，支持了我的理解，为此我很开心。我不愿对时飞的持续集成做评价，担心那会破坏读者的阅读胃口或者错误解读时飞的道场气氛。时飞说：“成功的产品研发都是持续行为，是长期积累的结果。借助于持续集成，产品的交付质量得以持续改进。”这可能是我这点附庸风雅的文字中唯一地对持续集成的直接喝彩。无论您是程序员，还是软件开发管理人员，我相信本书都能给您带来知识和启发。

祝您阅读愉快。

贺乐天 博士
首席架构师，宇信易诚

持续集成宣言

在现有主流的软件研发实践中，存在一种名为“持续集成”的敏捷实践。持续集成（Continuous Integration, CI）是“节能减排”的一种重要体现，是一种绿色实践，它在拒绝浪费、降低产品研发风险、提高软件质量方面起到了举足轻重的作用。

纵观企业应用研发的全生命周期，其中，80%的工作几乎都是重复劳动，这类工作完全可以托付给计算机完成。而剩下的20%左右的工作则需要智力劳动的大量投入，这正是“没有银弹”（No Silver Bullet）的深刻体现。持续集成能够辅助及自动完成这些重复劳动，并为智力劳动保驾护航。

对于软件研发而言，其中的智力劳动体现在很多方面，比如数据库设计、产品架构设计、编写出高质量的代码、设计出一流的测试方案等。计算机是无法自动完成这类工作内容的，因为完成它们需要富有经验和智慧，现阶段的计算机还无法展现出这方面的涵养。至于重复劳动，也包括众多的工作类型，比如数据库的创建及初始化、运行各类测试代码、已实现基代码是否遵循了架构及开发规范等。

重复劳动是果，而智力劳动是因，因为有了智力劳动，才会产出大量的软件产品。然而，为了获得高质量的软件产品，在产品研发期间，需要持续地完成大量的重复劳动。一直以来，人类在不断研发新的软件产品，因此 CI 便能够持续地介入进来，进而让计算机重复地完成各种具有重复性质的工作内容。这或许就是社会分工的延伸。

从全世界范围的软件研发趋势来看，开源项目都实施了 CI 工作，大量商用软件的研发工作也在逐渐拥抱这一敏捷实践。在作者过去多年的研发经历中，到处能看到 CI 的身影，涉及的软件行业千差万别，从金融（银行）到保险软件，从商业流通到出入境软件，等等。

翻开此书后，CI 集成人员将了解和掌握 CI。也正是从善待 CI 的那一刻开始，我们一起向全世界宣布，全力拥抱持续集成吧！

在此，作者要感谢出版社、感谢广大读者的支持。感谢与我共事的宇信易诚商业智能（Business Intelligence, BI）研发团队，是你们的努力付出和宽容，才使得持续集成及 CruiseControl 在大大小小的项目中被广泛、深入应用开来。同时，还要感谢中国建设银行厦门开发中心对 CI 的支持。另外，感谢多年好友贺乐天百忙中写的代序，也感谢他一直以来的鼓励和我写作的支持。没有你们的支持，本书的写作和出版不会这么顺利，再次真诚地谢谢你们。最后，不得不提的是，感谢妻子王女士一直以来对我写作的支持。

另外，持续集成及 CruiseControl 涉及的知识面非常广，内容很广泛，本书错误在所难免，希望读者、同行批评指正。

罗时飞

网站介绍

关于 www.open-v.com

我们专注于 Java EE 平台、敏捷方法及开源技术咨询工作。图书创作能够将我们的咨询经验带给整个社区，我们也希望在不远的将来，open-v.com 能够成为图书创作的重要平台。有特色的技术培训能够缩短开发者、开发团队采纳敏捷技术、方法的时间，因为我们的培训注重理念、学习方法和学习能力的提高。有价值的咨询服务也是我们的重点，我们非常注重咨询的效果，从而真正为企业带来实实在在的价值。

现如今，持续集成（Continuous Integration，CI）是软件研发领域应用最为广泛的敏捷实践之一，为辅助和保证 CI 工作的成功实施，CI 服务器是不可或缺的。CruiseControl 是最为流行、功能最强大的 CI 服务器之一。可以看出，借助 CruiseControl 实施 CI 工作，无形中，这已经为 CI 工作的顺利实施奠定了良好的基础。

作者有幸深入到这一实践中，并围绕它完成了此书的写作工作。写作工作前后历时 2 年，现在终于成功出版。实践出真知，这一直是指引作者前行的指南和动力。同样地，作者也一直不敢忽略理论与实践相结合的重要性，因为前脚迈出的同时，后脚还得立即跟上。在过去 4 年的工作当中，作者一直在自身的团队中努力借助 CruiseControl 实施 CI 工作，并不断地改进整个团队的 CI 过程。

本书特点

- 遵循理论与实践相结合。
- 注重持续集成背景知识及相关理论的阐述，第 1 部分内容（“初涉持续集成”）、第 3 部分内容（“持续集成权威指南”）便是见证。
- 深入到基于 CruiseControl 2.7.2 的持续集成工作的各个层面中。从 CruiseControl 的基本使用，到基于各种主流 SCM 工具的 CruiseControl CI 实战，到 CruiseControl 分布式集成构建，到 CruiseControl 构建网格，等等。读者能够从此书分享到全方位的 CruiseControl。
- 认真检阅<cruisecontrol/>全家福，从而为 CruiseControl 配置文件的编写扫清障碍，最终为 CI 工作的成功实施奠定坚实的基础。

全书内容安排

全书共分为 5 部分内容。

- 第 1 部分（初涉持续集成）：由持续集成概述（第 1 章）、实施持续集成（第 2 章）构成，这部分内容的着眼点在于阐述相关的背景知识及初步认识持续集成。
- 第 2 部分（敏捷持续集成）：由基于 Subversion 的持续集成实战（第 3 章）、基于 CVS 的持续集成实战（第 4 章）、基于 VSS 的持续集成实战（第 5 章）、基于 ClearCase 的持续集成实战（第 6 章）构成，这部分内容将实战摆在了首当其冲的位置。Subversion、CVS、VSS、ClearCase，这些都是主流的 SCM 工具，围绕它们实战持续集成工作具有不寻常的意义。
- 第 3 部分（持续集成权威指南）：由持续数据库集成（第 7 章）、持续单元及集成测试（第 8 章）、持续评审（第 9 章）、持续部署（第 10 章）、持续功能及负载测试（第 11 章）、持续反馈（第 12 章）构成，研究实战背后的理论知识成为了这部分内容的重点。
- 第 4 部分（CruiseControl 高级特性）：由分布式集成构建（第 13 章）、CruiseControl 内核研究及扩展（第 14 章）构成，这是属于 CruiseControl 自身的盛宴。

•Foreword•

- 第 5 部分：由维护 CC 配置文件（附录 A）构成，它是查找 CruiseControl 配置文件相关知识点的权威去处。几乎可以认为，只要 CI 集成人员掌握了 CruiseControl 配置文件的编写，则单纯地从技术层面看，CI 工作的实施将不会存在太多的障碍。

服务网站

针对书中展示的各种 CruiseControl 配置文件、Ant build.xml、其他脚本，我们特别提供了支持网站（<http://www.open-v.com>）。同时，为保证图书同 CruiseControl 最新发布版的同步，我们会时常更新它们，并公布到这一支持网站中，欢迎广大读者下载使用。

罗时飞

第 1 部分 初涉持续集成

第 1 章 持续集成概述	3
1.1 敏捷软件研发	3
1.1.1 提炼重复任务——实现可回归性的重要前提	4
1.1.2 敏捷性——奔向成功的致胜法宝	7
1.2 何谓持续集成	7
1.2.1 驱动 CI 的引擎——CI 服务器	10
1.2.2 敏捷配置管理	11
1.3 CruiseControl 初探	11
1.4 小结	14
第 2 章 实施持续集成	17
2.1 CruiseControl 的下载和安装	17
2.1.1 登录到 CruiseControl 的遗留 Web 控制台	19
2.1.2 登录到 CruiseControl 的新一代 Web 控制台	21
2.2 connectfour 示例分析	22
2.2.1 connectfour 项目背景	22
2.2.2 分析 config.xml 配置文件	24
2.3 自定义 CruiseControl 的运行时行为	26
2.3.1 保护 JMX HttpAdapter 适配器	29
2.3.2 借助于 RMI 操控 CruiseControl 服务器	31
2.3.3 调整 CruiseControl 的 JVM 内存使用	33
2.4 小结	34

第 2 部分 敏捷持续集成

第 3 章 基于 Subversion 的持续集成实战	37
3.1 CruiseControl 内置的 SCM 支持	37
3.2 Subversion 的安装及配置	38
3.2.1 安装 Subversion	38
3.2.2 配置及初始化 Subversion 配置库	40
3.3 Spring PetClinic 概述	43
3.4 借助 Subversion 和 CruiseControl 实施持续集成	44
3.4.1 从 Subversion 检出 petclinic 项目	44
3.4.2 配置 config.xml	44
3.4.3 触发持续集成工作	47
3.4.4 查看持续构建结果	48
3.5 Subversion 客户端支持	56
3.6 CruiseControl 若干技巧	57

3.6.1	模块化 Ant 文件	57
3.6.2	稳健应对构建失败	58
3.6.3	控制收集修改集合的策略	60
3.6.4	自定义构建产出物的分发渠道	61
3.6.5	借助 SVNLabelIncrementer 插件控制构建 Label 的生成	62
3.7	小结	64
第 4 章	基于 CVS 的持续集成实战	65
4.1	CVS 的安装及配置	65
4.2	基于 CruiseControl 的持续集成实战	67
4.2.1	检出位于 CVS 配置库中的 petclinic 项目	67
4.2.2	配置 config.xml	68
4.2.3	开始实施 CI	69
4.3	CVS 客户端支持	71
4.4	CruiseControl 若干技巧	71
4.4.1	自定义邮件分发策略	71
4.4.2	控制不同项目的并行构建能力	74
4.4.3	监控 logs 中的日志文件	77
4.4.4	传递给各构建脚本的系统属性	79
4.4.5	借助 CVSLabelIncrementer 插件控制构建 Label 的生成	81
4.4.6	自定义构建时机	82
4.5	小结	84
第 5 章	基于 VSS 的持续集成实战	85
5.1	VSS 的安装及配置	85
5.1.1	VSS petclinic 配置库的初始化工作	85
5.1.2	借助 Eclipse VSS 插件导入 petclinic 项目	86
5.2	实施基于 VSS 的持续集成	89
5.2.1	“检出” petclinic 项目	89
5.2.2	配置 config.xml	90
5.2.3	调整 Ant build.xml 构建文件	91
5.2.4	涉 CI 之初	92
5.3	CruiseControl 单次构建过程详解	93
5.4	小结	99
第 6 章	基于 ClearCase 的持续集成实战	101
6.1	ClearCase Explorer 客户端的安装及配置	101
6.1.1	设定 CLEARCASE_PRIMARY_GROUP 环境变量	102
6.1.2	创建共享目录	102
6.1.3	借助 ClearCase Explorer 创建静态视图	103

6.2	CruiseControl 持续集成的实施.....	106
6.2.1	验证 ClearCase Explorer 是否正常.....	106
6.2.2	实施基于 ClearCase 的持续集成.....	108
6.3	ClearCase Remote Client for Eclipse 工作情形.....	110
6.4	ClearCase 相关资源介绍.....	111
6.5	小结.....	113

第 3 部分 持续集成权威指南

第 7 章	持续数据库集成.....	117
7.1	持续数据库集成概述.....	117
7.1.1	DBA 的痛楚.....	117
7.1.2	将 SQL 脚本作为 SCM 配置项.....	117
7.1.3	持续 RDBMS 集成.....	118
7.2	数据库集成具体实践.....	119
7.2.1	Ant <code><sql/></code> 任务.....	119
7.2.2	数据库的销毁工作.....	120
7.2.3	数据库的创建工作.....	121
7.2.4	数据库中已有数据的删除工作.....	121
7.2.5	数据库中初始数据的准备工作.....	122
7.2.6	集成各 Ant 目标.....	123
7.2.7	手工运行“db-ci”目标.....	124
7.3	小结.....	126
第 8 章	持续单元及集成测试.....	129
8.1	持续单元及集成测试概述.....	129
8.1.1	开发经理的烦恼.....	129
8.1.2	引入测试覆盖度.....	131
8.1.3	持续单元及集成测试.....	131
8.2	有关单元及集成测试的具体实践.....	132
8.2.1	基于 JUnit 及 Clover 的持续集成.....	132
8.2.2	Cobertura 使用介绍.....	136
8.2.3	EclEmma 概览.....	142
8.3	小结.....	143
第 9 章	持续评审.....	145
9.1	持续评审概述.....	145
9.1.1	架构师揪心的那些事.....	145
9.1.2	量化开发规范.....	146
9.1.3	持续评审.....	146
9.2	持续评审实践.....	147

9.2.1	详细设计评审	147
9.2.2	PMD	149
9.2.3	Checkstyle	151
9.2.4	JDepend	152
9.2.5	JavaNCSS	154
9.2.6	FindBugs	156
9.2.7	Fortify	158
9.3	小结	159
第 10 章	持续部署	161
10.1	持续部署概述	161
10.1.1	烦恼的不只是你	161
10.1.2	制度化部署过程	162
10.1.3	持续部署	162
10.2	有关持续部署的具体实践	163
10.2.1	部署测试版本的 petclinic 应用	163
10.2.2	主流应用服务器内置的 Ant 部署支持	166
10.2.3	SmartFrog	168
10.3	小结	168
第 11 章	持续功能及负载测试	169
11.1	持续功能及负载测试概述	169
11.1.1	功能及负载测试人员的烦恼	169
11.1.2	脚本化功能及负载测试工作	170
11.1.3	持续功能及负载测试	173
11.2	有关持续功能及负载测试的具体实践	174
11.2.1	基于 Selenium 的自动化功能测试	174
11.2.2	基于 Apache JMeter 的自动化负载测试	179
11.3	小结	183
第 12 章	持续反馈	185
12.1	持续反馈概述	185
12.2	有关持续反馈的具体实践	186
12.2.1	E-mail	186
12.2.2	Ant 脚本	187
12.2.3	RSS	188
12.2.4	FTP	188
12.2.5	组合使用各种<publishers/>子元素	189
12.3	小结	190

第 4 部分 CruiseControl 高级特性

第 13 章 分布式集成构建	193
13.1 分布式集成构建概述	193
13.2 实施分布式集成构建	194
13.2.1 构建内置分布式集成构建的 CruiseControl 版本	194
13.2.2 启动“查找服务”	196
13.2.3 启用构建 Agent	196
13.2.4 从 SCM 配置库检出项目快照	198
13.2.5 分布式主构建器启航	200
13.2.6 启动 CruiseControl 服务器	202
13.3 有关分布式集成构建的若干技巧	203
13.3.1 关于构建 Agent 的一些实践	203
13.3.2 CruiseControl 内置的若干诊断工具	204
13.4 小结	206
第 14 章 CruiseControl 内核研究及扩展	207
14.1 远程调试 CruiseControl 服务器	207
14.2 剥离 Web 控制台	209
14.2.1 剥离/dashboard 控制台	209
14.2.2 剥离/cruisecontrol 控制台	214
14.3 基于/dashboard 控制台的构建网格	215
14.4 基于插件的架构	219
14.4.1 插件体系架构	220
14.4.2 现有的插件集合	223
14.4.3 灵活运用<plugin/>插件	227
14.5 有关 CruiseControl 的其他高级技术	229
14.5.1 揭秘 CruiseControl 构建周期	229
14.5.2 若干 CruiseControl 构建循环实现细节	231
14.5.3 CruiseControl 是如何构建类路径的	236
14.6 小结	237
附录 A 维护 CC 配置文件	239
A.1 <cruisecontrol/>权威指南	239
A.1.1 全局<property/>权威指南	241
A.1.2 <dashboard/>权威指南	242
A.1.3 <include.projects/>权威指南	243
A.1.4 <threads/>权威指南	243
A.2 全局<plugin/>权威指南	244
A.3 <project/>权威指南	245

A.3.1	项目级<property/>权威指南	246
A.3.2	项目级<plugin/>权威指南	246
A.3.3	<dateformat/>权威指南	246
A.3.4	<labelincrementer/>权威指南	246
A.4	<listeners/>权威指南	250
A.4.1	<cmsynergysessionmonitor/>权威指南	250
A.4.2	<currentbuildstatusftplistener/>权威指南	251
A.4.3	<currentbuildstatuslistener/>权威指南	252
A.4.4	<currentbuildstatuspagelistener/>权威指南	252
A.4.5	<lockfilelistener/>权威指南	252
A.5	<bootstrappers/>权威指南	253
A.5.1	<accurevbootstrapper/>权威指南	254
A.5.2	<alienbrainbootstrapper/>权威指南	254
A.5.3	<antbootstrapper/>权威指南	255
A.5.4	<clearcasebootstrapper/>权威指南	256
A.5.5	<clearcaseviewstrapper/>权威指南	256
A.5.6	<cmsynergybootstrapper/>权威指南	256
A.5.7	<currentbuildstatusbootstrapper/>权威指南	256
A.5.8	<currentbuildstatusftpbootstrapper/>权威指南	257
A.5.9	<cvsbootstrapper/>权威指南	258
A.5.10	<execbootstrapper/>权威指南	258
A.5.11	<gitbootstrapper/>权威指南	259
A.5.12	<harvestbootstrapper/>权威指南	259
A.5.13	<lockfilebootstrapper/>权威指南	260
A.5.14	<mercurialbootstrapper/>权威指南	260
A.5.15	<p4bootstrapper/>权威指南	261
A.5.16	<plasticscmbootstrapper/>权威指南	261
A.5.17	<snapshotcmbootstrapper/>权威指南	261
A.5.18	<starteambootstrapper/>权威指南	262
A.5.19	<surroundbootstrapper/>权威指南	262
A.5.20	<svnbootstrapper/>权威指南	263
A.5.21	<tfsbootstrapper/>权威指南	263
A.5.22	<vssbootstrapper/>权威指南	264
A.6	<modificationset/>权威指南	264
A.6.1	<accurev/>权威指南	266
A.6.2	<alienbrain/>权威指南	266
A.6.3	<alwaysbuild/>权威指南	267
A.6.4	<buildstatus/>权威指南	268
A.6.5	<clearcase/>权威指南	269